der Bundeswehr
Universität München

DEPARTMENT OF COMPUTER SCIENCE

Future Internet

Prof. Dr. Gabi Dreo Rodosek
Michael Hauser
Alexander Reinhold
(Hrsg.)

Institut für Technische Informatik

Report 2011-03
Dezember 2011

Werner-Heisenberg-Weg 39 • 85577 Neubiberg • Germany

Preface

The Internet was designed once to connect military facilities and became the most important communication infrastructure of our current highly developed civilization. It brings us affordable and comfortable services as e-mail that certainly influenced our everyday life and culture. Since the beginning of the Internet in the 1960ies it has been a continuously changing, growing and evolving system. Taken to extremes, this could mean the Internet is the central nervous system of our highly engineered culture.

But from another perspective there is a very fast technical progress and there are more ambitious applications which may expose shortcomings of the Internet as it is today. Highly mobile users and resource intensive applications define a completely new requirements set that must be delivered through the Internet. Therefore many research projects in the context of the Internet and its future (known under the term *Future Internet*) were initiated: From a complete redesign of architecture (clean slate approach) to an evolutionary development of current techniques several aspects were addressed in those projects. For example the ubiquitous connection of every-day utensils, high data availability in and between computing clouds and data centers, privacy concerns or the real-time detection of malicious hard- and software in a network describe just some problem fields existing in the present Internet.

In this seminar proceedings we will highlight some aspects of the major research area of Future Internet. Therefore we will have a look at new architectural approaches as well as security threats and current trends in our global and fast growing central communications infrastructure.

Michael Hauser

Contents

Chapter 1

Internet Privacy

Felix Ritscher

In our modern society, the Information Technology becomes more and more influential. The technical and social benefits, brought to us by this development, should not be underestimated. But this development also poses a lot of dangers. While the Internet is achieving every corner in our everyday lives, privacy loses more ground. In this paper, I will work out, how the situation of privacy actually is on the Internet. And I will demonstrate how the situation should be. And I will also try to figure out how privacy will change in future. To do so, I will introduce techniques, that protect privacy but also techniques that tries to resolve the privacy.

Contents

1.1 Introduction

This paper is part of the lecture ”Future Internet” which focuses on the privacy of data in the Internet.

”If you have something that you do not want anyone to know, maybe you should not be doing it in the first place.” [Google CEO Eric Schmidt][2]

This and many similar quotes concerning privacy are found in great numbers on the Internet. Reading such statements, the question arises if privacy is a lost concept of the analog world. The ongoing infestation of technology in our everyday life pushes the Internet more and more into the center of our society. Does this mean that we should not expect any privacy in the near future? If one is to trust these quotes, it is the only logical conclusion that the transparent citizen is no longer a vision of the future or a faraway horror story.

To analyze Internet privacy it is mandatory to predefine following definitions. The definition of privacy is fundamental, even if it has been taken granted in the past. The notion of privacy has to be restricted by the new technological world of the Internet, so that it enables us to clearly understand Internet privacy.

1.1.1 Privacy

Privacy is often defined as *”the right to be let alone”*[Warren and Brandeis][3]. However, this isn’t restricted to peace and solitude. Privacy means having the right, to develop ones personality independent from exterior influences and without the concern of being ascertained. Thus there have to be domains that are not controlled by ordinal parties. Nevertheless there are also other aspects concerning privacy. The right to be let alone brings forth the concept of intimacy and physical privacy; the privacy of one’s own body and his own celestial sphere of physical privacy which permits the an autarkic melioration of ones personality as unbound as sought after. Usually one simplifies the sphere of physical privacy in their own home. Other properties of the spheres of privacy are the interactional privacy and the informational privacy. The interactional privacy is the mastery over one’s interaction and communication with another person. Thus all conversations and analogous actions between small multitudes are a part of the interactional privacy. The informational privacy is particularly the control over private data. Someone’s private data, for an example sexuality, is colloquially referred to as private data. This information forms the informational privacy.

Privacy is a human right; in Germany various areas of privacy are protected by law. Physical privacy is governed by the *’Basic Law’*. According to *’Article 13, Number 1’* of the *’Basic Law’ ”ones domicile is inviolable”* This right of inviolability of one's home can only be restricted under certain circumstances as well as a judicial order. Furthermore the *'Basic Law'* also protects informational privacy. According to *'Article 10, Number 1'* of the *'Basic Law' "the privacy of correspondences and telecommunication is inviolable"*. But once again this law can be circumscribed by court order in particular cases. Many other laws and regulations exist which control the use of personal data, which protect the informational privacy. In modern society it is no longer evident that the development of the free and democratic society and government requires a comprehensive protection of privacy. Privacy is not only the cornerstone of society and government but also culture, science and religion. These and other areas are cultivated by unconventional intellectuals. People, who perceive things from a different angle compared to others, may come to inconceivable conclusions and results. The fact is however, that these outsiders and mavericks are openly marked or even prosecuted. Anyone who distances themselves away from the status quo is quickly discriminated. For an example, if Galileo Galilei would have been monitored with today's technological possibilities one could easily deduct his perspectives. Who knows if his ideas would have been ever published or if he would have even begun his studies?

"He, who is subjected to visibility and is aware of this, takes the coercive power and uses it against himself; he internalizes the balance of power in which he simultaneously plays both roles; he becomes the dictator of his own subjection" [Michel Foucault][1]

Those who are cognizant of potential surveillance will always be constraining their behavior in private areas more than he would do it otherwise. One would consciously adapt their behavior to the expected norms and will impute them over time. This mechanism is also called Panoptismus. In a long term this could be seen as a self-imposed constraint which is caused by potential surveillance and possible reprisals. Ironically, this restraint impinges on the principle of a emancipated development of one's personality which is in fact restriction of one's own possible actions. Even the potential restrictions of the right of privacy, as set-aside in German law, can induce such an effect if it is not handled with circumstantial precaution. For an example, it would be extremely indecorous if a police investigator could tap the telephone line solely based on his suspicions.

"I have nothing to hide, everyone can know everything about me and everything I do is legal." [Philipp Schaumann][2]

These and other statements are ergodic arguments against the integral protection of privacy. It is often proclaimed in our society there is no requirement to have secrets unless one plans to transgress the law. Such a noesis is credulous because not only the necessity exists to have secrets from the state but often enough there are justifications to have secrets from others .

For Example:

- Some secrets can cause embarrassment or shame in people when they are unveiled, may it be result of personal weaknesses or things that affect sexuality
- If secrets are revealed by a third person, these could be used to gain control or exercise power over the victim. In some cases these secrets could be used to blackmail someone. To build up the pressure on someone it is not necessary that these secrets have anything to do with illegality, it is enough if it could harm the profession or political career.
- Things you want to keep secret in order to prevent envy or jealousy
- If someone has specific information about a person, which could have negative consequences, such as discrimination.
- A harmless piece of information one may prefer to keep secret, because certain people or groups of people might come to wrong conclusions

There are certainly other examples of notions why one would want to have a secret, even though it may not be related to anything illegal. There are also areas of life that we have always perceived as very private. Such areas include:

- Religious affiliation
- Sicknesses
- Addictions
- Family problems, e.g. Divorce, illegitimate children
- Political orientation
- Sexual preferences
- Past and present legal offenses
- Financial status

A well-protected privacy is therefore of interest to anyone. Every person can make mistakes. Even law-abiding citizens, whose moral values coincide with public opinion, need a reserved area where they can make such mistakes. Otherwise it may, without an adequate protection of privacy, easily happen that one is exposed to public derision.

1.1.2 Internet Privacy

Modern technology, especially the Internet, increasingly impacts our present lives. Meanwhile, almost all households in Germany own a computer as well as an Internet connection. In addition, mobile devices with Internet capabilities are becoming increasingly popular. For anyone with a smartphone, being ”offline” is a state of emergency.

The offers on the Internet are enormous. A significant portion of societal life takes place in the Internet. On Facebook and GooglePlus, one can share ones status with friends and acquaintances. On these social networks, you hold contact with old friends and are informed about what ones Internet friends are planning to do in the future. You can upload photos to social networks, so friends can see and rate them. On YouTube you can watch entertaining videos or video blogs of friends and Internet celebrities. You can also get creative and created your own video addressed to the YouTube community. You can search for carpooling and sleeping facilities on the Internet or offer them. You can plan trips and book flights and hotels online. You can buy a wide range of wares on the Internet, whether one desires electronics, household items, or food, you can order anything and everything on the Internet. The Internet has no limits. To master your way across the cyber highway of information, you only need to google yourself across social platforms, forums, and shopping Google can be trusted with any question or request. With Google you can search the Internet for everything your heart desires. You can search for political articles and news or you can satisfy your religious or sexual curiosity. There is nothing on the Internet where Google cannot help. You can also write emails on the Internet or work with colleagues on a document online. Via Voice over IP, you can start real time conference calls, or make a video call to your family. In short, on the Internet nothing is impossible.

But the technology also poses various threats. Increasingly more social interactions take place on the Internet. This means that more and more of ones social life is digitized and then sent back and forth across the Internet. The Internet is not private but it is accessible to everyone. Large areas of private life, things that are normally done only under protection of privacy, expand to the realms of the Internet. It will be divulged personal data, relinquishing itself from ones control. The Internet jeopardizes privacy greatly. Therefore, privacy must also be protected on the Internet. Especially since the Internet, much like a public square, is exposed to everyone.

But what exactly is the intellection of privacy on the Internet? On the Internet there is neither an intimate sphere nor physical privacy. This neglects the interactional and informational privacy. The interactional privacy is the protection of all interactions and communications over the Internet. The interactional privacy on the Internet can be reduced to two properties, anonymity and confidentiality. If and only if both properties can be guaranteed on the Internet, the interactional privacy is protected. Although both properties should be guaranteed, confidentiality is the most important. Confidentiality means that all data that is sent over the Internet can be read only by receipient. This means that only the transmitter and the recipient may have the ability to read the data. In general, this can be achieved through an end-to-end encryption.

Anonymity is another very important aspect. To acquire and uphold anonymity on the Internet is impossible. Sometimes, if anonymity is required, it is also expected that one or all participants of a conversation are anonymous to each other. If the communication on the Internet is not anonymous, it results in a couple of ancillary effects. First, you can easily keep track of which sites Internet users visited. Web pages are relatively static. This may allow using only the information that a user has invoked a number of websites to reconstruct the actions of the user on the Internet. Without anonymity on the Internet, it is possible to track participants of a conversation. That alone contradicts the understanding of the interactional privacy.

Informational privacy on the Internet is an even more sensitive subject. The desire is that everyone maintains control over information relating to him. Everyone can delete this information and can determine with whom it may be shared. Achieving this is not easy. With each use of a service on the Internet, it reveals information about itself, such information is highly sought after by companies in order to develop more effective advertising. With this data, it is possible for companies to earn a lot of money. Access to the corresponding services is then, in many cases, offered free or heavily discounted. There are many business models based on such data. The question is what information is personal and should be protected. The opinions about what information is private differ significantly so that it cannot be answered easily. The problem is that it would destroy many business models, if one prohibits the collection of data or restricts it too much. However, it can quickly lead to a transparent human, if the collection of data will neither be restricted nor controlled. Much information is also disclosed voluntarily by the users, for example, if you want to share experiences online with friends or get the the above mentioned discounts. Rules to protect this data differ in many countries around the world. These regulations are in most cases only valid in the respective countries. Because the Internet is a global network, it is extremely difficult to track violations of these regulations. It is therefore almost impossible to enforce these rules on the Internet.

To create effective data protection rules, it is required that an international consensus on this issue is formulated. The so-developed data protection rules must then be ratified and enforced internationally. This subject is beyond the scope of this paper. The focus of this paper is on how we can avoid the spread of private data in the first place. So, how can we guarantee our anonymity and confidentiality over the Internet through technical means?

1.2 Confidentiality

The first topic which is discussed in this report is confidentiality. Confidentiality means that the content of a communication may only be read by certain authorized people. In general, this is the transmitter and the receiver of data. To ensure confidentiality, the data must be secured against unauthorized access. This can be done in different ways.

One possibility is to send the data only over secure channels. A secure channel or tap-proof channel is a compound that cannot be intercepted. In such a channel, it is physically impossible for an attacker to read the traffic thus guaranteeing the confidentiality. However, secure channels in practice are relatively rare. It is also easy to understand why these channels are rare. Each channel which is additionally utilized by people who may not follow a communication is no longer physically intercepted. Even in an exclusive channel, the physical security against eavesdropping is not given automatically. You may need to ensure that there is no point at which a potential attacker can gain access. For any wireless communication that cannot be guaranteed. These conditions can be fulfilled only by physical connections, such as direct cable connections. Such compounds have to be monitored accordingly to ensure their safety or they have to be so short that it needs no supervision. Even if the demands are high, there are still examples of secure channels. Such a channel does not have to be a cable connection. It is such a secure channel, if one collects the relevant data on a mobile device and transfers this unit to the communication partner. The normal four-eye meeting is, in the age of bugs, cell phones and directional microphones, no more a secure channel.

As the Internet is freely accessible to all, there are no physically secure channels. However, it is possible to have a virtual secure channel with the help of data encryption. With the help of cryptography, it is possible to encrypt messages in such a way so that it is impossible for an attacker to decrypt the message. At least it must be so expensive that it no longer pays for the attacker to decrypt the message.

The creation of a virtual secure channels can be achieved in different ways. The most popular option is the end-to-end encryption. In this variant, the communication is encrypted by the sender directly and can only be decrypted by the real recipient. Such procedures are very safe, because their safety depends only on the encryption used. If an end-to-end encryption is used, the communication can also be handled over insecure channels and is still confidential. Whether it is a download, an e-mail or a normal letter, the principle remains the same. Sender and receiver agree on a key and then use it to encrypt their messages. The encrypted message can be sent over any channel to the receiver without the need to worry that someone intercepting the message and reads it. The recipient then uses the negotiated key and decrypts the message. Then he transforms the message back into its original shape. The keys can be exchanged in various ways. Either they are previously exchanged over a secure channel, then the encryption procedure is also called symmetric, or the sender receives from the receiver over a non-secure channel, keys to encrypt the message with which he alone cannot decode. In this case the process is also called asymmetric. The difference between these two methods is that the symmetric encryption requires a secure channel and the asymmetric encryption needs more processing power. Therefore, both methods are often combined, which is then called hybrid encryption. It generates a secure channel by using an asymmetric encryption. Through this channel, the keys for symmetric encryption are exchanged.

Another option is to create a secure channel to the Internet is a tunnel. A tunnel connects two points on the Internet, using a virtual secure channel. The connection is established in a similar manner as the end-to-end encryption. It is not necessary that the two connected points are the end points of a communication. This is useful because it may happen that you want to access a private and secure network from outside. It creates, with the help of a tunnel, a secure channel to another network. Through this tunnel you can now communicate confidentially with the devices of the network, so that not every device must encrypt their communications separately.

The Internet Protocol is not designed for these safety implementations. In the time when the Internet protocol was designed, nobody was expecting such a wide use of the Internet, as we experience today. Because of this low incidence, there were no safety concerns in the first place. Today it is possible to intercept data on the Internet with simple technical means. A general encryption as described was not intended in the original Internet Protocol.

This is not optimal, because all of the encryption must be implemented retroactively. It will not encrypt the IP packet itself, but the data sent in an IP packet has to be encrypted. One can imagine the difference like a sealed letter and an open letter. A sealed letter is like an encrypted IP packet. The contents of the letter cannot be read because it is protected through the seal. In an open letter on the other hand one can view the content of the letter, which makes it necessary to protect the text of the letter itself. The content of an IP packet needs to be protected by an additional protocol. Unfortunately, there is no unified protocol so that a multitude of different protocols have been created. These protocols differ when it comes to confidentiality, mainly caused by their encryption methods.

Despite, or perhaps because of the great abundance of protocols, encrypted traffic is still not widespread. For example, just nine percent of U.S. companies encrypt their VOIP calls. It is likely that an even smaller portion of private persons encrypt their traffic. In order for encryption to be successful, the sender and receiver must agree on a protocol which is not always possible. Another obstacle is that it is also necessary for both receiver and transmitter have enough computing power so that the encryption/decryption can be performed in an adequate period.

This could change with the new version of Internet Protocol (IPv6). With IPv6, there will be a much larger address space, and many things will be simplified. With IPv6, the IPsec protocol is a standard feature. IPsec is a security protocol which operates directly on the Internet Protocol. Unlike the current version of the Internet protocol, Ipsec automatically features the possibility to encrypt data. Experts also say that IPsec is the best available security protocol and even before Ipv6, it was used in many other areas..

With IPsec, you have, among other functions, the possibility of selecting one of several encryption methods with which data can be protected. This includes both symmetric and asymmetric algorithms; it implements a tunnel-mode and an end-to-end encryption. Generally, there are a lot of settings for IPsec. The abundance of settings is however facing a lot of negative critique. There is criticism that IPsec is too complex and thus only functional with restrictions. Due to the enormous complexity of the IPsec protocol, a lot of problems arise concerning the implementation. A very complex implementation can quickly contain errors that lead to security vulnerabilities. Although the protocol is already used, it is constantly in development. The protocol features that faced criticism have been completely revised. After this revision, many deficiencies have been rectified, so that it has lost part of its complexity. Experts however still complain that IPsec is inherently too complex and contains too many unnecessary features.

Despite its shortcomings, IPsec is still the best IP-based security protocol available. Even if IPsec has vulnerabilities, it is nevertheless a major advance in order to keep data confidential. In the context of privacy, it is important to protect all data possible. Through techniques of data mining and large data collectors, such as Google, even worthless information set in a context reveal a lot about someone. Even a weak encryption is useful in order to protect a large amount of unimportant data. If all data is protected, even if they seem irrelevant, then the cost increases for a potential attacker. Even when using weak encryption, this effort may be high enough so that the attacker loses the interest. As the standard protocol is IPsec available to everyone, it is not necessary to agree on a shared protocol, before starting a communication. Each communication partner only needs to configure his hardware and enable encryption. So that common people can also use IPsec, the protocol must however be improved further.

1.3 Anonymity

The next important point in this paper is the anonymity of the Internet. It is not enough that the data is confidential because you can, even if the traffic is encrypted, reconstruct communication partners and find out who has visited which website. Judging the user's Internet surfing habits one can possibly render even important information. If for example the website of the NPD is often called, it is easy to draw conclusions about the political orientation of a person. Whether these conclusions are correct or not is irrelevant, because privacy is harmed anyway.

The Internet protocol was not designed for anonymous communications. It was the basic idea of the Internet protocol that each device that is connected to the network is assigned a unique address. This address is the so-called IP address, a 32 bit number. The address range consists of IP addresses, summing up to about four billion IP combinations. Of these addresses one part is reserved for private or related users, these addresses are not available on the Internet. They are used, for example, to set up private networks. The IP address of the sender and the receiver are attached to each IP packet. This is like a signature on each IP packet. These data can not be encrypted; else the IP packets are not deliverable.

IP addresses are assigned by the Internet Assigned Numbers Authority (IANA) in large blocks. The IANA distributes these blocks to the five Regional Internet Registries (RIR). The RIRs divide their allocated blocks and assign them within their region to the Local Internet Registry (LIR). The LIRs are, in general, Internet Service Provider (ISP). The ISPs assign IP addresses either directly to a customer or another provider. Sometimes the RIRs assign blocks directly to larger companies or universities. The allocation process of the IP addresses can be tracked via the WHOIS database of the RIR.

Because of the great success of the Internet, more and more devices are connected to the Internet, resulting in the number of units now exceeding the number of available IP addresses. This quickly leads to a new procurement strategy. Today, IP addresses are not permanently assigned, but are assigned only when needed. A customer will then get the IP address assigned, which is currently free so that the ISP has the capability to maintain more customers with fewer IP addresses. For customers, this means that he gets a different IP address every time he connects to the Internet. Through this dynamic IP address assignment, it is not possible to conclude from the IP address to users. ISPs are companies that collect fees for their services. To collect these fees, ISPs must store data that will help them to create an invoice. In this connection data, the IP address is stored. With the help of such data connections, you can track the identity of the owner of an IP address. With a court order, it is possible to get access to these data connection. The data can also be stolen. The dynamic IP address assignment does not guarantee anonymity on the Internet.

With the new IPv6, the old IP address is replaced by a new 128-bit number. This represents an incredible amount of addresses. It would be theoretically possible to assign every square millimeter of the Earth, 669 quadrillion IP addresses. With this huge amount of IP addresses, it will no longer be needed to assign IP addresses dynamically. You can easily assign each connected device a globally unique and permanent IP address. That would make it possible to trace identities with far less effort. And sooner or later databases would be created, through which it becomes possible, to trace any IP address back to their owner.

1.3.1 Anonymizer

Precisely because of these dangers, it is important to protect ones own anonymity on the Internet. This can be achieved by using so-called anonymizer services. It is the goal of anonymity services to allow communication on the Internet without revealing your own IP address. With these services, it's all about the anonymity of the IP address. One's identity can also be revealed by the content of communication, but this is not handled by these services. The focus of this paper is only on the anonymity of ones own IP address. This means that traffic information is hidden to outside attackers as well as the intended receiver.

The simplest of these services is called the proxy. A proxy server acts on the Internet as a substitute for a communication participant. Such a proxy server then identifies itself to the recipient as the sender of a message. In the best case, the recipient doesn't know that he communicates with a proxy. The proxy works surprisingly simple. If someone wants to build a connection for example to a web page, without revealing his IP address, he sends his request to the proxy instead of directly to the website. When the proxy receives the message, it stamps it with its own IP address and passes on the request. When the proxy receives a response, it forwards them to the sender of the request. Such a procedure is not safe. To restore the original IP address, you only need to access the proxy server. It would also be enough if you listen to all communications of the proxy server. A proxy server must be very trustworthy, because you direct all communication via this proxy server.

JonDo

JonDo is a mix network that has emerged from the former project ”AN.ON Java Proxy” (JAP). JonDo itself stands for ”JonDonym Anonymous Proxy Server”. This network creates mixed cascades, from so-called mixes. Mixed cascades are serially connected mixes.

A mix works much like a proxy. It receives messages and them on with its reference. The difference to a normal proxy is that a mix of protective measures are taken to guard against disclosure of communications data. So a mix can work optimally, it must consolidate multiple communications from different users. The mix collects the incoming data and edits it before he reroutes it back to the user. He sorts the incoming data, so that it the mix does not send it in the same order as they have arrived. He also deletes identical messages. In Mix server the messages are transcoded so that no incoming and outgoing messages can be linked. The messages are transcoded so that an attacker cannot compare them bit by bit.

A mix alone is not trustworthy enough. An attacker can gain access to a mix or the mix itself may be an attacker. To secure the system against such an attacker, different mixes are connected to a Mix Cascade. When a message is received by the first mix of the cascade, the mix sends it to the next mix in the cascade. This process is repeated until the end of the Mix Cascade. The last mix in the series then passes the message to the appropriate recipient.

Theoretically, this principle is very safe. An attacker would have to check all the mixes so that he can return the traffic data. It is not even enough if an attacker listens to the entire underlying network. For various reasons, not all functions of the mixes are fully implemented. This can lead to security vulnerabilities. Due to such vulnerabilities, it is possible that an attacker can return the traffic data de-anonymized by controlling only the first mix of the cascade.

JonDo tries to eliminate these weaknesses through technical certifications. JonDo offers its customers a choice of Mix Cascades. For each cascade, the users can view the identity of the Mix operators. Then they can decide which of the Mix Cascades is suited best for them. JonDo verifies the mixes, with the help of their own ”JonDonym Certification Authority”.

JonDo charges a fee to finance these mixes. A four-month service with a capacity of five gigabytes per month costs €75. JonDo also offers some free Mix Cascades, these are funded through donations. This Mix Cascades achieve speeds of only 30-50 Kbit / s. In addition, the free Mix Cascades have to accept cuts in safety. These mix cascades are shorter and are usually not distributed globally. The functionality of the free mix cascade is much lower, this only works for HTTP and HTTPS protocols. Accordingly you have to decide whether you want to spend money for an efficient system or are satisfied with the free alternative.

The Onion Routing

Another way to hide the IP address is ”The Onion Routing” (TOR). Unlike JonDo, TOR is based on a peer-to-peer network, the use of this network should ensure anonymity. As a peer-to-peer solution, TOR is free.

If you use Tor to anonymously retrieve a Web page, no direct connection between user and website will be established. The TOR client randomly selects three active TOR nodes; these nodes will set up a connection to the website. After a fixed period of time, another connection will be set up. A TOR node is not a dedicated server. All Internet users can create and deploy a TOR node. Each TOR node is registered so they can be found by other clients.

The request to the Web page is encrypted in multiple layers. The data is encrypted so that each Tor node which is involved in the connection can decrypt exactly one layer. Hence the term ”onion routing”. The data is then sent over the previously chosen path. Here, each node decodes the incoming data and obtains instructions wherever he needs to send the data. If no end-to-end encryption is used, the last node gets access to the data in plain text. This procedure ensures that each node knows only its immediate neighbors. Such a connection must use at least three nodes, so that the exit node does not know the entry node and vice versa. The response of the website is handled similarly; it arrives at the exit node. The exit node then encrypts the response and sends them on. The TOR client then receives the response from the website; it is encrypted in multiple layers and must be decrypted by the client. The subsequent detection of the IP address for TOR is not possible, since each node, if configured correctly, does not create log files.

TOR has another big advantage. Since TOR is a peer-to-peer network, anyone can create a node. Just by running a Tor node, the local traffic will be anonymized. When one runs a TOR node, you get messages to forward them. For an attacker it is no longer immediately clear which data is incurred locally and which data is only forwarded. Your own traffic is also obscured.

TOR is a relatively safe system. Nevertheless, as with any other system vulnerabilities exist. It is possible, if someone controls a sufficiently large number of TOR nodes, to reverse the effect of TOR anonymisation. Since TOR is a global peer-to-peer network, such an attack is at least very costly. If one monitors the entry and exit node of a connection, it is possible to reconstruct the traffic, if one uses sophisticated statistical analysis. This great effort will suffice in most cases, an attacker from finding out the hidden identity. The protection of anonymity must not be absolutely given, because even a successful attack can disclose only a short period of data traffic.

Another problem with TOR is that the exit node the data available into plain text. This means that I entrust sensitive data to a node, which I do not know and cannot review. It is therefore absolutely necessary to use TOR only when it is combined with an end-to-end encryption. For example, the Swede Dan Egerstad swindled sensitive data from embassies and governments, because he installed password sniffers on five exit nodes. Among the data collected were 100 e-mail accounts of international embassies. In TOR, multiple TCP connections are sent through the same TOR tunnel. This process is intended to lead to a lower latency. This may have the result that even if all the important or sensitive data is encrypted, you can still find out ones identity. For example, if you build up a secure SFTP connection through TOR, but parallel to that, TOR users surf the Internet unencrypted. Then the exit node can identify them by pursuing the non-secure HTTP communications. One must pay very close attention to how to use the TOR network. If in doubt you could do more harm than good.

1.3.2 Tracking

Even if you use TOR to hide your own IP address and data, you are not automatically anonymous. A major threat to the anonymity, in the Internet is often the communication partner in the Internet, because when you retrieve a service it leaves traces; not just crumbs but entire cookies. These traces are used by website owners to create profiles of users. This is not done in bad faith, but should help to fund the services that are usually offered for free. A website is often financed by advertisement. This advertising will only bring one enough money when it reaches the right person. Therefore, the profiles are used. These profiles determine what preferences a user has and then show him the respective ads. Frequently, profiles of several websites are linked and result in larger and more accurate profiles. There is a risk that such profiles allow conclusions about real people. This is an enormous threat to privacy.

Such cross-server profiles are also called tracking. There are some technical ways to track a user. The simplest and most common is the use of cookies. A cookie is a small text file that is sent when you call up a website in the browser, the browser then stores this cookie and sends it every time the browser calls up the website again. A cookie often includes a unique identification number, with this number the website can determine which virtual person is viewing the website. With this identification it is, for example, possible to create a shopping cart or to personalize a web page for the user. Cookies are able to collect a large amount of information. Because a cookie is sent only to the referring website, it is not possible to track the users across multiple websites. In order to track a user across different servers, the developers use a trick. In the source code of the website, links or banners of a partner's website are embedded. When the browser loads this banner, it sends a cookie to the other website. This is called ”Third party Cookies”.

To avoid this tracking, you have to ban these third-party cookies in your browser. Only the cookies of the actual website remain. In a popular site like Google, corresponding profiles can be created even without third-party cookies. To avoid these profiles , one would have to block all cookies, which then usually means that you cannot continue to use various services. Alternatively, you can delete your cookies regularly, to avoid formation of an effective profile over long periods. Even this effort is in vain, if you log onto the website, since you identify yourself with the registration completely voluntary.

Since you can earn a lot of money with the tracking , other tricks have also been developed. It is possible to identify a user, only with the digital fingerprint of his browser. In other cases you can apply some tricks to foist some kind of cookie to the users. For example, the function of the browser to cache web pages is often misused. The browser stores the downloaded content locally in order to utilize it later on. Before he reuses the content, the browser compares the local and the online versions, with the help of short identifiers. This comparison of the versions can be misused to identify a user.

Each browser has a wide variety of settings. What language is used, what font is preferred, are cookies enabled, what Java scripts and what plug-ins are installed. Most browsers are differ from each other because of these settings. The browser sends all these settings in a header to each requested website to establish the site in a displayable form. This browser header is in many cases so unique that it acts like a digital fingerprint due to the browser. Providers of web sites use this digital fingerprint to link profiles of different websites.

There are other similar security flaws, some are known and appropriate solutions exist for them. Other vulnerabilities have not been discovered yet, but you can almost certainly assume that there are more. Other security issues will surely arise in the future when new protocols have been developed. The problem, which all of these vulnerabilities have in common, is that the user must always become active in order to close them. To close these security holes, there are some very interesting programs.

The CookieCooker is one of these programs developed by the University of Dresden. The CookieCooker allows users to surf with many different identities, so that tracking, using logins and cookies is no longer possible. Therefore, the program has a little trick. Since one does not want to completely give up cookies, CookieCooker receives cookies and then exchanges them with other users. The program mixes the profiles of several individuals, so that the profiles are worthless. The CookieCooker also manages login data. The CookieCooker allows you to automatically register with randomly generated account data online. The CookieCooker then stores the login data and makes them available again. This will enable you to manage practically infinite number of identities.

Another tool is JonDoFox, a plug-in for the Firefox browser. JonDoFox comes from the creators of the anonymity service JonDo. JonDoFox includes some of the just mentioned vulnerabilities, such as the identification over caching. JonDoFox tries to prevent tracking completely. But it can only close already known vulnerabilities. It therefore remains uncertain whether tracking is effectively prevented, but in many case, it is more difficult.

So far there are is no panacea, with which one can guarantee absolute anonymity in the Internet. Some non-technical approaches are designed to prevent tracking and profiling, without being firmly placed on a technical implementation of tracking. These approaches are based on the idea to tell the data collectors, that one does not wish that information is stored. In Firefox there is a setting option, with the name ”Do Not Track”, this should tell all communications partners that you do not want to be tracked. An initiative of some of the merged companies offers a so-called opt-out cookie to set. If this cookie is set, the companies won't create further cookies, and refrain from tracking. More than half of these companies continue to set cookies and more than ten percent of these companies ignore their own privacy policies and continue tracking. A study of the ”Stanford Center for Internet & Society” shows that anti tracking measures appear only on a voluntary basis rather.

1.4 Conclusion

Although it is possible to ensure interactional privacy with the help of a good anonymizer and good encryption it does not apply to informational privacy.

The situation of the interactional privacy has not changed much in recent years. In order to ensure it a reasonable anonymity service and a good encryption is required. Here, encryption and anonymity services are chosen independently, because their safety goals do not interdict . Even with the birth of IPv6 will not change much. It will nevertheless be necessary to anonymize IP address and encrypt traffic.

Even if there are vulnerabilities in TOR, JonDo, IPsec, and other systems, one must appreciate that these vulnerabilities can only be exploited by a professional attack. For example, using TOR and an IPsec end-to end encryption is a very safe method to protect privacy against ordinary threats. Suspicion independent data retention, as implemented in Germany, loses all meaning for someone who locks down his data traffic as described. Also curious acquaintances, colleagues or parents are barely able to circumvent such protection.

The only big problem for the protection of the interactional privacy in the future is the ever-increasing need for security. After the terrorist attacks on the World Trade Center in New York on September 11th, 2001, an irrational fear started to spread out. This fear was, among other things, responsible for the adoption of the ”US PATRIOT Act”. This Federal law restricts some of the civil rights of the United States considerably. Among other things, the protection of the interactional privacy was completely abolished. Under the Patriot Act, it is permissible for the FBI, without a court order. To conduct telephone and Internet surveillance, telephone companies and Internet providers must also disclose their data. The fear even went so far that it was required that service providers should install backdoors in its encryption algorithms, so the FBI can decrypt these. These fears also impacted Germany 2007, so that, at least for a short time, the suspicion independent data retention was introduced. If such fears continue to spread, it can quickly occur, that Internet privacy will be no longer be possible. If all mixes of JonDo can be forced to record and publish their data traffic, this network would be pointless. Private operators of TOR nodes would slip off with the operation of a node in the illegality, which may eventually lead to the collapse of the TOR network. Encryption protocols that should always contain a backdoor would be left no protection of confidentiality. However, it is unlikely that the world evolves to such a state but it would be bad enough if such trends continue only in the United States.

The situation of the informational privacy is more critical. New offers on the Internet have attracted more users. The Internet is now as extensive as never before. The incredible amount of users and the enormous associated purchasing power has the consequence that advertisement is emphasized increasingly more on the Internet. It is also very easy to collect empirical data on the Internet. It does not matter whether advertisers are trying to place their advertising optimally or if politicians want to find out how to win the next election. The effect on privacy is the same. The users not only lose control of the personal data, they also lose track of what personal data is circulating on the Internet. *"The Internet knows more about you than you yourself"* [L.M.][22] Everything we do on the Internet is digital. And everything digital, you can save, copy, transmit, compare, evaluate and delete. As memory costs are close to next to nothing these days, we can save everything.

This flood of data cannot be prevented. It is absolutely impossible to leave no traces on the Internet. But if you want to participate in the virtual social life, even this will not be successful. In the Web 2.0 one does not take part with a pseudonym, but with his real name. And this phenomenon is more prevalent. One uses social networks and rummages on Amazon or is planning events with friends, all online with full name and completely transparent. How could it be different? How could someone speak with his friends or invite them to a party if no one knows who they are? For Smartphone users, it is customary to inform friends where you are. Everything is made public, one is literally an informational exhibitionist. The question now is, can you combine both. Is it possible to create a comprehensive privacy on the Internet where one can act anonymously and unmolested, so that you still can still use the Web 2.0.

Unfortunately it is not enough to flip a lever, to switch these both worlds. When one publishes his likes and desires freely in the Web 2.0, this can turn out to be a calamity in the anonymous world. With modern techniques such as data mining, it is possible to link the profiles of the anonymous world with the profile based on the Web 2.0. If one researched the topic "Privacy" long and extensively enough on the Internet, it isn't a difficult task to detect a connection. The separation between these two worlds is therefore only possible if there are no data collectors in the anonymous world.

Such a separation of anonymous world and Web 2.0 would have several positive effects. The anonymous world could be comparable to the physical privacy that was described at the beginning of this work. The interactional privacy would remain unaffected by this separation. Informational privacy would be guaranteed immediately. In the anonymous world, there would be no personal data to worry about, all personal data would safely buzz around in the Web 2.0 world. There, all the information of users are actively entered with full knowledge that this information becomes public. Each user would have full control to their personal data.

It would be necessary and technically possible to distinguish the two worlds. This requires, from beginning on, anonymous exchange protocols. This protocol could be implemented as an overlay network, so that no new hardware is required. In addition, a number of application protocols would be required. These application protocols would have to all be designed so that no data is subjected to profiling or tracking. Of course, every side would have to be an anonymous process, for example, the payment process.

The content of the Internet would remain unchanged and solely business models need to become adjusted minimally. Personal advertising and profiling would then focus on the Web 2.0 world. Profiling, in the Web 2.0 world, stands not in conflict with the informational privacy, as all available information has been voluntarily released. The services in the anonymous network would then have to resort to other financing methods, such as a monthly fee or non-personalized advertising but they would provide a comprehensive protection of privacy.

Bibliography

[1] Michel Foucault. *Überwachen und Strafen-Die Geburt des Gefängnisses*,Suhrkamp ,Frankfurt M. 1992.

[2] Philipp Schaumann und Christian Reiser. *Die Bedrohung der Privatsphäre (Privacy)*, http://sicherheitskultur.at/privacy.htm , August 2011

[3] Warren and Brandeis. *The Right to Privacy*, Harvard Law Review, http://groups.csail.mit.edu/mac/classes/6.805/articles/privacy/Privacy_brand_warr2.html, August 2011

[4] J. Loughney, Ed.. *RFC 4294 - IPv6 Node Requirements*, http://tools.ietf.org/html/rfc4294, August 2011

[5] Kaushik Das. *IPSec & IPv6 - Securing the NextGen Internet*, http://ipv6.com/articles/security/IPsec.htm, August 2011

[6] Niels Ferguson and Bruce Schneier. *A Cryptographic Evaluation of IPsec*, http://www.schneier.com/paper-ipsec.pdf, August 2011

[7] Tobias Bandh. *Einführung in IP-Sec*, http://www.net2.uni-tuebingen.de/fileadmin/RI/teaching/seminar_iit/ss03/IPSec_Einfuehrung.pdf, August 2011

[8] Dr. Andreas Steffen. *Leichter tunneln - IPSec-VPNs werden einfacher und flexibler dank IKEv2*, c't 20/2007 http://www.heise.de/security/artikel/Einfacher-VPN-Tunnelbau-dank-IKEv2-270056.html, August 2011

[9] JAP. *Projekt: AN.ON - Anonymität.Online*, http://anon.inf.tu-dresden.de/index.html, Abgerufen August 2011

[10] Benedikt Westermann, Rolf Wendolsky, Lexi Pimenidis, Dogan Kesdogan. *Cryptographic Protocol analysis of AN.ON*, http://freehaven.net/anonbib/papers/wwpk2010.pdf, August 2011

[11] Tor Project. *Tor: Overview*, https://www.torproject.org/about/overview, August 2011

[12] Nicholas Hopper, Eugene Y. Vasserman, Eirc Chan-Tin. *How Much Anonymity does Network Latency Leak?*, http://freehaven.net/anonbib/cache/tissec-latency-leak.pdf, August 2011

[13] Heise Online. *Anonymisierungsnetz Tor "abgephisht"*, http://www.heise.de/newsticker/meldung/Anonymisierungsnetz-Tor-abgephisht-173525.html, August 2011

[14] Peter Eckersley. *A Primer on Information Theory and Privacy*, https://www.eff.org/deeplinks/2010/01/primer-information-theory-and-privacy, August 2011

[15] Peter Eckersley. *Web Browsers Leave 'Fingerprints' Behind as You Surf the Net*, https://www.eff.org/press/archives/2010/05/13, August 2011

[16] Philipp Schaumann. *Ihre Datenspuren im Internet*, http://sicherheitskultur.at/spuren_im_Internet.htm , August 2011

[17] Philipp Schaumann. *Die Problematik von Data Mining und Profiling* , http://sicherheitskultur.at/data_mining.htm , August 2011

[18] Heise Online. *Websites hebeln Anti-Cookie-Maßnahmen aus*, Mozilla Labs http://www.heise.de/security/meldung/Websites-hebeln-Anti-Cookie-Massnahmen-aus-1288914.html, August 2011

[19] Michael Hanso. *Thoughts on Do-Not-Track*, http://www.open-mike.org/entry/thoughts-on-do-not-track, August 2011

[20] CookieCooker.de. *Funktionsweise von CookieCooker*, http://www.cookiecooker.de, August 2011

[21] JAP. *Features des JonDoFox Add-on*, Social-inside http://anon.inf.tu-dresden.de/help/jap_help/de/help/jondofox2a.html, August 2011

[22] L.M.. *Das Internet weiß mehr über Dich als Du selbst* , http://www.social-inside.de/si/2011/08/07/das-Internet-weis-mehr-uber-dich-als-du-selbst, August 2011

Chapter 2

IDS in MANETs

Christian Marciniak

MANETs (mobile ad-hoc networks, without fixed infrastructure) are used in more and more scenarios. Therefore the security has to be taken into account. An aspect of security in networks are IDS (intrusion detection systems). These systems are well established in wired networks. But combining IDS and MANETs is a relatively new aspect in research. Because of the lack of fixed security points like firewalls, IDS are very important in MANETs.

This paper first introduces the main concepts of IDS and MANETs. Then some solutions, that have already been researched, are presented. From these solutions derive some requirements to the software and the architecture of the systems in future use, that are described later. They will lead to more secure MANETs in the "Future Internet". At last, there is an outlook to the future, how IDS in MANETs should be established.

Contents

2.1 Introduction

The internet is becoming more and more an “internet of things”. Every device in our environment is becoming connected to the internet. Furthermore, the devices are connecting with each other. Sometimes these connections use some fixed infrastructure points, sometimes not. If there is no infrastructure point, we are most likely talking about a Mobile Ad-hoc Network (MANET). These networks are the new infrastructure, everyone thinks about when talking about the “future internet”.

Yes, it is very useful to connect devices without given infrastructure. Yes, especially in military or rescue scenarios these networks are the only possibility to establish a connection between different devices. But because of the loose connection of the nodes and the lack of the possibility to determine, whether a node should be in this network or not, these networks are a highly attractive aim for attackers.

In wired networks, there has been very much research about so called Intrusion Detection System (IDS), as a “second line of defence” (behind firewalls). These systems detect attacks and usually initiate actions, to prevent further damage. But many of the well established techniques of IDS in wired networks are not suitable for the mobile networks, we are talking about in this paper. IDS in MANETs have to be treated in a different way.

This paper is structured as follows. In section 2.2 on the following page I will introduce IDS and tell about the different types and methodologies used. Section 2.3 on page 34 introduces MANETs and describes the threats to them. Furthermore this section describes the problems, that arise, when installing an intrusion detection system in a mobile ad hoc network. Next, in section 2.4 on page 38, I will give an overview about current solutions from researches, that are supposed to solve the problems. Section 2.5 on page 50 describes the requirements for future MANETs, that will provide good support for intrusion detection. In the end, section 2.6 on page 54 gives a summary of this paper.

2.2 IDS

In this section I will tell about IDS. This includes how they work and what types there are and what methodologies are being used.

2.2.1 Definition

According to [5], an IDS is a (software-) system, that monitors events in order to detect incidents. In this context, an incident is a violation to the security policy of the system. The software automates the intrusion detection process. An important component of an IDS is the logging facility, which is used to analyse the events or even to detect unusual activity by the user. When an IDS is capable of preventing attacks, it is often called IDPS (intrusion detection and prevention system), but mostly "IDS" is used as a generalisation of both functionalities.

An IDS uses three main functions. First of all, it records information about what is happening in the network. Next, the system will notify the administrator, if something unusual happens or if human consultation is needed. The third component is the report component, which generates assessments of the situation in regularly intervals. These reports are very important for arguing with the management about the security situation of the company. In addition to that, these reports give the administrator a feeling for the situation and are an advice, whether he should be more careful ore not.

2.2.2 Types

There are basically two types of IDS - network based IDS (NIDS) and host based IDS (HIDS) as described in [13]. NIDS are deployed on routers or other network infrastructure devices. This technique is almost invisible for an attacker and therefore very effective. In addition to that, the nodes do not have to spend CPU capacity for detecting intrusions. Distributed network attacks can be detected, too. But this technique also has some drawbacks. First of all it is impossible to protect a specific node. Next, if the communication is encrypted, there will be no chance to detect an attack within the packets. Another drawback is the lack of the capability to inspect high traffic with a high packet rate. The current systems are not capable of computing so much actions, so there are lots of packets not recognized by the system.

A HIDS is a software system that recognizes attacks on the application or operating system layer. Therefore specific reactions of the specific node can be detected and reported. But these systems have to be computed on the current node. So they reduce the available CPU capacity and - in our case - the battery life. In addition to that, these systems are not invisible to an attacker, which may let him modify his attack in order not to be detected. Another important drawback is the cost factor. These systems have to be very specific for each system. Because of that, they are very expensive.

2.2.3 Methodologies

The “Guide to Intrusion Detection and Prevention Systems (IDPS)” [5] speaks about three main methodologies of intrusion detection. At first there is the signature based detection. This kind of detection is good for known attacks that can be described in a signature. The detection of unknown attacks on the other hand is almost impossible. But, this is a very fast and simple way of detection, that can be easily implemented. Another drawback is the missing capability of remembering previous actions. This means, that a signature based detection is not capable of correlating some actions in order to detect an incident. In addition to that, the size of the signature database is important. It can get very big and therefore difficult to handle.

The next technique mentioned is the anomaly based detection. At first, there is a mechanism to collect data in order to define the “normal” traffic of the network. Then, when the system is activated, it detects differences to the normal traffic. If these differences become greater than a predefined threshold, the system will alert the administrator. It is even possible to change the “normal” traffic during time, but this opens the system to the so called evasion attack, where the attacker slowly increases his attacks in order to higher the threshold. In general, anomaly based detection is very good at detecting unknown attacks, but has a high false positive rate for unlikely benign events.

The third technique is called stateful protocol analysis. It uses deep packet inspection. Therefore it can pair the request and response messages of the different protocols used in the network. But there are other use cases. For example a command could have an argument, which usually has a length of 20 characters. In this case, an argument with a length of 1000 characters would be very suspicious. This methodology works only on layer 4 of the Open Systems Interconnection (OSI) model, which decreases its effectiveness. The greatest problem of the stateful protocol analysis is the resource intensity. Because of this, it is only implemented in a few IDS.

In [13] the authors speak about a fourth technique called “honeypots”. These are dedicated systems that claim to be productive and highly important, but are not. They are only used to learn new attack types in order to improve the “real” IDS. The honeypots can only be used to learn attacks, but not to prevent attacks. In addition to that, they are not usable to learn attacks from insiders, who know, these systems are not productive.

The authors of [14] name two more methodologies: AI (artificial intelligence) based detection and detection based on statistical data. AI based detection is used to support human intuition, but has a high false positive rate and is not commercially available. The detection based on statistical data first defines statistical values (arithmetic average, variance, etc.) for specific actions (time of use, length of use, etc.). Then the system checks, whether the current situation is significantly different to the “normal state”. In contrast to the stateful protocol analysis, the statistical detection can be used in lower layers than layer 4. This methodology is not commercially available either.

[14] categorizes the named methodologies as follows: The two main categories are pattern based detection and anomaly based detection. Pattern based detection means the detection based on signatures, whereas anomaly based detection includes protocol analysis, detection based on statistical data, AI based detection and honeypots.

2.2.4 Summary

In this section, I introduced IDS. These systems are used to detect attacks on a network. Furthermore, some IDS are able to start damage preventing actions. I discussed the two types of IDS - network based systems, that work on routers or other network infrastructure, and host based, that work on the nodes in the networks. Moreover I described the methodologies used. There are signature based detections, that rely on predefined attack signatures, anomaly based detections, which detect differences in network traffic to previously learned states and stateful protocol analysis, which checks, whether the protocol messages are in correct order. Honeypots are used to learn new attacks by simulating a worthy node. Furthermore there are AI based detection systems and detection systems, that rely on statistical data; but these two systems are not commercially available. The list on the next page shows the main facts of IDS.

Main facts of IDS

- main functions
  - information recording
  - notifying administrators
  - generating reports
- types
  - network based
  - host based
- methodologies of detection
  - signature (pattern) based
  - anomaly based
    - stateful protocol analysis
    - honeypots
    - AI based detection
    - detection based on statistical data

2.3 MANETs

In this section I will characterize MANETs. Then I will give an overview about the threats against them and tell about the difficulties compared to wired networks.

2.3.1 Definition

Let us try to analyse the components of the acronym MANET with the use of [9]. At first, there is the “mobile” component. That means that routers and nodes in such a network are mobile and lack a determined position.

Next is the “ad hoc” component. In a MANET you do not have a fixed infrastructure to which you could connect. Because certain infrastructure components as routers can move along with the nodes, there is nothing you can rely on. So nodes are connecting spontaneously with each other in order to form a network. There is even the possibility that no fixed infrastructure is available so that the nodes must do tasks like routing and address management on their own.

Because of the attributes described above there arise some defining characteristics of MANETs. These networks are bandwidth-constrained, because wireless links still have a lower capacity than wired links. Furthermore you have to take care of the energy management of all components. These networks are used in environments that usually have no energy supplier. Because of this the nodes have to rely on their batteries which rules out CPU-expensive routing algorithms or other complex activities. Last but not least, MANETs have highly dynamic topologies because of the movement of the nodes.

2.3.2 Threats

As said in [9], “mobile wireless networks are generally more prone to information and physical security threats than are fixed, hardwired nets.” When looking into [5] this becomes more clearly. An attacker only needs to get in the transmission range of a node to sniff data from it. This is much more likely than getting physically into a wired network. Furthermore you can easily fake your identity and start a Man-in-the-Middle-attack in order to sniff all the traffic directed to or from the nodes next to you. This is because most wireless networks lack a strong authentication process, which would make these attack more difficult.

In [6] there are more types of threats mentioned. These include the battery exhaustion of a neighbour node by making it compute new routes or flooding it with packets. This is a certain wireless attack, that is not possible in wired networks. Moreover a node simply can falsify a route in order to route all the traffic through itself. The difficulty is, that this behaviour is not directly noted by all other routes and hard to detect.

More threats are described in [2]. First of all friendly nodes can be possessed by the enemy. So - from one moment to the next - a trustworthy node can become malicious and start an attack on the network. Next friendly nodes can be infected by viruses, worms or other malicious software. This means that these nodes can either be unable to work properly or they work in a malicious way, that has to be detected. In case of a connection to a reach-back network there can be corrupted nodes in this network. This would mean that the link to the organisational base would have been corrupted. When thinking of usage scenarios of a MANET, like military (as shown in figure 2.1) or catastrophe response, this could have high impacts on the operation.

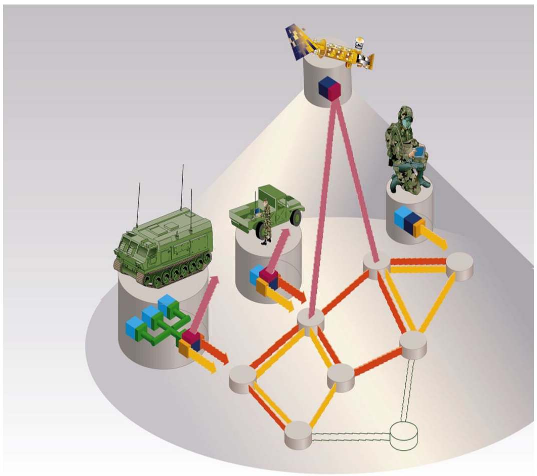


Figure 2.1: Example scenario for a MANET (see p. 7 of [9])

Another paper describing threats to MANET is [12]. The authors think, that the auto-configuration of MANETs is a problem, because it is based on a concept of total cooperation. That means all the algorithms in MANETs rely on the trust to other nodes. Therefore it is very easily seen, that this arises desires in attackers. Furthermore mobile wireless networks do not have a fixed entry point like wired networks, such as a firewall. This lack of control and ability to filter incoming packets gives attackers more possibilities to hack into the network.

2.3.3 Difficulties

After describing some threats connected to MANETs, I will now explain, why it is so hard to fight against them, using [4]. First of all there is the limited battery capacity. Therefore any IDS needs to save as much CPU-power as possible. This of course leads to the problem of finding efficient algorithms, because very complex algorithms cannot be executed on mobile nodes.

Furthermore there is the problem of the easiness of use. An IDS should have a very low false negative and - may be sometimes more important - a low false positive rate. In this context, a false negative is an attack, which is not detected by the IDS; a false positive is a benign action, that is considered harmful (sometimes called false alarm). When an IDS is having a high false negative rate, it is useless. On the other hand - when the false positive rate is high, the system administrator would probably ignore all attacks, because he thinks, they are false positives. This makes the IDS useless again. The problem here is, that the two rates are usually connected to each other. A low false positive rate means a high false negative rate and vice versa.

The next problem is the Interoperability. Because of the heterogeneity of the nodes in a MANET, different IDS could be installed on them. So these different systems have to interact in an effective and efficient way. But even if there is only one type of intrusion detection system, it will likely be deployed on all nodes at the same time. Therefore the nodes need to collaborate in order to produce useful results, which, in this case, means the detection of attacks.

Other problems, as stated in [2], are for example the limited bandwidth. The sent data must be reduced as much as possible for two reasons. First of all, the transmission channel should be free for "really important" messages (think of the usage scenarios). Next, every received packet must be computed, which again costs battery power. Because of this, an IDS has to send as few packets as possible with as much information in them as possible.

The authors also mention the problem of dynamism and mobility. The decentralized infrastructure of a MANET makes it hard to identify neuralgic nodes, that should be watched carefully. In addition to that, MANETs lack a fixed traffic concentration point. Such a point would make the intrusion detection much more easy. In a MANET there is no node, which receives every packet. This again leads to the need of having parts of the IDS on many nodes in the network.

Because of the need of having a part of the IDS on many nodes, the problem of event correlation arises. The nodes not only have to cooperate in the meaning of sending data to each other, as mentioned above, but they must correlate recognized events. It is possible, that a "harmless" event on one node in correlation with other "harmless" events on other nodes in the network can mean, that a serious attack is going on. The problem of detecting these attacks is not easy to solve.

2.3.4 Summary

MANETs are networks, that are built by wireless nodes without any network infrastructure. Important characteristics are the limited battery capacity and the low CPU power. A great security threat is the easy physical access to these networks. Therefore sniffing becomes a piece of cake. Another threat is the rely on self-configuration, which needs cooperation between all nodes and makes it easy for attackers to infiltrate the configuration processes. IDS are difficult to implement in MANETs. Reasons are the limited battery and calculating power, the heterogeneity of the nodes and the limited bandwidth. The lack of a traffic concentration point brings further problems. At last, I mentioned the problem of the event correlation.

Main facts of MANETs

- facts
  - mobile nodes
  - dynamic topology
  - no fixed infrastructure
  - bandwidth and CPU constraints
  - limited battery power
- threats
  - easy physical access
  - easy faking of identity
  - battery exhaustion
  - falsifying routes
  - selfish and malicious nodes
  - rely on cooperation
- difficulties
  - limited battery capacity
  - low false positive rate
  - interoperability
  - limited bandwidth
  - dynamisms and mobility
  - event correlation

2.4 Solutions

This section will describe some solutions to the problems named in section 2.3.3 on page 35, which already have been researched. These solutions mainly rely on distributed architectures.

2.4.1 Cooperative IDS

The first solution, I want to describe, is a cooperative intrusion detection architecture as proposed in [2]. The authors propose an architecture, that collects data bottom up and makes decisions top down. Some main facts of this architecture are described in the following paragraphs.

The architecture described in the paper is called "dynamic hierarchy". This means that there is no static configuration needed, but the nodes negotiate the hierarchy. The connectivity (to how many other nodes a node is connected), processing power, storage capacity and proximity (the first higher layer should only be one hop away) are taken into account. When the hierarchy is built, the nodes report along the hierarchy. They try to detect intrusions on the lowest possible layer in order to minimize the latency and communication overhead. An example hierarchy is shown in figure 2.2.

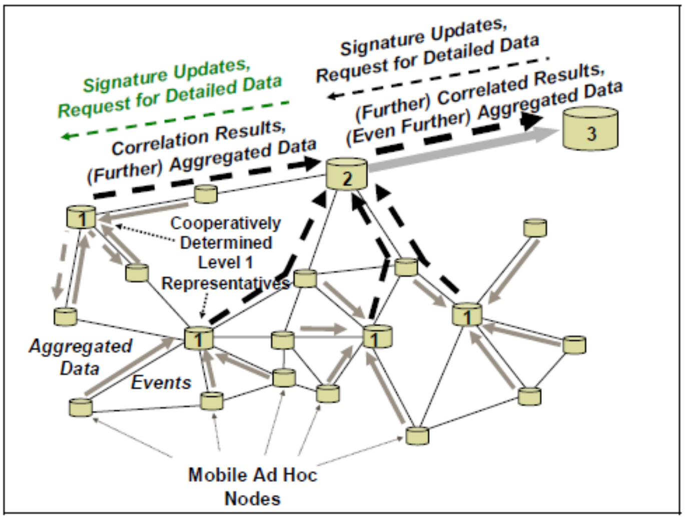


Figure 2.2: Architecture of the cooperative IDS (see p. 4 of [2])

The IDS uses two monitoring systems. First of all, it uses promiscuous monitoring. That means, that the local agents monitor the whole traffic within their range. Because there could be problems (like the hidden or exposed station problem), this technique is not sufficient. In order to validate the data, the agents also report statistics about how many packets they received and sent to their head node in regular intervals. This is less bandwidth consuming than reporting the payload of every packet.

Monitoring end-to-end-traffic is very difficult. The nodes would loose their battery power too quickly if they would monitor and inspect every flow they transmit. Furthermore every node would have to have the encryption key in order to inspect the packets and search for manipulations. The authors solve this problem with a trick. They let only the first and last hop monitor the traffic. In case, there would be a flow from A to E via ABCDE, B and D would have to monitor the traffic. In case the route changes, the responsibilities for monitoring would change automatically. This technique reduces bandwidth consumption, increases battery life time and reduces the key exchanges.

The authors claim, that their architecture is good for detecting intentional data packet dropping. The aggregation of the packet statistics of neighbours would reveal the lost packets, because the nodes log all packets that could be forwarded and include these data in the reports. Furthermore, man in the middle attacks against routing protocols should be easily detected, when counting the packets on the different nodes. This way, the origin of bad route request packets can be identified and further actions can be performed. At last, the authors claim, that their architecture could detect attacks to OSI layer 2 and 3, because of the traffic reports of the nodes. The aggregation of these reports reveals e. g. Denial of Service (DoS)-attacks.

Main facts of the cooperative IDS

- data collecting bottom up, decision making top down
- dynamic hierarchy
- promiscuous and statistical monitoring
- last-hop-monitoring

2.4.2 Lightweight IDS agents

A common architecture is the client-server-model. But, according to the authors of [7], this architecture has a big scalability problem. If there are too many clients, the server will run out of CPU capacity. Because of this, the authors propose a manager-agent-model with lightweight agents in order to achieve the scalability. The agents used are called “lightweight”, because they use minimal code. Therefore the agents are much easier to transport, simpler and dynamically updatable and upgradable. For example, if there are different OS used in the network, an lightweight agent will only need the detection engine for one specific OS instead of having the capability of detecting intrusions in all possible OS.

Another interesting feature of their agents is the sensitivity level. Depending of that level an action could be considered either benign or harmful. Because the agents are able to dynamically aggregate data they collected from other agents, this level can change immediately. For example, think of failed login attempts to a system. In a normal situation some failed attempts could be caused by mistyping of the authorized user or because he mixed up some passwords. But if the agents detected a portscanning short time ago, then failed attempts could be related to the scanning attack and could be considered harmful. This technique also reduces bandwidth consumption, because the agents can remain small, when no intrusion is present. This reduces the CPU load, too.

In addition to that, the mobile agent system has no vulnerabilities in credentials, because the agents carry these information within them. There is no need to connect to a server in order to achieve the intrusion detection methods. When an agent travels to a node, it carries all needed functionality and information with it, so no possibly insecure authentication process is needed.

The lightweight agent architecture uses a mobile agent platform called “Voyager”, which is capable of dynamically upgrading the agents. Because this platform and the whole IDS are written in Java, it is almost platform independent, which means, that only the lowest detection level needs to be platform dependent in order to correctly detect intrusions. Because of the strong use of interfaces and other Java features, the proposed architecture is easily expandable and reusable.

There are some other components worth to be mentioned here. First of all there are the data mining agents, that are used for collecting data from the systems in order to create rules for the intrusion detection. The next component is the data warehouse. This is a global database, that includes all collected data. This is useful for discovering new attacks and identifying weaknesses in the network.

Main facts of the lightweight IDS agents

- minimal code
- sensitivity level
- dynamical upgrading of agents

2.4.3 Application-Layer IDS

Another interesting try to establish a good working IDS in MANETs is described in [1]. This architecture, seen in figure 2.3 on the facing page, also uses local agents for the intrusion detection process. Here, the agents consist of three different components. First, there is the monitoring and detection component. This component collects activity data and system calls in order to compare them with the profiles stored in the local database of each node. The response component formulates a response to the intrusion, in case the detection component has detected one. If the detection was based on anomalies, then the mobile agent requests another agent from the server in order to perform further analysis on the attack. The last component is the secure communication component, which is used for the communication between the mobile agent and the server and with other nodes.

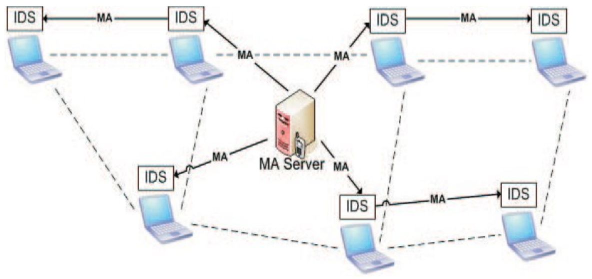


Figure 2.3: Architecture of the application-layer-IDS (see p. 3 of [1])

The mobile agents in this architecture use both detection techniques, signature based and anomaly detection. The signatures in this model are behaviour based. This means, that they contain unallowed sequences of instructions of the programs. These signatures are smaller than usual signatures. As said above, when an intrusion is detected by anomaly detection, another agent is requested to perform further analysis. After the analysis a new signature is created and stored in the local database and will be delivered to the other nodes.

Moreover there is another interesting function of the mobile agents. Periodically, the agent server sends verification agents to the nodes. These agents are used to determine the integrity of the nodes. This function assures, that no node can be compromised without knowing of the server. When a compromised agent is detected, it is either updated or, if not possible, shut down.

Because the mobile agents are not bound to a specific node, but travel around, the server has to determine how many agents are needed in the current network situation. If there are too many agents, they will generate too much traffic in the network for travelling around and exchanging data with each other. Furthermore, some nodes would be visited more than one time from each agent. On the other hand, if there are too less agents, the few agents will need more time to visit other nodes. This will decrease the detection rate and will lower the accuracy of results, because the data aggregation will not be sufficient.

A drawback of this architecture is the need of an offline certification authority. The mobile agent server and the nodes need certificates to establish the secure communication and to authenticate against each other. This leads to two situations. Either the certificates are installed on each node, that possibly can have access to the MANET or there are installed on demand. The first version would mean, that no node could connect to the MANET, even if it was benign. The last version would mean a new vulnerability to the system, which should not be.

Main facts of the Application-Layer IDS

- signature and anomaly based detection
- requesting another agents for further analysis
- verification agents
- problem of finding right amount of agents
- need of an offline certification authority

2.4.4 Watchdog and pathrater

An interesting technique, that could be part of an IDS is described in [8]. The authors do not claim, that their technique is part of an existing or upcoming IDS, but it could be perfectly used for one. The only drawback of this technique is the requirement of a source routing protocol, as described below.

The authors tell about different kinds of nodes, that do not work as supposed. There can be overloaded nodes, that do not have the capacity to work well, e. g. forward packets to other nodes. Then, there can be selfish nodes, that are unwilling to spend their battery power and CPU capacity on packets, that do not directly belong to them. The third group of bad behaving nodes are the malicious nodes, which have become victims of an attack. The last group are the broken nodes, that are shut down and are no longer able to work.

In this paper, there is written, that an a priori trust relationship on the one hand would be one possibility of securing the communication between the different nodes. On the other hand, this technique has some drawbacks. First of all, there still can be compromised or broken nodes, which are not able of forwarding packets properly. Furthermore, untrusted nodes, that would work well and benign, are excluded from the routing process, because they are not in the relationship. Therefore, the authors claim, that other techniques have to be used.

The main problem of most routing protocols is the fact, that the protocols assume all nodes behave well, which in reality is quite unrealistic. Because of this, the authors describe two add ons for routing protocols: a watchdog and a pathrater. The watchdog identifies misbehaving nodes by promiscuously listening to the neighbour nodes. It overhears, whether its neighbour transmits the sent packet or not. If not, a counter for lost packets is increased until a threshold is reached. If the threshold is reached, the source would be notified, that the neighbour is misbehaving. Then, the source can choose another route to its destination. Although, the watchdog has some disadvantages. There can be collision on the watchdog node, that make it impossible to overhear the correct transmitting of its neighbour. Furthermore there can be collisions on the receiver node. In this case, the watchdog would assume, the packet is transmitted correctly, but it is not. In addition to that, all nodes need to have the watchdog installed and need a great transmission power in order to let all of their neighbours overhear their transmissions.

The second component of this architecture is the pathrater, which runs on the source. The pathrater collects data from all nodes about their neighbours. The nodes start at a neutral value and increase this value for correct transmissions. If a node is misbehaving, its value is highly decreased. The pathrater tries to find a route to the destination only by using nodes, that have positive values. When no route without misbehaving nodes is available, the pathrater sends a route request, in order to find a new route to the desired destination.

The authors claim, that their architecture increases the network throughput, because misbehaving nodes are consequently ignored in the routing process. On the other hand, the overhead is increased, because the pathrater often sends route requests. In addition to that, there are often false positives, because of collisions either on the watchdog node or on the receiver nodes, which lead to many pathes, that include misbehaving nodes. This increases the frequency of the pathrater of sending route requests. Although, this leads to fresh routes for the most time, which is - especially in case of MANETs - very useful.

Main facts of the Watchdog and pathrater approach

- watchdog overhears whether neighbours transmit packets
- pathrater tries to find routes without misbehaving nodes
- increases network throughput
- high false positive rate, increasing overhead
- source routing protocol needed

2.4.5 Detecting malicious nodes

[3] describes a technique for detecting malicious nodes. The technique can be used in all networks, that have more than 5 ($n > 4k + 1$) nodes, i. e. in almost every MANET. The so called "Adcli"-algorithm works as follows.

One node sends a message to all nodes in its radio range. The benign nodes are supposed to answer, whereas the malicious node would not answer. This assumption has to be made in order to execute this algorithm. Furthermore the message should be a regular, unsuspicious packet, that does not reveal its purpose in order to hide the detection process from the malicious node.

The initiating node then requests a malicious node vote by sending a specific message. At this moment, the malicious node starts to recognize, that a detection is going on, but it is already too late for it to change the result of the vote.

After collecting all votes, the data is aggregated in order to proof, that the one node is malicious. Because there can be false votes or even unreceived nodes, the suspicious node will be declared malicious only if at least $k + 1$ nodes vote against it.

This algorithm has the advantage, that the detection process is spread to multiple nodes instead of having one single node for the detection. This reduces the needed CPU power and the complexity of the algorithm. Because the detection process starts randomly by different monitor nodes, the detection rate is quite high, because the malicious node has no clue, when a new detection will begin. On the other hand, this algorithm produces network overhead, because of the voting process.

Main facts of the detection process with adcli

- adcli algorithm
    - sending messages
    - benign nodes answer, malicious do not
    - malicious node vote request
- detection spread to multiple nodes

2.4.6 A court based IDS

A so called court based IDS is proposed in [10]. This system models courts with their main components in order to identify malicious nodes. The whole system is based on clusters, i. e. one hop clusters, that are voted in a secure way. Therefore, the authors call their system “CCIDS” - court-like cluster-based IDS. The cluster head is voted regularly in order to avoid one node domination.

The system consists of different components. The first component, almost every IDS has, is the monitoring component. This module monitors per hop behaviour as well as per cluster behaviour.

The next module is the accusation module, which accuses a node of malicious behaviour. Therefore it sends messages containing the ID of the suspected node, the supposed attack type and a timestamp of the accused attack.

Moreover, there is the arbitration module. Its purpose is to investigate and analyse the defence. In the real world, you would call it the judge. The investigation messages are sent with a timestamp, too, so that the accused node can look into its history and build its defence using the defending module.

The alert issuing module is used to generate alarms, if a node is declared malicious. Every node, that receives the alarm uses the alert checking module to check, whether the alarm comes from the cluster head, which is the only node allowed to send these messages. In addition to that, every node checks, whether itself is blacklisted. If this is the case, the node will start its defence.

In case there are too many alarms defended, a re-election of the cluster head is avoided or a de-election is initiated in order to defend malicious alerts. There are other mechanisms to reduce false positives, too. First there is the local accusation filtering. This means, that there is a local analysis before generating an accusation. This lowers the detection rate a bit, but also lowers the false positive rate, which is a fair trade off. Furthermore, there are lawyers. These are nearby nodes that can defend other nodes. They use their own history for the defence and check, whether they recorded the malicious activity, too.

The authors claim, that the detection rate is around 90 % in their setting and the false positive rate is kept between 0.1 % and 0.8 %. The communication overhead is slightly low, which is, in terms of the authors, a proof for the scalability of this system.

Main facts of the court based IDS

- models courts from reality
- accusation module
- arbitration module
- possibility of lawyers
- history on every node

2.4.7 Random walker

The paper [11] starts with a summary of the weaknesses of previously invented IDS in MANETs. The hierarchical model, for example, adds a process of finding the cluster head, which reduces battery power and consumes CPU capacity. Furthermore the clusterhead has a much higher workload than the other nodes in the cluster, which is quite unfair. In addition to that, these systems are prone to the so called Byzantine attack, where malicious nodes vote another malicious node as the clusterhead.

Moreover the authors claim, that signature based IDS produce network overhead for distributing the signatures. Malicious nodes can falsify the signatures, too. These nodes also can slowly higher the thresholds of anomaly based IDS, which is why the authors propose the specification based detection. The specifications need to be established manually, but they can be reduced to a few most used protocols on the OSI layers 2, 3 and 4.

The agents of this IDS use a technique called “random walker”. The agents travel around in the network using a path of random steps. This is like exploring a graph randomly without knowledge of the topology. This adds only little overhead to the network traffic, because the agents only need to travel around. There is no need for calculating the next steps properly or considering the whole system. Moreover, these technique is stable for changes in the topology of the network, because the agents do not carry any information about the topology with them.

A random walker consists of different modules. First of all, there is the migration module, which is used to travel from one node to another. It connects to the docking service module of the next node via a secure channel, which is secured by Advanced Encryption Standard (AES) over elliptic curve Diffie-Hellman. The detection module or engine is a simple multi-layer detection. It can be so simple, because the agent is directly on a node and uses specifications, which are modelled using finite state machines (FSM). An example FSM is given in figure 2.4. The agents stay for different times on each node, depending of the pre-defined worth of a node. Next there is the replication module, which is used for replicating, i. e. sending away, a node. This is probability-based. Of course there is a response module, which starts actions, in case there is an attack detected.

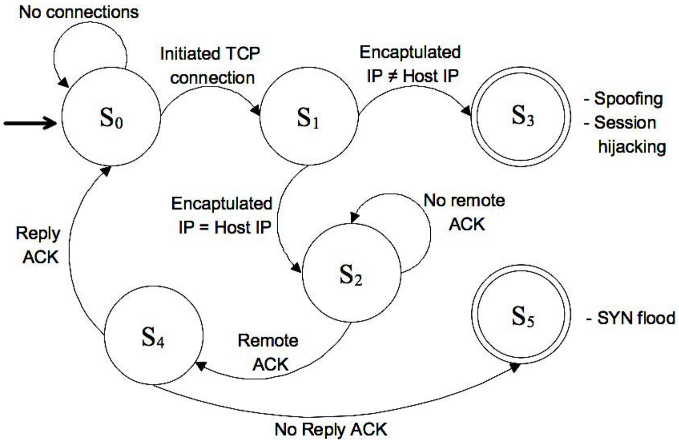


Figure 2.4: Example FSM for the random walker (see p. 7 of [11])

In this paper there is written, that the used technique is capable of detecting new attacks while using less CPU power. In addition to that, there is no need to rely on the cooperation of the other nodes, because every agent works on its own. This stabilizes the system in case of topology changes. There is a low false positive rate, too. But there also is one problem. It is hard to find the right amount of random walkers for the network like described in section 2.4.3 on page 40.

Main facts of the random walker approach

- specification based detection
- agents travelling around randomly
- stable to topology changes

2.4.8 Summary

In this section, I presented some previously invented IDS techniques, that work fine in MANETs. There is the cooperative IDS, that builds up a dynamic hierarchy. The system relies on the cooperation of every node and presents the last-hop-monitoring for reducing CPU consumption on the nodes.

The lightweight IDS agents are based on a platform, that enables the agents to be dynamically upgraded. Therefore, the travelling agents need only minimal code. Furthermore the authors presented the sensitivity level, which brings a new dynamism to the detection process.

When discussing the application layer IDS, I wrote that this system uses both signature and anomaly based detection. When there is the suspicion, that an attack is going on, more agents for further analysis are requested. In addition to that, there are verification agents, that check the integrity of the nodes.

The watchdog and pathrater approach was discussed next. The watchdog overhears the transmissions of its neighbours and rates the nodes. The pathrater tries to find routes without misbehaving nodes by summing up all values of the nodes on a path to the destination. The biggest drawback of this solution is the need for an source routing protocol in order to work properly.

Section 2.4.5 on page 43 discussed the so called “Adcli”-algorithm, which is used to identify malicious nodes by sending messages and waiting for the responses of all nodes in the radio range. The misbehaving node is supposed not to answer, which is used in a malicious node vote request to spot the malicious nodes.

Next was the court based IDS which models a court from reality. An accusation made by one node is inspected properly and the accused node is given the chance to defend itself or with the help of lawyers. Therefore every nodes logs its history in order to be able to check for previous events.

The last attempt discussed was the random walker. It uses specification based detection, because of the chance of detecting unknown attacks. Because of the fact, that the agents travel around randomly, this technique is stable to topology changes.

Table 2.1 on the facing page shows the pros and cons of the different solutions discussed in this section.

Summing up the pros listed in table 2.1, I come to the conclusion, that all authors claim to have a well working IDS. All systems have different advantages, that cannot be easily compared, because of the great differences between the systems. The combination of the advantages and the conclusions for future MANETs and IDS will be given in section 2.5 on page 50.

On the other hand, there are some similarities in the disadvantages of the systems. For example, many systems produce a great network overhead, which consumes battery power and blocks the transmission channel. In addition to that, some systems require a specific infrastructure, that often needs to be installed previously, which foils the idea of MANETs. Conclusions made of these cons are written in section 2.5, too.

Table 2.1: Evaluation of the inspected solutions

solution	pros	cons
cooperative IDS	dynamic hierarchy statistical and promiscuous monitoring monitoring end-to-end traffic partly	bandwidth and CPU power consuming hierarchy negotiation no full end-to-end monitoring
lightweight IDS agents	scalable, expandable, reusable easy agent transport sensitivity level for dynamic detection dynamic upgrading of agents	need of upgrading for full functionality vulnerabilities of the Java virtual machine need of global database
application-layer IDS	signature and anomaly based detection more agents for further analysis small signatures use of verification agents	more agents needed for specific detection great network overhead problem of finding right amount of agents need of offline certification authority
watchdog and pathrater	increases network throughput value based system no agents needed every note participates in the detection	requirement of source routing protocol prone to collision problems high false positive rate network overhead
Adcli-algorithm	detection process invisible for nodes detection spread to multiple nodes high detection rate	malicious nodes must not answer network overhead
court based IDS	cluster head voted regularly possibility of defence high detection, low false positive rate	need of local history on every node detection engine needed on almost every node complex engine on cluster head needed
random walker	specification based approach stable for changes in topology high encryption standard low CPU consumption	specifications manually established no assurance that every node is visited problem of finding right amount of agents specifications only for few protocols

2.5 Requirements

This section will name some requirements for MANETs in the future in order to ease the use of IDS. They all derive from the solutions described in section 2.4. I divided the requirements into three sections: Architecture, Protocols and Functions.

2.5.1 Architecture

Using the watchdog approach of section 2.4.4 on page 42 leads to the requirement for almost equal nodes or at least equal transmission power. This is needed, because the nodes need to overhear the transmissions of their neighbours in order to determine, whether they behave benign and forward the packets or not. In case the nodes cannot hear the transmissions of their neighbours, the neighbours would be considered malicious and the watchdog would not work properly.

An interesting requirement arises when using the court based IDS of section 2.4.6 on page 44: The need of spending battery power for other nodes. Defending a neighbour node requires searching the history for suspicious actions and sending several messages to the "judge" node. Because we are in MANETs, spending battery power for other nodes is not seen very often in other systems, because the limited battery power is one of the most important constraints in MANETs (see section 2.3.3).

There is the need for an upgradable agent platform, in case you want to use the lightweight IDS approach of section 2.4.2 on page 39. The standard agents are very small and do not have a rich set of functionality. Therefore they need to be upgraded on demand, when they are already on the node. So there has to be a system that delivers these upgrades to the agents and coordinates the upgrades.

Using the random walker approach of section 2.4.7 on page 45 leads to the need of pre installed modules on the nodes in order to be able to integrate agents. Such modules are needed whenever a system is used, in which the agents travel around and only spend some time on the nodes. From this need derives another one. All nodes, that want to be part of the MANET have to be pre-checked and have to pre-install all the needed modules in order to take part in the network. So these networks are no longer that spontaneous.

The court based IDS of section 2.4.6 and all other systems, that detect nodes collaboratively and not by using one master node, need a software on the nodes, that is capable of detecting attacks and identifying malicious nodes. This software has to be pre-installed, too. Moreover this software needs very efficient algorithms in order to save as much battery power as possible while detecting attacks.

The lightweight approach of section 2.4.2 needs a centralized database for collecting data, storing signatures and other. This database has to be on a node, that is in close range to all other nodes in order to minimize network overhead. In addition to that the node that carries this database has to have a high battery power, because storing all this data and processing all the requests from other nodes is very power consumptive.

2.5.2 Protocols

Installing an IDS using the random walker approach named in section 2.4.7 on page 45 leads to a reduced amount of protocols used in the MANET. Otherwise it would not be possible to create specifications, the base of this approach, efficiently. In addition to that, reducing the amound of protocols will lead to smaller agents, because they only need to support fewer actions, and to easy configurable firewalls, because there are not many possible actions to think about.

Speaking about protocols, in case you want to use the watchdog approach of section 2.4.4 on page 42, you will have to use a source routing protocol. This is because the pathrater needs to calculate routes from the source of a packet to its destination. Furthermore the watchdog has to notify the source, that a node in the route is misbehaving. This is only possible in source routing protocols.

When thinking about using the cooperative IDS described in section 2.4.1 on page 38, you come to the conclusion, that the routing protocol has to have the capability of dynamically order nodes to monitor flows. This capability is needed, because of the last-hop-monitoring of this approach. In case the route changes, the duty of monitoring the flow has to change to another node. This must be included in the routing protocol.

2.5.3 Functions

There are some functions, the nodes must provide, when using the previously described solutions. E. g. when using the cooperative IDS of section 2.4.1 on page 38 the nodes have to monitor their status, such as remaining battery power, CPU capacity, connectivity, proximity and other. This is used by the other nodes to decide, which node should be the clusterhead.

Furthermore, the nodes have to have a mechanism of calculating their actual sensitivity status and to propagate it to managers and agents, when using the lightweight IDS of section section 2.4.2 on page 39. This is used for the event correlation and the decision process, whether a suspicious action is benign or malicious. Therefore, these status updates are quite important.

In case you want to implement the application layer IDS of section 2.4.3 on page 40 or the random walker of section 2.4.7 on page 45 there has to be a function, that is capable of calculating the right amount of agents. This is quite hard to decide, because too many agents can slow down the network by producing too much overhead and too less agents are not able to detect all ongoing attacks. The right trade off has to be calculated by the nodes or some central manager.

In addition to that, the nodes have to log all activities in order to create a history when implementing the court based IDS of section 2.4.6 on page 44. The histories are needed for proving the innocence or guilt of some nodes and are used by the accused nodes themselves and by neighbours, who act as lawyers. Therefore, the nodes have to have enough memory to save the log files.

Authentication is very important in almost every MANET. But when using the application layer IDS of section 2.4.3 it is a must-have, because of the travelling agents. There is no determined agent on every node, but they travel around in the whole network, so it must be assured, that these agents are legitimated and not malicious.

Propagating the information, that one node is malicious is important for every IDS in MANETs. But when it comes to the watchdog approach of section 2.4.4, it is a key component of the system. This information is used by the pathrater to calculate the optimal path from the source to the destination. So the information, that a node is malicious has to be propagated quickly and most likely without knowing of the malicious node in order to prevent disturbing actions.

2.5.4 Summary

In this section, I summarized the requirements, that arise, when using the IDS described in section 2.4. These requirements can be divided into three groups: architecture, protocols, software. Speaking about architecture, there are the requirements for equal transmission power on the nodes, an upgradable agent platform, efficient algorithms, a centralized database, detection modules on the nodes and spending battery power for defending other nodes.

When it comes to protocols, you should have a reduced amount of protocols used. Furthermore you need source routing protocols for some solutions and a functionality, that orders nodes to monitor the flows.

There are a lot functions needed. E. g. you need status reports of the remaining battery power or the sensitivity level. Very important for some solutions is the ability to calculate the right amount of agents suitable for the current network topology. In addition to that, there are needs for a history and authentication mechanisms. Last, but not least, there has to be a function, that propagates the information, that a node is malicious.

Summary of requirements for IDS in MANETs

- architecture
  - equal transmission power
  - spending battery power for other nodes
  - upgradable agent platform
  - pre-installed integration modules
  - detection modules on all nodes
  - centralized database
- protocols
  - reduced amount
  - source routing
  - order to monitor flows
- functions
  - monitor status
  - calculating sensitivity status
  - calculating amount of agents
  - history
  - authentication module
  - propagating malicious nodes

2.6 Summary and Conclusions

2.6.1 Summary

In this paper, I first introduced IDS in section 2.2 and told about the different types and methodologies used for the detection of attacks. Section 2.3 told about MANETs and the threats and difficulties, that are typical for these networks. In section 2.4 I gave an overview about existing solutions for implementing IDS in MANETs. The derived requirements for networks and systems were explained in section 2.5.

The first conclusion I would like to state is the fact, that to my knowledge no one has actually implemented *the* IDS for MANETs. There is no solution available, that is battery power saving, very effective and adaptable at the same time. Even the proposed architectures have some shortcomings or make assumptions that cannot be transferred to reality. As an example I would like to mention the centralized database, needed by the lightweight IDS agents. Such an centralized object is unlikely to be practical in a mobile ad-hoc network scenario.

But some of the requirements, which look impractical at the first view, can be achieved in some usage scenarios. When thinking of military use, all potential nodes are known before the scenario begins. Therefore all nodes could be equipped with the needed modules. This gives the opportunity to use some of the described solutions. On the other hand, if the scenario is civil and the potential nodes are not known before the beginning, these solutions can only be used along with a high complexity.

After considering the presented solutions, there is no full-satisfying system. Therefore a good IDS for MANETs would be a combination of two or three of the given solutions. E. g. combining the court based approach with the watchdog approach would lower the false positives of the watchdog, while making the accusation process very easy. Combining the random walker approach with the cooperative IDS would bring the best of two worlds - flexible and mobile agents and the static detection of flows by specific nodes.

The most interesting feature I discovered writing this paper is the sensitivity level of the lightweight IDS agents. This level brings a dynamism to the detection process, that is capable of adapting to the current situation. This can be perfectly combined with the specification based detection of the random walker approach.

2.6.2 Looking into the future

In the future, MANETs have to focus on a few protocols. This does not mean, that every MANET has to rely on the same protocols, but in a system, there have to be only few protocols. This reduces the amount of vulnerabilities in the network. In addition to that, the reduced amount gives the opportunity to implement specification based IDS. So there will be a greater choice of usable IDS.

Another attempt to secure the IDS and the MANETs is a read-only memory (ROM) for the IDS. This would eliminate the possibility of manipulating the IDS. When writing the authentication modules and the keys in this memory, the whole system will be more secure. In order to provide flexibility, this should be an electrically (erasable) programmable ROM, which can only be rewritten with physical access. If such an implementation is not possible, the IDS should not rely on upgrades or updates in order to avoid manipulation. Alternatively it is possible to try to establish an pass-phrase-mechanism for the authentication of the nodes.

In addition to the ROM, there should be a writeable memory for extensions of the IDS. This will reduce the traffic in general, but will provide specific analysis and reaction capabilities, when needed. This memory and the extension modules have to be secured by a efficient, but secure algorithm. Relying on the usage scenario, secure can mean to be not hackable for a few hours, days or even months.

The two memories become possible, because physical size of memory is shrinking and the energy consumption is reduced as well. In addition to that, batteries become more powerful while reducing size. This gives more calculating power to the nodes and the IDS on them. This power should be saved as much as possible, but could be used for further and deeper analysis, when needed.

The last important feature, every future IDS in MANET should have is a great history component. This should provide the capability of analysing events, correlating some previously not correlated events and should give hints on how to improve the detection rate. In contrast to that, the needed CPU and battery capacity should be as low as possible, because the history is much less important, than the detection module.

Further work should evaluate, how these requirements can be achieved efficiently. Moreover the chosen IDS solution has to be tested in real environments; this is a fact, almost every system lacks till today.

Challenges in the future

- reduced amount of used protocols
- IDS in ROM
- writeable memory for IDS-extensions
- satisfying history component

Bibliography

[1] Chang Katharine, Shin Kang G., *Application-Layer Intrusion Detection in MANETs*, Proceedings of the 43rd Hawaii International Conference on System Sciences, 2010.

[2] Sterne D., Balasubramanyam P., Carman D., Wilson B., Talpade R., Ko C., Balupari R., Tseng C-Y., Bowen T., Levitt K., Rowe J., *A General Cooperative Instrusion Detection Architecture for MANETs*, Army Research Laboratory.

[3] Manikandan T., Sathyasheela K. B., *Detection of Malicious Nodes in MANETs*, IEEE, 2010.

[4] Axelsson Stefan, *The Base-Rate Fallacy and the Difficulty of Intrusion Detection*, ACM Transactions on Information and System Security, Vol. 3, No. 3, August 2000.

[5] Scarfone Karen, Mell Peter, *Guide to Intrusion Detection and Prevention Systems (IDPS) - Recommendations of the National Institute of Standards and Technology*, NIST Special Publication 800-94, February 2007.

[6] Mishra Amitabh, Nadkarni Ketan, Patch Animesh, *Intrusion Detecion in Wireless Ad Hoc Networks*, In: IEEE Wireless Communications, February 2004.

[7] Helmer Guy, Wong Johnny S. K., Honavar Vasant, Miller Les, Yanxin Wang, *Lightweight agents for intrusion detection*, The Journal of Systems and Software 67 / 2003, pages 109-122.

[8] Marti Sergio, Giuli T. J., Lai Kevin, Baker Mary, *Mitigating Routing Misbehavior in Mobile Ad Hoc Networks*, ACM, 2000.

[9] Corson M. Scott, Macker Joseph P., Cirincione Gregory H.. *Internetbased mobile ad hoc networking*, IEEE Internet Computing, July-August 1999.

[10] Zhang Da, Yeo Chai Kiat, *A Novel Architecture of Intrusion Detection System*, IEEE, 2010.

[11] Panos Christoforos, Xenakis Christos, Stavrakakis Ioannis, *A novel Intrusion Detection System for MANETs*, Athens, Greece.

[12] Albers Patrick, Camp Olivier, Percher Jean-Marc, Jouga Bernard, Mé Ludovic, Puttini Ricardo, *Security in Ad Hoc Networks: a General Intrusion Detection Architecture Enhancing Trust Based Approaches*, Ècole Supérieure d'Électronique de l'Ouest (ESEO), Angers, France.

[13] Dreo Gabi, Helmbrecht Udo, *Sicherheit in der Informationstechnik*, Universität der Bundeswehr München, Germany, 2011.

[14] *Einführung von Intrusion-Detection-Systemen: Grundlagen*, Version 1.0, Bundesamt für Sicherheit in der Informationstechnik, Germany, October 2002.

Chapter 3

The Internet of things and its connection with Wireless Sensor Networks

OFR Thomas Eberle

Abstract.Nowadays if someone mentions the term 'Internet', he usually relates it with manually sharing information with other people in a network. There are many ways of sharing information manually: Chat clients such as Skype and ICQ, sending emails, looking for information on websites, listening to internet radio etc. In the future, however, scientists believe that internet will play a different, but still important role in our human civilisation. One of the major changes will be the idea of connecting physical objects in a single network. This vision is called 'The Internet of Things'.

This futuristic vision aims for more intelligence and automation in different processes, e.g. cars communicate with traffic lights to optimise traffic or luggage systems communicate with luggage to improve the fluidity etc.. For the implementation a structured network system is a vital character. One of the technologies are Wireless Sensor Networks which do already exist.

This paper contains all the background information about Internet of Things (IoT) and wireless sensor networks and concentrates on the important connection between those two future visions. Furthermore, you will understand different types of technologies which partly are already in use and you will see different kinds of architectural structures of these WSN.

Contents

3.1 The Internet of Things

3.1.1 Defining the Term 'Internet of Things'

Like many other technologies and theories, there are many definitions for this term. IoT basically represents a new kind of internet, which uniquely identifies objects and represents them virtually in a network. The object are supposed to dynamically interact and communicate with other objects and are tagged for assured identification. They will be part of business and information processes to simplify and speed up our economy and our lives.

This means that we can connect objects with other objects to improve efficiency. Cars will be able to communicate and share information with traffic lights or traffic systems, letters and packages will share information with post offices, sorting offices etc. This requires, however, integration into the information network: The common and present internet. Meeting this condition is still a major challenge in implementing the future internet.
A good example is a luggage system in an airport. Every day, an airport needs to logistically transport thousands of luggage items from the terminal to the correct aeroplane. Luggage systems are usually extremely complex networks. To improve the fluidity in such a network, a *Wireless Sensor Networks (WSN)* could be implemented to identify and tag each luggage inside the system. By using this theory the system is able to dynamically control the flow inside this network and therefore save time and money.

The complete Future vision aims for a complete Internet system combining all different types of the Internet: The Internet of things,

3.1.2 The Present Situation

For the time being, many organisations are working on a possible design for IoT.Especially mobile network companies such as Vodafone, O2 and mobile hardware companies (e.g. Motorola) are investing in scientific research.
According to the investors there are already first signs of a possible IoT. One major technology proving that an internet of things already exists is Radio Frequency Identification Technologies (RFID). Other used technologies such as Bluetooth, Infrared or WLAN can also be used to communicate in such a network. We will talk about RFID and the other important technologies in the next chapter.

3.1.3 Technology

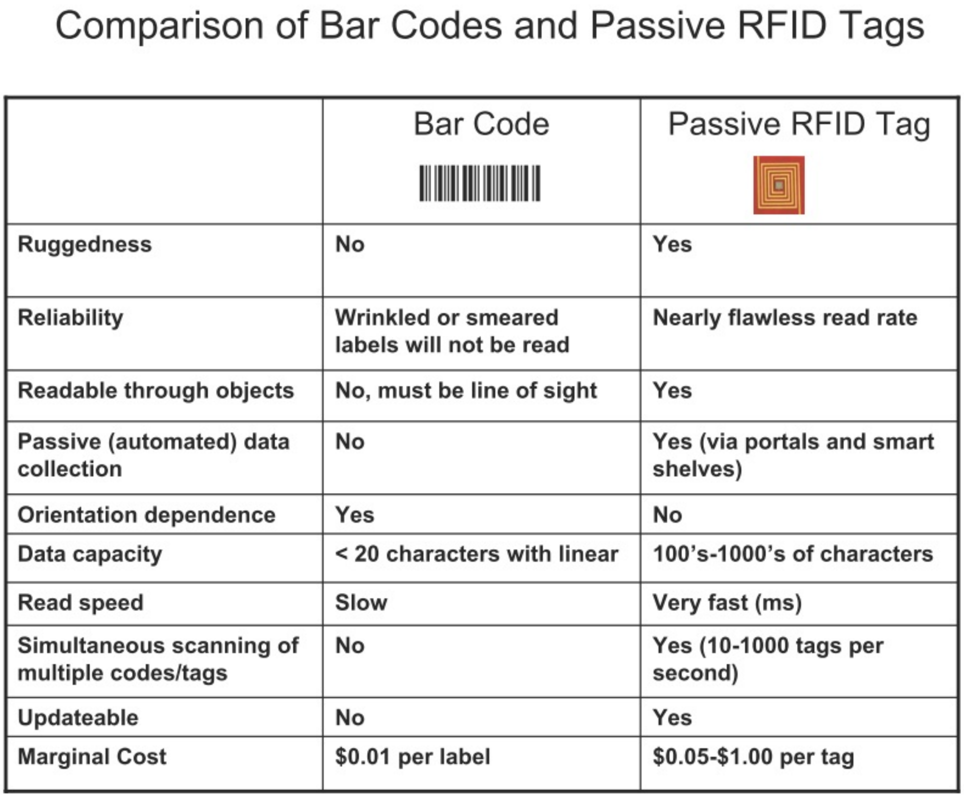
In this section, we will see different kinds of technology and have a look at the state-of-the-art RFID - the 'origin' of IoT in more depth.

RFID

Probably the best known technology in this topic is RFID - an identification, communication, localisation and tracking technology which follows the idea of Near Field Communication (NFC). RFID is summarising many aspects of IoT, which makes it a compulsory object in realising IoT.
An RFID tag (or transponder) is a small circuit containing an unique and individual ID , comparable with barcodes . The difference is, however, that these identification methods are completely unique where as a barcode usually categorises its objects. This ID has to be read by a RFID reader, which will then send the read key into a database which contains more background information about this object.

Advantages. There are many advantages with RFID:

- *Reliability.* RFID tags can be read from any angle without any data loss. The reading is compared to bar codes very fluent.
- *Speed and range.* Due to the fact that RFID uses a radio frequency technology with different frequencies, RFID tags and readers can operate with high speed and partly with high range. E.g. microwaves and high frequencies manage high reading speed on large distances.
- *Capacity.* RFID tags can save data in Kbyte range. Alternatively, barcodes only save a small limited amount of characters (approximately 20).
- *Update-ability.* RFID transponders can be updated with newer data.
- *Multi-reading.* Parallel reading is a possible feature in the RFID technology. RFID readers can multi-read RFID tags and therefore contribute to the speed range of the technology.
- *Transparency.* The circuits develop high transparency, as they can be read through other objects.
- *Energy Supply.* Passive RFID tags absorb their energy from their readers.
  Figure 3.1 shows the direct comparison between bar codes and RFID and summarises the main advantages of the RFID technology.



Comparison of Bar Codes and Passive RFID Tags

	Bar Code	Passive RFID Tag
Ruggedness	**No**	**Yes**
Reliability	**Wrinkled or smeared labels will not be read**	**Nearly flawless read rate**
Readable through objects	**No, must be line of sight**	**Yes**
Passive (automated) data collection	**No**	**Yes (via portals and smart shelves)**
Orientation dependence	**Yes**	**No**
Data capacity	**< 20 characters with linear**	**100's-1000's of characters**
Read speed	**Slow**	**Very fast (ms)**
Simultaneous scanning of multiple codes/tags	**No**	**Yes (10-1000 tags per second)**
Updateable	**No**	**Yes**
Marginal Cost	**$0.01 per label**	**$0.05-$1.00 per tag**

Figure 3.1: Comparison of Bar Codes and Passive RFID Tags

Standardisation of RFID. At the moment, RFID is still under development and not in complete business and commercial use. Many international ministries of technology are investing in different projects (for example SHAPE or the development of Object Naming Service (ONS) and RFID. ONS is the main infrastructure implemented with RFID tags and a database infrastructure, which is supposed to be standard in Germany in the future. In **3.1.4**(*Projects*) we will show the process chain of the ONS by showing the project ”EPCGlobal”.

RFID is already setting a standard for reaching the aim of a new generation of the internet. It is a promising technology and will definitely be used in our close future.

An example. Imagine using RFID tags instead of barcodes in a supermarket. The customer can easily fill their trolley and literally go through the till without moving any products from it. RFID tags are orientally independent, which means that they can be scanned from any angle through objects. On the other hand, a barcode needs to be in complete sight and clean, which makes it orientally dependent. RFID tags are therefore a simple solution for speeding up processes and make them more transparent, e.g. shopping in a supermarket.

Network Technologies and Various Protocols

To be able to uniquely identify billions of objects it is essential that the Network Technology should be developed. RFID as already mentioned and WSN - which we will discuss in Section 3.2 - guarantee the connection between physical objects and serve a possible connection to the internet. However, a protocol is needed that makes it possible to track and tag different objects uniquely. The most recent protocol is Internet Protocol Version 6 (IPv6). Compared to the present common standard, Internet Protocol Version 4 (IPv4), IPv6 supports approximately 3.4×10^{38}: That is 2^{96} times more addresses than IPv4 is able to support. The address space is compulsory to be able to connect all tagged objects in the internet of things. This means that the address space for the internet of things is almost given, as IPv6 already exists and is supposed to replace IPv4 in the next few years.

Nevertheless, network security is still an issue which needs to be addressed for the development of IoT. The network has to be scalable and cross platform compatible to meet Internet security standards. This means that different object platforms need to be able to connect with other object platforms, networks are supposed to connect and communicate with other networks. A full compatible network security is needed which can secure every single platform in this network. IoT developers will have to implement such a system to guarantee the success of IoT

Web Services. The term "Service" generally defines a set of goods or valuable functions offered by a service provider to a customer. In terms of telecommunication and internet, a service can be defined as a "packaged set of capabilities that are perceived by a human user when interacting with a telecommunications network or a service provider". The idea of IoT aims for an integrated physical world into the virtual or digital world. Furthermore, all interactions between machines and between a machine and a human are supposed to be simplified and smarter - Web Services literally aim for a more simple and easier environment and daily life.

More technique specific, services can be divided in two different groups:

- Low-level 'sensor data' services. These are all services which are related with sensor data and represent them in the network(e.g. Wireless Sensor networks). Furthermore, low-level services serve high level services with different results from their collected data

- High-Level services. Their target is reasoning and integrating data into a real-world model. High-level services therefore create the intelligent and most promising part for the future of Web-services.

(DPWS.) Devices Profile for Web Services (DPWS) is a very interesting idea for setting up a minimum of implementation on embedded devices to run web services. This profile has been defined and published in 2004 and standardised in 2008. This profile defines some compulsory features:

- Discovery Services (see Section 3.1.3)
- Web service Security
- Web service Description

Especially devices with restricted resources will profit from this standard as this profile was explicitly targeted on resource-constrained devices. It offers many web-service standards as well and is therefore an interesting protocol for inter-communication of distributed ”things”.

Eventing. Another important aspect of an Internet of Thing architecture is 'eventing' or monitoring the objects in the network. An Event is a type of message or action that usually occurs outside a program. After a 'fired' event, it replies a change to the programme and usually the programme interacts to this event. Very common examples are mouse clicks or key presses on the keyboard.
In this case, however, we are referring to real physical objects such as pallets or packages in a logistic company or products in a supermarket. IoT needs the ability to demand for the real time state of the object or thing. Compared to a simple computer, however, eventing in a complete internet is more challenging as the architecture is rather more complex and contains much more objects than a PC. To deal with this problem, developers have come to three different solutions:

- Complex Event Processing.
- Stateful Eventing.
- Event Web.

Operating Systems

For the correct task and system communication and operation of this system, a flexible and highly portable Operating System (OS) is needed. This section present 3 state-of-the art OS: Tiny Operating System (TinyOS), Contiki and Free Real-time Operating system (FreeRTOS). TinyOS and Contiki are specifically based on WSN, whereas FreeRTOS is a general operating system for a basic type of network. In this section the OS FreeRTOS will be described in more detail. Later in Section 3.2.2 we will go through TinyOS and Contiki in more detail.

FreeRTOS. FreeRTOS is a real-time c-programmed operating system generally designed for embedded devices, which has no specific relation with WSN. Similarly with TinyOS (later in section 3.2.2) FreeRTOS runs completely on the kernel space - based on a monolithic architecture. It provides however a system to schedule pre-emptive and non-pre-emptive tasks. Each task is completely independent towards other tasks within the system. Those tasks communicate either with queues(enqueue, dequeue), semaphores(Reserve and wait) or mutexes(avoidance of simultaneous use of common resource). Wireless communication and power management are not supported in this OS. A wired communication can be accomplished by various TCP/IP implementations with different restrictions. For wireless communication between sensors and different tasks, other OS such as Contiki are needed (see Section 3.2.2).

TinyOS. Same as FreeRTOS, TinyOS is a monolithic operating system, which is specially designed for WSN. It follows an event-driven scheme and is run by a complete event schedule in ”first come first serve”-order: Tasks can only be pre-empted by events but not by other tasks, which makes TinyOS very secure and powerful. Users in TinyOS can control CPU and radio options and specifications.Furthermore, TinyOS provides power management.The complete operating system is implemented in nesC, which stands for ”network embedded systems C”: A special type of the C language.
In terms of wireless communication , TinyOS is a proprietary MAC layer loosely based on IEEE 802.15.4 (Wireless Personal Area Network (WPAN)). However, a complete implementation for this standard is in progress.

Discovery Systems

An intelligent global network such as IoT also requires an intelligent discovery system in either way: Networks need to be automatically recognized and resources are supposed to provide the demanded information. This problem can be solved by discovery mechanisms, which this section will briefly describe.

Resource Discovery. For research and referral matters the Internet of Things will be based on a new search engine and data discovery technology. Compared to a data discovery system a data or information discovery technology recognises various data attributes and differentiates between them. It requires a development of lookup/referral services to link things. A Resource Discovery is supposed to supply the source with the demanded information and possibly refer to other linked information connected with this source.
An example: A product in a supermarket contains information about its producer, date of production and expiry date. A data requester could possibly ask for more information about its producer, depending on what authorisation they have.
To guarantee security, this referral service has to support security of access by sharing different authorisations for each source and system to build a trusted relationship between data requesters and providers. Furthermore, the system needs to be able to monitor the things in its recent network environment to discover the environment's capabilities such as

- availability of sensors and actuators,
- network communication interfaces,
- physical processing,
- alerts (e.g. alerting qualified personal to report technical problems),
- handling,
- facilities (processing data or similar facilities) etc.

.

Network Discovery. In an automated network, new "things" will be frequently added and out-dated "things" removed and, furthermore, networks will have to be moved around: A mapping administration (or management) system is also required to make automation and interaction in IoT possible by identifying Web services and things in different networks.On LAN level, software has already been implemented to map and control the movements of things and networks. Examples for this are Bonjour for Apple Systems, Web service - Discovery (WS-Discovery) and Simple Service Discovery Protocol (SSDP) for Windows operating systems. However, a key challenge will be the implementation of a network discovery software on worldwide or larger network basis. This will be one of many aspects we will be looking forward

to as soon as the internet of things is running.
An example: Some systems and devices do not support various network protocols. If, for example, a device doesn't support DPWS but only HTTP, a mechanism has to be introduced to automatically register the device with a DPWS.

3.1.4 Project 'EPCGlobal'

To understand how the technology based on the vision of IoT is working, the next section demonstrates an few up-to-date example to demonstrate the functionality of IoT.

A very recent project which has been started in October 2003 is the Electronic Product Code global (EPCglobal) project, which is supported by many companies and governments in the world. It has been formed to swap the product identification system from bar code to a much more efficient, flexible and transparent form of identification: The usage of RFID.
As we already discovered, RFID is able to save much more data than a barcode: In this case every object can get its complete individual Electronic Product Code (EPC): A code which basically is divided into different parts which describe the product with defined aspects. An EPC could therefore look like this:

- Head: Company Prefix : 20 - 40 Bits
- Centre: Object Class: 4 - 24 Bits
- Tail: Serial Number: 38 Bits

Short example: A company produces chocolate bars in any kind and type. The company prefix is a number which directly identifies this company. The object class determines that this object is a chocolate bar. With the last part the chocolate bar can be differentiated from other chocolate bars (e.g. white chocolate with chunks and caramel, low fat etc.). Rather than a simple bar code other companies would be able to identify this product without asking for any permission. EPCGlobal is therefore a project which mainly stands for globalisation. Companies who support this project are trying to build in restrictions, as this flexible system could be a security risk for the economy and its companies.

However, to give them product names and specifications a look-up system is needed. A service called ONS fulfils this task exactly:

- Read the EPC,

- Search or Look up for the name of this product,
- give name and details of this product.

All these details and specifications are saved in a product database called Electronic Product Code Information Services (EPCIS). Names from objects are saved in an ONS server. Those two servers build the management part of the complete EPC system. To see how the process works, see picture **3.2**.

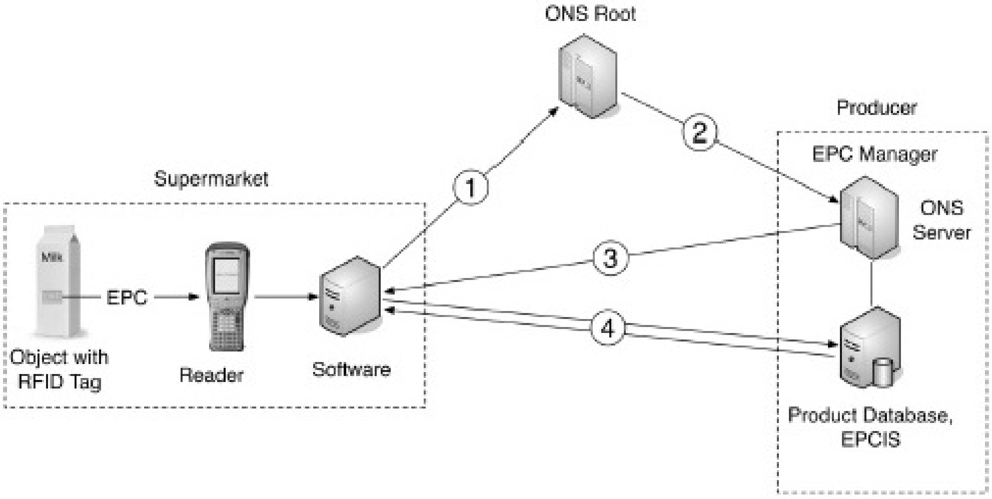


Figure 3.2: Main concept of EPCGlobal

3.2 Wireless Sensor Networks

3.2.1 Requirements

The concept of WSN is gradually gaining impact on our daily lives. Many applications across different sections such as health - care, environment, military, traffic and transport, economy, general management etc. do already exist.A WSN is a network which runs on low data and low energy sensors. Due to the fact that most sensors have a specific capacity in features and standards, there is a need for implementers and programmers:

- Reliability: A WSN system is supposed to gather a lot of sensor data in a short amount of time: data loss is supposed to be prevented by Quality of Service (QoS)
- Mobility: With a WSN it is possible to move sensors and system: a WSN can be a complete mobile system.
- Efficiency: Large amount of data is sent by using as little power as possible.

- Transparency: Sensors and network are completely invisible for the user.
- Accessibility: In an emergency, sensors can be easily reached by users.
- Replaceability: Following accessibility, sensor nodes need to be able to be replaced
- Environment-friendly: WSN are supposed to run on renewable enviromental friendly energies such as solar and wind
- Security: Attacks from outside the network and stealing secret information or private data has to be prevented. Furthermore, requlations and rights inside the network are compulsory to keep this network secure.

3.2.2 Technology

Standards

UWB. This is a fast growing technology in the WSN sector. The main idea is to use a huge frequency area to communicate with many different objects which use different frequencies. This technology is used on short distance. It is commonly used in the Standard IEEE 802.15 (Wireless Personal Area Network) as a Physical layer (PHY).
The method used is very simple. By sending or transmitting pulses with a frequency less than one nm the system is able to reach a complete spectrum of frequencies to guarantee that the signal reaches the correct object. This technology is also called *Impulse Radio-UWB* and is supposed to be the most promising technology for WSN applications.

The main disadvantage of this technology , however, is the fact that it could conflict with other radio systems as it uses a large range of frequencies. This could possibly cause interference, which could destroy or change important data.

Zigbee. This is a technological standard mainly created for the standard IEEE 802.15.4, which is designed for short range communication systems. This is usually used in home automation, industrial chain productions and management, medical sensor applications (see Paragraph **Body Area Network (BAN)**). Similar to the UWB technology, ZigBee is a highly supported technology from many communication technologies.
For the IEEE 802.15.4 standard there are 3 different operated and unlicensed bands/frequencies:

- 868 MHz: 10 channels , 20 kpps data rate, 1 kilometre range, European coverage.

- 915 MHz: 16 channels , 40 kbps data rate, 1 kilometre range, American coverage.
- 2.4 GHZ: 16 channels , 250 kbps data rate, 220 metres range, worldwide coverage.

Bluetooth physical interface. The oldest and probably best known low range low power and low cost communication is Bluetooth, which was designed by many communication companies such as Intel, Ericsson, Toshiba, Nokia and IBM. Similar to Zigbee, it operates in the Industrial, Scientific and Medical radio band (ISM band) of 2.4 GHz. The data rate and power consumption are higher compared to the Zigbee technology.
However, Bluetooth was mainly designed for the communication of a small amount of devices with an Ad-Hoc connection, where as Zigbee was designed to save power and protect the lives of batteries. This is one critical argument why Zigbee is better for Sensor networks than the Bluetooth technology.

Bluetooth Low Energy Technology ”Wibree”. Due to the fact that nowadays Bluetooth is a pretty out-of-date technology, scientists have developed a new technology based on Bluetooth with a lower energy demand. It was released in 2006 under the name ”Wibree”. There are 4 main differences to Bluetooth:

- Optimisation of the point-to-point link between two devices
- Optimisation of the transmitted packets during a connection
- Lower size of a data packet
- Lower number of channels.

The technology is not on the market yet. However, companies such as Texas instruments and Nordic Semiconductor are already developing Wibree devices.

Z-Wave. Z-Wave is already a common communication protocol standard used in households for home and lighting automation. Unlike ZigBee and Bluetooth it uses a 900 MHz frequency band and therefore offers an almost interference-free network. Z-wave networks rely on European regulations and are only used for duty cycles of 1 percent or even smaller. Most home controls systems do have a 1 percent duty control.

This technology offers a maximum of 232 objects in a network and transmission rates of 9,6 kbps and 40 kbps. It also allows a stable connection of devices and objects in a distance of 100 metres. Due to its good performance and low cost and closed radio standard, many companies are changing to this technology.

Operating Systems

Contiki. Contiki is a highly portable, C-programmed and open-source OS for small machines ranging from 8-bit machine until micro controllers and sensor nodes and targes explicitly WSN. It has been created at the Swedish Institute of Computer Science and is still under development.
Contiki supports embedded systems with a small amount of memory. It only needs a few kilobytes of space and needs just a few hundreds of bytes of RAM. Due to the fact that micro controllers and sensor nodes only have a small amount of memory due to their size this OS fits perfectly to their hardware architecture. For example, a common configuration of Contiki uses 2 kilobytes of RAM and 40 kilobytes of ROM, which is standard in resource-constraint technologies.
Furthermore, Contiki provides a complete IP stack IPv4 and IPv6. The addresses are needed to address the items and networks. As in recent time the addresses for the common Internet Protocol IPv4 have mostly been used up, IPv6 is, as already stated, needed and a main key for the success of IoT. The compatibility is another important feature of Contiki. Many systems and platforms can be run by Contiki. In the future, things will all run on different systems or platforms and will therefore need a unique and compatible OS. To simply name some systems,here are a few examples which support Contiki:

- Atari 8-bit (Computer)
- Casio Pocket Viewer (Computer)
- Game Boy (Handheld Game Console)
- Game Boy Advance (Handheld Game Console)
- Atari Jaguar (Video game console)
- Atmel AVR (Microcontroller)

Nevertheless, Contiki is specifically based on WSN. To fulfil the gap for a working network, another OS is obligatory. FreeRTOS is a part solution for embedded devices which are not connected through WSN.

3.2.3 WBAN: A Special Type of WSN

In this section we will look a bit closer at different types wireless networks, which are related with WSN.

WBAN

This is probably one of the most interesting researches in medicine since the last 3 decades: Wireless Body Area Network (WBAN) - A network which is wireless is implanted into a human's body, measuring important data about the physical and medical condition.WBAN is a transparent and highly flexible system which is meant to ease the conditions and work in hospitals and medical public accommodations. In the next few paragraphs you will learn more about this special type of wireless sensor network.

Standard IEEE 802.15.6 . In November 2007, the IEEE Standards Association formed a new Task Group specialising WBAN. It has the following definition: "The IEEE 802.15 Task group 6 (BAN) is developing a communication standard optimized for low power devices and operating on, in or around the human body (but not limited to humans) to serve a variety of applications including, consumer electronics/personal entertainment and other." In comparison, a WPAN doesn't meet certain conditions which can put a human in potential danger. On the other hand it is necessary for a WBAN to meet the following guidelines:

- Quality of Service.
- low power operation and consumption.
- no interference with other systems and networks.
- Meeting medical standard (e.g. no potential danger for the human tissue)

The creation of this task group shows us how important the concept WBAN for the industry is.

Services managed by WBAN. In present times, many deaths are caused by insufficient medical services and healthcare. A main cause of death are heart attack or stroke: In 4 years time (2015), these diseases will cause approximately 20 million deaths.
However, heart attacks and strokes are preventable. Medical technology has a state where heart attacks can usually be forecasted if patients are being monitored: This is the point where WBAN starts to become important, because WBAN enables medical workers, doctors and scientists to monitor their patients 24 hours 7 days a week without having their presence in medical institutions (hospital, pharmacy etc.): Patient monitoring is born.
To make such a type of monitoring possible, patients get sensors and actuators inside and outside their body. These sensors/actuators control and check all important and vulnerable places and functions of the human's body: Blood pressure, heart, brain, hearing, motion etc.

Comparison WBAN and WSN. In this paragraph we will compare the differences between WSN and WBAN.

- **Node Number:** WBAN will have a lower Node number in contrast to WSN: A WSN is supposed to cover a large area, where as a WBAN is only covering the human body.
- **Result accuracy:** WSN is becoming accurate due to its node redundancy , where as in a WBAN the nodes themselves need to be accurate: WBAN aims for quality, WSN for quantity.
- **Node tasks:** In a WSN, a node does a dedicated task for its complete lifetime, where as in a WBAN a node has to manage many different tasks.
- **Node size:** A small size for a WBAN is essential, whereas for a WSN a small size is preferred
- **Topology:** Due to variable motion in the human body the topology of the WBAN is variable compared to a likely static topology for a WSN
- **Data Rates:** The body motion also leads to varied rates for a WBAN.A WSN has more steady data rates.
- **Node replacement:** Replacing nodes in a WBAN is difficult due to implanted nodes in the human's body. In a WSN it is very easy replacing the nodes as they are much more accessible.
- **Power Supply:** Again difficulties arise in a WBAN making power supply more or less impossible, whereas in a WSN however power supply is easy to handle.
- **Power Demand:** A WBAN will have to use less power than a WSN
- **Energy scavenging source:** A WSN be powered by Solar and wind, a WBAN is powered by with body motion and thermal supply.
- **Biological compatibility:** It is critically compulsory for a WBAN to be biologically compatible with the human body. In comparison, this is not really necessary for a WSN
- **Security level:** The security level for a WBAN is higher to protect patient's private and secure information.
- **Impact of data loss:** A data loss in a WSN is likely to be compensated by redundant nodes. In a WSN, however, a data loss will be more significant and , furthermore may require additional measures to ensure QoS and real-time data delivery.
- **Technology:** A WSN is going to use wireless technology such as Bluetooth, ZigBee, WLAN etc. For WBAN, a low power energy is required (Bluetooth Low Energy?).

Connection with the IoT. In this case we are not talking about a lifeless object, but a human body. However, theoretically the human body can also be seen as a thing concerning the concept of the IoT: Sensors and actuators are attached to the body, making it accessible to communication. Furthermore, it is a step into complete automation of simple life circles: Illnesses and injuries can be diagnosed by WBAN even before you - as a human - are aware of any problems. To sum this up, a WBAN supports the concept due to its simplicity, transparency, automation and efficiency.

3.3 The Connection between IoT and WSN in the future

After giving you many specific examples for both concepts IoT and WSN, it is time to give a few more examples. However, you might already have noticed that WSN and IoT are usually combined.
The reason for this close connection between these two concepts is simply the similar aims between each of them: A WSN is aiming for a low range To clarify the importance of the connection between WSN and IoT. I first chose a protocol called TinyREST, which is already an attempt for the concept of IoT implemented by WSN. The second example I decided to demonstrate is an attempt to build a prototype of an IoT into traffic and transport: Car-to-Car communication.

3.3.1 TinyREST

To reach the aim using sensor networks to get and set information in a new kind of internet, sensor networks will have to be implemented into a global network. Nevertheless, this will lead to another standardisation of commands to control all sensors and objects in this network: a type of protocol is needed to manage all sensors.

There have already been a few attempts: This paper will give you the example 'TinyREST'.

Basics of TinyREST. TinyREST is based on the Representational State Transfer (REST), which is a concept based on the two common protocols Hypertext Transfer Protocol (HTTP) and Uniform Resource Locators (URL). By using simple HTTPcommands such as GET and POST a resource (The primary abstraction in REST) can get and set a defined state (e.g. 'on' or 'off', 'stop' or 'go' etc). Devices simply then get a HTTP address for easy access (e.g. http://sensor or http://sensor/light). This concept has the following advantages.

- User friendly, as HTTP is a common standard and widespread protocol.
- Good representation of data and information: All the information is formatted in American Standard Code for Information Interchange (ASCII) format making it for humans and human devices readable.
- Save system due to the ability of tunneling over firewalls: This makes this system secure and encrypted.

The 'Tiny' from the phrase TinyREST stands for the low energy consumption and small sensor concept: the idea is to make everything on sensor technology as small and low as possible, but still efficient enough to construct a stable and worthy global sensor network or in other words: A stable IoT. In the next paragraph you will see how this can be implemented.

Implementation of TinyREST. To get a low energy consumption, wireless and efficient network, the ZigBee standard (IEEE 802.15.4) can be used to reach this goal. The hardware used for this project which supports ZigBee are called MICAz: A 2,4 GHz mote module , ZigBee-ready and ISM band supporting with a data rate of 250 kbps. The 2,4 GHz band is the only ISM band which runs worldwide (see paragraph Zigbee).
Another advantage of this TinyREST protocol is that those MICAz sensors not only act as sensors but also as actuators due to the fact that this protocol runs on HTTP and is able to manage commands such as POST, GET and SUBSCRIBE: This brings us another step closer to the main concept of the IoT: Automation. These commands have the following meanings:

- POST: Make an action (command for actuators).
- GET: Get some information (command for sensors).
- SUBSCRIBE: Get a notification if a MICAz has reached a defined border (command for sensors).

An interface is necessary to control the connection and communication between clients and sensors. The interface in this project is provided by a HTTP-2-TinyREST gateway, which is responsible for establishing, handling and dropping a connection between client and sensor/actuators. Common HTTP regulations (e.g. validity checks or message format mapping) are included.
Furthermore, addressing is in this protocol very simple and user friendly due to the fact that we use HTTP and URL. The resources or devices are usually identified by their mote id or - if it is a group of resources - the group id. These ids,however, are saved on a client server inside the network: Users can simply find their device by specifying the direction of their request without remembering any mote or group ids.

3.3.2 Car-to-car communication

Another good example for combining both concepts is called Vehicular ad-hoc network (VANET), another term for car-to-car communication. there is a huge interest for these communication platforms. Many programmess and consortiums have been found in recent years (e.g. Car2Car communication Consortium Europe/USA or Honda's Advanced Safety Vehicle Programme). Similar to WBAN, the IEEE is working on a WiFi-standard for Car2X communication. One of the most promising candidates is the IEEE 802.11p : Wireless Access in Vehicular Environments (WAVE). The system works as follows: Each vehicle is equipped with the technology that allows the drivers to communicate with other vehicles and so called Roadside Unit (RSU), which are located in hazard traffic zones (e.g. intersections, traffic lights, stop signs): The goal of this system is the safety of all drivers. This ad-hoc network is automatically organised and also offers not only communication, but also other network applications such as internet access and services: According to the U.S. Federal Communication Commission, Vehicle-to-Vehicle (V2V) andVehicle-to-Roadside Unit (V2R) communication could save lives and improve traffic flow.
To gather all the information, different types of sensors are necessary to measure velocity, acceleration, position, condition of different vehicle parts such as brakes, dampers, tyres etc. Wireless sensor nodes are used to determine all this data. Furthermore, this data is then sent to a gateway inside the vehicle: This gateway then connects and communicates using its collected information with other vehicles to prevent accidents. All this is made possible by the idea of a WSN: Sensors are gaining information and cooperating with other sensors to analyse the data into information and sending this information to other users. To give you a brief example for the future of this technology, the next paragraph describes some of the following challenges and requirements

Challenges and Requirements in VANET design. To prevent attacks from outside which could cause harm to the traffic and transport, certain regulations and applications are needed.

- Privacy and Anonymity: User-related information has to be protected from unauthorised access including driver name license plate, speed, position and travelling routes.

- Jamming: This is preventing legitimate communications in reception range by generating interfering transmissions.Such attacks need to be prevented

- Impersonation: Where the attacker disguises himself as something or someone else, for example an emergency vehicle and misleads other vehicles by sending incorrect or made up information to the user. So therefore impersonators are a threat insofar as they can corrupt the system by inputting false information.

- Access Control: As already said in privacy and anonymity, access has to be controlled through local policies to prevent unauthorised access.
- Message Authentication and Integrity: messages from users have to be authorised and protected to ensure that no user is trying to attack the system.
- Forgery: An attack forging and transmitting wrong information about traffic hazards. Again, these attacks have to be prevented.
- Electronic Toll Collection (ETC): An application for paying toll electronically: this can save money by not needing toll stations and staff and saving time for drivers and users.
- Life-Critical Safety Applications: An application warning the driver of life threatening hazards such as collision at an intersection.
- Safety Warning Applications: These are applications warning the driver from work zones and traffic priorities. The difference between life critical safety and safety warning applications is the latency: Life Critical Applications have to be send immediately , whereas safety warning applications can be delayed.

Comparing this with the concept of IoT, we can again see the connection between WSN and IoT. You will find that WSN is a basic implementation in the IoT. Cars will be able to communicate with each other and share important data, which could save lives: The Internet of Things and wireless sensor networks are therefore not only an option, but a compulsory object even for saving lives.

Epilogue

In this paper we have seen that almost in all cases IoT and WSN have a strong relationship: Many sensor networks create a new way of communication between objects and the concept of IoT makes this communication global. Furthermore, the Internet of things is no longer a vision: It's already a true and realistic technology which is still under construction, but already has some realistic and tested ideas such as Car2X communication, EPCglobal etc. In the near future we will definitely see a new kind of internet, where objects can automatically communicate and improve our human way of life: Wireless sensor networks are the first step to this new way of communication. Future intelligent object will save us money and time and similar to the normal user internet, the IoT together with WSN will be an unavoidable and inevitable project for our civilisation.

Bibliography

[1] Nicola Bui. *Internet of Things Architecture*,Consorzio Ferrara Ricerche, Seventh Framework Programme 2011.

[2] Thomas Luckenbach, Peter Gober, Stefan Arbanowski, Andreas Kotsopoulos, Kyle Kim.*TinyREST - A Protocol for Integrating Sensor Networks into the Internet*, Germany/Greece/Korea.

[3] Sven Siorpaes, Gregor Broll, Massimo Paolucci, Enrico Rukzio, John Hamard, Matthias Wagner, Albrecht Schmidt.*Mobile Interaction with the Internet of Things*, Embedded Interaction Research Group, MEdia Informatics Group, University of Munich, DoCoMo Eurolabs

[4] Altons Botthof, Dr. Marc Bovenschulte, Dr. Sergei Evdokimov, Dr.Benjamin Fabian, Peter Gabriel, Prof.Dr.Oliver GÃ¼nther, Prof.Dr.Ernst A.Hartmann.*Internet of Things - Documentation* Federal Ministry of Economics and Technology Germany, May 2009.

[5] Chris Otto, Aleksandar Milenkovic, Corey Sanders, Emil Jovanov.*System Architecture of a Wireless Body Area Sensor Network for ubiquitous Health Monitoring*, University of Alabama, January 2006

[6] Lars Hoehmann, Anton Kummert.*Car2X-Communication for Vision-based Object Detection*,University of Wuppertal.

[7] Patrick Guillemin, Peter Friess.*Internet of Things - Strategic Research Roadmap*,CERP-IoT Project, September 2009.

[8] Delphine Christin, Andreas Reinhardt, Parag S. Mogre, Ralf Steinmetz. *Wireless Sensor Networks and the Internet of Things: Selected Challenges*, Multimedia Communications Lab, Technische UniversitÃ¤t Darmstadt, Germany.

[9] Commission staff of the European Communities.*Future networks and the internet - Early Challenges regarding the "Internet of Things"*,Commission of the European Communities, Brussels, 2008.

[10] Yi Qian, Nader Moayeri.*Design of Secure and Application-oriented VANETs*, National Institute of Standards and Technology, Gaithersburg.

[11] BenoÃ®t LatrÃ©, Bart Braem, Ingrid Moerman, Chris Blondia, Piet Demeester.*A survey on wireless body area networks*,Belgium ,November 2010

[12] Luca Mottola, Gian Pietro Picco.*Programming Wireless Sensor Networks: Fundamental Concepts and State of the Art*, University of Trento.

[13] Patrick Kinney.*Zigbee Technology: Wireless Control that Simply Works*,Kinney Consulting LLC, Chair of IEEE 802.15.4 Task Group.

Chapter 4

RFID Security and Privacy

Daniel Franz Breu

This paper surveys technical research on the problems of privacy and security for RFID (Radio Frequency IDentification)

RFID tags are small, wireless devices that help identify people and objects. Thanks to falling costs, they are likely to proliferate into the billions in the next several years - and eventually into the trillions. RFID tags track objects in supply chains, and are making their way into the pockets, belongings and even the bodies of consumers. This article examines approaches proposed by scientists for privacy protection and integrity assurance in RFID systems, and treats the social and technical context of their work.

Contents

4.1 Introduction - What is RFID?

RFID (Radio-Frequency IDentification) is a technology for automated identification of objects and people. Human beings are skillful at identifying objects under a variety of challenge circumstances. Computer vision, though, performs such tasks poorly. RFID may be viewed as a means of explicitly labeling objects to facilitate their *perception* by computing devices.
An RFID device - frequently called an RFID *tag* - is a small microchip designed for wireless data transmission. It is generally attached to an antenna in a package that resembles an ordinary sticker. The microchip itself can be as small as a grain of sand, some 0.4 square millimeters. An RFID tag transmits data over the air in response to interrogation by an RFID reader. Advocates of RFID see it as a successor to the optical barcode familiarly printed on consumer products, with two huge advantages:

1. *Unique identification*: A barcode indicates the type of object on which it is printed, e.g., "This is a 1 liter bottle milk of XYZ brand 1,5% fat." An RFID tag goes a step further. It emits a unique serial number that distinguishes among mans millions of identically manufactured objects; it might indicate, e.g. that "This is a 1 liter bottle milk of XYZ brand 1,5% fat, serial no. 685941258." The unique identifiers in RFID tags can act as pointers to a database entries containing rich transaction histories for individual items.

2. *Automation*: Barcodes, being optically scanned, require line-of-sight contact with readers, and thus careful physical positioning of scanned objects. Except in the most rigorously controlled environments, barcode scanning requires human intervention. In contrast, RFID tags are readable without line-of-sight contact and without precise positioning. RFID readers can scan tags at rates of hundreds per second. For example, an RFID reader by a warehouse dock door can today scan stacks of passing crates with high accuracy. In the future, point-of-sale terminals may be able to scan all of the items in passing shopping carts.

Today RFID helps to improve speed in supply chains by scanning the goos much faster than a normal scanner.
The main-form is known as EPC (Electronic Product Code) tag. The standards of these tags are overseen by an organization known as EPCglobal Inc.. Not surprisingly, EPCglobal is a joint venture of the UCC and EAN, the bodies that regulate barcode use in the United States and the rest of the world respectively.
The goal of the organization is to drop the costs per EPC tag under 5 cents apiece. The readers are more expensive and cost a few thousand dollars each, but it is likely that their cost will soon drop dramatically.
In the quest of low cost, EPC tags adhere to a minimalist design. They got little data on a on-board memory. The EPC code, that is the unique index of

an EPC code, carries information like that in a ordinary barcode but can also serve as a pointer to database records for the tag. The length of the tag can be up to 96 bit. But with the database pointers it is technically possible to save unlimited data. EPCglobal has invented ONS (Object Name Service), a public lookup system for the tags, which is analog in name and operation with the DNS (Domain Name Service). It routes general tag queries to the databases of tag owners and managers.

There are two types of RFID tags:
First the passive ones. They are quite inexpensive and have no on-board power source; the interrogating readers transmit the power to them. Passive RFID tags can operate in different frequency bands. Low-Frequency tags (124 kHz - 135 kHz) have nominal read ranges of up to half a meter. High-Frequency tags (13.56MHz) have ranges up to a meter and more. And the Ultra High-Frequency tags (860 MHz - 960 MHz, sometimes 2.45 GHz) have a range of up to tens of meters.
Second the active ones. They contain batteries and are divided into two types: semi-passive tag, which batteries power their circuitry when they are interrogated, and active tags, which batteries power their transmissions. These can initiate communication on their own and reach ranges up to 100m and more. Of course they are more expensive and cost $ 20 and more.

4.1.1 Current applications

- **Automobile immobilizers**: In this systems, the car key incorporates a passive RFID tag that the steering column authenticates, thereby enabling vehicle operation. The tags are usually factory programmed and cannot be rewritten in the field. Some versions include cryptographic communication between the key and the steering column.
  Widely credited with reducing auto theft by as much as 50 percent these systems are probably the best-known examples of RFID deployment translating into a measurable end-use benefit.

- **Animal tracking**: Organizations and individuals are increasingly equipping pets, livestock, exotic animals and endangered species with RFID tags to enable, recovery and management. In the US, many domestic cat and dog owners have RFID chips implanted in their pets. In August 2000, the Los Angeles City Council adopted a measure requiring that all animals adopted from the city's animal shelters have a microchip implanted at a cost of $ 15 per animal. Because the shelters also have RFID readers, lost animals recovered by a shelter can be easily returned to their owners. RFID chips are also being increasingly embedded into ear tags affixed to cattle. As another example, researchers have tracked dolphins and other marine animals with systems combining a GPS receiver with a radio transmitter that can be picked up by satellite (which costs approximately $ 4,000 per tag).

- **Payment systems**: RFID tags are being used as credit-card-like payment tokens that contain a serial number. A reader sends the number over a network and a remote computer debits value from the consumers's account. To make fraud more difficult, some systems combine the serial number with a simple challenge-response control. One of the most popular RFID payment system is Texas Instrument's Speedpass pay-at-the-pump system, introduced in Mobil stations in the mid-1990s. Several years ago, the European Central Bank purportedly considered embedding RFID tags into currency.

- **Automatic toll collection**: Highway authorities in many metropolitan areas now let travelers pay tolls using RFID tags linked to their debit accounts. One of the most popular is E-ZPass, first used widely in New York. E-ZPass is based on a 921.75 MHz semi-passive tag with a shelf life of about five to seven years and a read range of several meters. The tags can be read as cars move up to 100 miles per hour, making it possible to use the tags for traffic monitoring and other applications. Several million US consumers are now using these tags nationwide.

- **Inventory management**: For many, inventory management is the "holy grail" of RFID deployments. Individually serialized RFID tags are already being affixed to some consumer goods' packaging at the factory, then used to track packages as they get on the truck, travel by boat, arrive in the foreign country, leave the boat, enter the supply chain, travel through distribution and eventually reach their in-store destinations. Tags can assure that products produced and sold in one market are not illegally diverted to another. Further, “smart shelves” equipped with RFID readers could integrate with inventory systems, tracking all merchandise and alerting store personnel when items are misshelved. RFID tags might even be used after the sale, for example, to ensure that consumers actually bought items that they are attempting to return or have serviced.

4.2 Privacy and Security Problems

4.2.1 Personal privacy threats

RFID raises two main privacy concerns for users: clandestine *tracking* and *inventorying.*

RFID tags respond to reader interrogation without alerting their owners or bearers. Thus, where read range permits, clandestine scanning of tags is a plausible threat. As discussed above, most RFID tags emit unique identifiers, even tags that protect data with cryptographic algorithms (as we discuss below). In consequence, a person carrying an RFID tag effectively broadcasts a fixed serial number to nearby readers, providing a ready vehicle for clandestine physical tracking. Such tracking is possible even if a fixed tag serial number is random and carries no intrinsic data.

The threat to privacy grows when a tag serial number is combined with personal information. For example, when a consumer makes a purchase with a credit card, a shop can establish a link between her identity and the serial numbers of the tags on her person. Marketers can then identify and profile the consumer using networks of RFID readers - both inside shops and without. The problem of clandestine tracking is not unique to RFID, of course. It affects many other wireless devices, such as Bluetooth-enabled ones.

In addition to their unique serial numbers, certain tags - EPC tags in particular - carry information about the items to which they are attached. EPC tags include a field for the “General Manager”, typically the manufacturer of the object, and an “Object Class”, typically a product code, known formally as a Stock Keeping Unit (SKU). Thus a person carrying EPC tags is subject to clandestine inventorying. A reader can silently determine what objects she has on her person, and harvest important personal information: What types of medications she is carrying, and therefore what illnesses she may suffer from; the RFID-enabled loyalty cards she carries, and therefore where she shops; her clothing sizes and accessory preferences, and so forth. This problem of inventorying is largely particular to RFID.

Today the problems of clandestine RFID tracking and inventorying are of limited concern, since RFID infrastructure is scarce and fragmentary. As explained above, the tagging of individual retail items is probably some years away. Once RFID becomes pervasive, however, as is almost inevitable, the privacy problem will assume more formidable dimensions. One harbinger of the emerging RFID infrastructure is Verisign’s EPC Discovery Service. It creates a unified view of sightings of individual EPC tags across organizations.

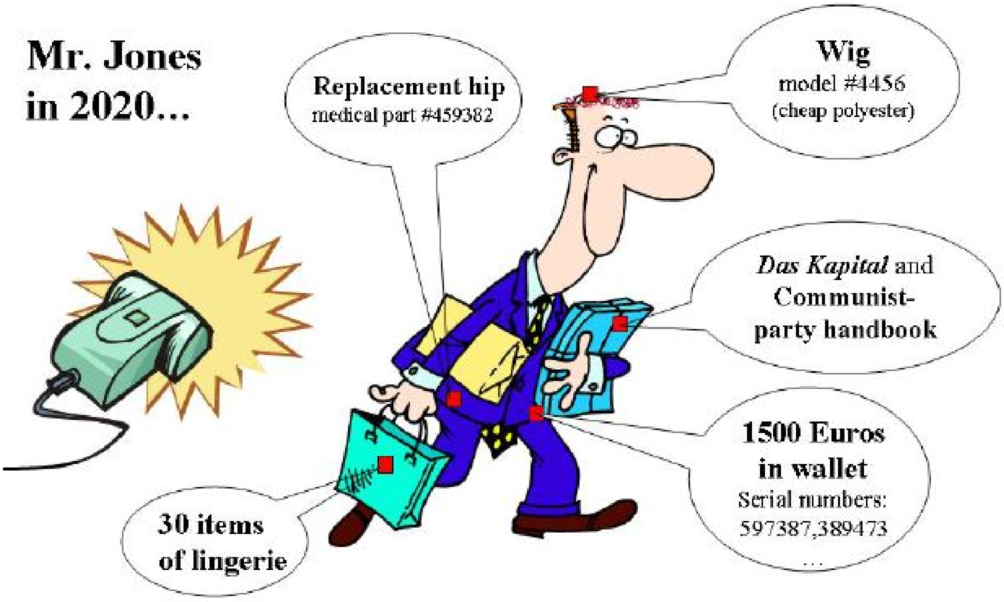

Mr. Jones in 2020...
Replacement hip medical part #459382
Wig model #4456 (cheap polyester)
Das Kapital and Communist-party handbook
1500 Euros in wallet Serial numbers: 597387,389473 ...
30 items of lingerie

Figure 1 illustrates the threat of clandestine RFID inventorying as it might in principle emerge in the future.

Remark: Some people like to point out that mobile phones already permit wireless physical tracking, and are practically ubiquitous. Mobile phones, however, have on/off switches. More importantly, mobile phones transmit signals receivable only by specialized telecommunication equipment. The owner of a mobile phone mainly reposes trust in her service provider. By contrast, most RFID tags are scannable by commodity RFID readers, which will soon be everywhere. Of course, mobile handsets increasingly exploit new channels like Bluetooth and WiFi, so some of the privacy distinctions between RFID tags and mobile phones will erode. Mobile phones, though, have fairly considerable computing power, and can support sophisticated forms of access control.

There is already considerable political and media ferment around RFID privacy. Several consumer advocacy groups have mounted campaigns against RFID deployment in retail settings. In 2003, for example, a boycott caused Benetton to disavow RFID-tagging plans for its garments (amid misconceptions about the company's plans). In the same year, a group of privacy organizations signed a position statement on the use of RFID in consumer products.
RFID privacy is already of concern in several areas of everyday life:

- **Toll-payment transponders**: Automated toll-payment transponders - small plaques positioned in windshield corners - are commonplace worldwide. In at least one celebrated instance, a court subpoenaed the data gathered from such a transponder for use in a divorce case, undercutting the alibi of the defendant.

- **Euro banknotes**: As early as 2001, the media reported plans by the European Central Bank to embed RFID tags in banknotes as an anti-counterfeiting measure. This project seems increasingly implausible given the attendant technical difficulties (not to mention the purported target date of 2005). It has served off and on, however, as a flashpoint for privacy concerns.

- **Libraries**: Some libraries have implemented RFID systems to facilitate book check-out and inventory control and to reduce repetitive stress injuries in librarians. Concerns about monitoring of book selections, stimulated in part by the USA Patriot Act, have fueled privacy concerns around RFID.

- **Human implantation**: Few other RFID systems have inflamed the passions of privacy advocates like the VeriChip system. VeriChip is a human-implantable RFID tag, much like the variety for house pets. One intended application is medical-record indexing; by scanning a patient's tag, a hospital can locate her medical record. Indeed, hospitals have begun experimentation with these devices. Physical access control is another application in view for the VeriChip.

In the United States, several states have initiated RFID privacy legislation, most notably California, where the state assembly considered (and rejected) bills in 2004 and 2005. Often overlooked in policy discussion is the REAL ID Act, recently passed by the U.S. legislature. This bill mandates the development of federal U.S. standards for drivers' licenses, and could stimulate wide deployment of RFID tags.

1. **Read Ranges**: Tag read ranges are an important factor in discussions about privacy. Different operating frequencies for tags induce different ranges, thanks to their distinctive physical properties. Under ideal conditions, for instance, UHF tags have read ranges of over ten meters; for HF tags, the maximum effective read distance is just a couple of meters. Additionally, environmental conditions impact RFID efficacy. The proximity of radio-reflective materials, e.g., metals and radio-absorbing materials, like liquids, as well as ambient radio noise, affect scanning distances. At least one manufacturer, Avery Dennison, has devised RFID tags specially for application to metal objects. Liquids - like beverages and liquid detergents - have hampered the scanning of UHF tags in industry RFID pilots. Protocol and hardware-design choices also affect read ranges.
   The human body, consisting as it does primarily of liquid, impedes the scanning of UHF tags, a fact consequential to RFID privacy. If in

the future you find yourself worried about clandestine scanning of the RFID tag in your sweater, the most effective countermeasure may be to wear it!

Sometimes RFID tags can foul systems by reason of excessively long range. In prototypes of automated supermarket-checkout trials run by NCR Corporation, some (experimental) patrons found themselves paying for the groceries of the people behind them in line.

Certainly, the RFID industry will overcome many of these impediments, so it would be a mistake to extrapolate tag capabilities too far into the future. It is important, however, to keep the limitations of physics in mind.

For the study of RFID privacy in passive tags, it is more accurate to speak not of the read range of a tag, but of the read ranges of a tag. Loosely speaking, there are four different ranges to consider. In roughly increasing distance, they are:

- **Nominal read range**: RFID standards and product specifications generally indicate the read ranges at which they intend tags to operate. These ranges represent the maximum distances at which a normally operating reader, with an ordinary antenna and power output, can reliably scan tag data. ISO 14443, for example, specifies a nominal range of 10 cm for contactless smartcards.
- **Rogue scanning range**: The range of a sensitive reader equipped with a powerful antenna - or antenna array - can exceed the nominal read range. High power output further amplifies read ranges. A rogue reader may even output power exceeding legal limits. For example, Kfir and Wool suggest that a battery-powered reading device can potentially scan ISO 14443 tags at a range of as much as 50 cm, i. e., five times the nominal range. The rogue scanning range is the maximum range at which a reader can power and read a tag.
- **Tag-to-reader eavesdropping range**: Read-range limitations for passive RFID result primarily from the requirement that the reader power the tag. Once a reader has powered a tag, a second reader can monitor resulting tag emissions without itself outputting a signal, i.e., it can eavesdrop. The maximum distance of such a second, eavesdropping reader may be larger than its rogue scanning range.
- **Reader-to-tag eavesdropping range**: In some RFID protocols, a reader transmits tag-specific information to the tag. Because readers transmit at much higher power than tags, they are subject to eavesdropping at much greater distances than tag-to-reader communications - perhaps even kilometers away.

Also of concern in some special cases are detection ranges, that is, the distance at which an adversary can detect the presence of tags or readers. In military scenarios, for example, tag-detecting munitions or reader-seeking missiles pose a plausible threat.

2. **Privacy from cradle to grave**: The importance of RFID privacy in military operations reinforces an oft-neglected point: Privacy is not just a consumer concern. The enhanced supply-chain visibility that makes RFID so attractive to industry can also, in another guise, betray competitive intelligence. Enemy forces monitoring or harvesting RFID communications in a military supply chain could learn about troop movements. In civilian applications, similar risks apply. For example, many retailers see item-level RFID tagging as a means to monitor stock levels on retail shelves, and avoid out-of-stock products. Individually tagged objects could also make it easier for competitors to learn about stock turnover rates; corporate spies could walk through shops surreptitiously scanning items. Many of the privacy-enhancing techniques we discuss in this survey aim to protect consumers, or at least human bearers of RFID tags. It is useful to bear in mind the full scope of the privacy problem, though. In a recent survey article, Garkfinkel et al. offer a taxonomy of threats across the different stages of a typical industrial supply chain.

4.2.2 Corporate data security threats

EPC poses a threat to corporate data security because many different parties can read tags. We have identified four threats here:

- **Corporate espionage threat**: Tagged objects in the supply chain make it easier for competitors to remotely gather supply chain data, which is some of industry's most confidential information. For example, an agent could purchase a competitor's products from several locations, then monitor the locations' replenishment dynamics. In some scenarios, they could read tags in a store or even as the merchandise is unloaded. Because tagged objects are uniquely numbered, it is easier for competitors to unobtrusively gather large volumes of data.
- **Competitive marketing threat**: Tagged objects make it easier for competitors to gain unauthorized access to customer preferences and use the data in competitive marketing scenarios.
- **Infrastructure threat**: This is not a threat specific to RFID per se. However, a corporate infrastructure that's dependent on easily jammed radio frequency signals makes organizations susceptible to new kinds of DenialofService attacks. Such attacks could be especially devastating as RFID becomes a mission-critical component of corporate infrastructure.
- **Trust perimeter threat**: Although not specific to RFID, as organizations increasingly share larger volumes of data electronically, the sharing mechanisms offer new opportunities for attack.

4.2.3 The cloning threat

Researchers at Johns Hopkins University and RSA Laboratories recently identified a serious security weakness in the RFID tag in Speedpass devices and many automobile immobilizer systems. By demonstrating that such tags could be cloned, the researchers revealed the possibility of payment fraud and new modes of automobile theft. Although their discovery does not directly undermine consumer privacy, it demonstrates that RFID tags could have security consequences beyond merely tracking or profiling consumers.

4.3 Attack models

In order to define the notions of “secure” and “private” for RFID tags in a rigorous way, we must first ask: “Secure” and “private” against what? The best answer is a formal model that characterizes the capabilities of potential adversaries. In cryptography, such a model usually takes the form of an “experiment”, a program that intermediates communications between a model adversary, characterized as a probabilistic algorithm (or Turing machine), and a model runtime environment containing system components (often called oracles). In the model for an RFID system, for example, the adversary would have access to system components representing tags and readers.

In most cryptographic models, the adversary is assumed to have more-or-less unfettered access to system components in the runtime environment. In security models for the Internet, this makes sense: An adversary can more or less access any networked computing device at any time. A server, for instance, is always on-line, and responds freely to queries from around the world. For RFID systems, however, aroundtheclock access by adversaries to tags is usually too strong an assumption. In order to scan a tag, an adversary must have physical proximity to it - a sporadic event in most environments. It is important to adapt RFID security models to such realities. Because low-cost RFID tags cannot execute standard cryptographic functions, they cannot provide meaningful security in models that are too strong.

4.3.1 Physical attacks

The tags are powerless versus attacks like “fault induction”, “timing attacks” and “sudden power interruption”. Also direct attacks like embedding radiation or corrosive acid are possible. Because of the high costs to make the tags secure, they have to be classified as physically vulnerable.

4.3.2 Evaluation of the data traffic

Even to know that there are RFID tags in the environment can be a threat. But this very possibility is necessary for a RFID system.

The evaluation of the traffic pattern between tags and readers, counting the read operations, quantify the broad casted data gives a attacker the possibility to extract some information. For example a person could be identified because she carries a unusual high amount of tags.

4.3.3 Wiretapping

The communication takes place in the air and is therefore public. With the right equipment you can wiretap passive tags at a huge distance of up to one kilometer (900 MHz tags).

4.3.4 Spoofing

An attacker could try to imitate a tag or reader. For this he can use the wiretapped data. Therefore he could imitate the reader and get the saved confidential information.
He also can intercept messages and send faked ones instead to disturb the communication protocols and get useful information for compromising the system.

4.3.5 DoS attacks

There are a lot of ways to disrupt the work of an RFID system, because the tag depends on several things: it's own integrity, the reliability of the radio interface, correct operating protocols and so on.
The sledgehammer method for a DoS attack is to destroy the tag, often called "kill". You can kill a tag in electromagnetic ways, too much mechanic stress or by using aggressive chemicals. In contradistinction to other attack methods the tag is irreversible useless.
The radio interface can be shielded or disturbed with interfering signals. Shielding works with objects, which act as a Faraday cage. For example handbags with metal strips in it. The easier way to disturb the communication is sending a interfering signal at the frequency the system is acting in.

4.4 Models for basic and symmetric-key tags

- **basic tag**: cannot execute standard cryptographic operations like encryption, strong pseudo random number generation and hashing.

- **symmetric-key tag**: more expensive, but can perform symmetric-key cryptographic operations.

The categorization is a rough one, of course, as it neglects many other tag features and resources, like memory, communication speed, random-number generation, power, and so forth. It serves our purposes, however, in demarcating available security tools. We separately consider the problems of privacy and authentication protocols within each of the two categories.
Devices like RFID tags for shipping-container security, high-security contactless smartcards, and RFID-enabled passports can often perform public-key operations. While the general points in this survey apply to such tags, they are not treated explicitly. The majority of RFID tags - certainly passive ones - do not have public-key functionality. Moreover, existing cryptographic literature already offers much more abundant treatment of the problems of privacy and security for computationally powerful devices than for the weak devices that typify RFID.

4.4.1 Basic RFID tags

Basic RFID tags lack the resources to perform true cryptographic operations. Low-cost tags, such as EPC tags, possess at most a couple of thousand gates, devoted mainly to basic operations. Few gates - on the order of hundreds - remain for security functionality. It is tempting to dismiss this computational poverty as a temporary state of affairs, in the hope that Moore's Law will soon render inexpensive tags more computationally powerful. But pricing pressure is a strong countervailing force. RFID tags will come to be used in vast numbers; if and when they replace bar codes on individual items, they will contribute substantially to the cost of those items. Thus, given the choice between, say, a ten-cent RFID tag that can do cryptography, and a five-cent tag that cannot, it seems inevitable that most retailers and manufacturers will plump for the five-cent tag. They will address security and privacy concerns using other, cheaper measures.
The lack of cryptography in basic RFID is a big impediment to security design; cryptography, after all, is one of the linchpins of data security. On the other hand, the lack of cryptography in basic tags poses intriguing research challenges. As we shall see, researchers have devised a farrago of lightweight technical approaches to the problems of privacy and authentication.

Privacy

Most privacy-protecting schemes for basic tags have focused on the consumer privacy problems discussed above. (Industrial privacy, i.e., data secrecy, is important too, but less frequently considered.)

1. **“Killing” and “Sleeping”**: EPC tags address consumer privacy with a simple and draconian provision: Tag “killing”. When an EPC tag receives a “kill” command from a reader, it renders itself permanently inoperative. To prevent wanton deactivation of tags, this kill command is PIN protected. To kill a tag, a reader must also transmit a tag-specific PIN (32 bits long in the EPC Class-1 Gen-2 standard). As “dead tags tell no tales”, killing is a highly effective privacy measure. It is envisioned that once RFID tags become prevalent on retail items, point-of-sale devices will kill the RFID tags on purchased items to protect consumer privacy. For example, after you roll your supermarket cart through an automated checkout kiosk and pay the resulting total, all of the associated RFID tags will be killed on the spot.
   Removable RFID tags support a similar approach. Marks and Spencer, for example, include RFID tags on garments in their shops. These RFID tags, however, reside in price tags, and are therefore easily removed and discarded.
   Killing or discarding tags enforces consumer privacy effectively, but it eliminates all of the post-purchase benefits of RFID for the consumer. The receiptless item returns, smart appliances, aids for the elderly, and other beneficial systems described earlier in this article will not work with deactivated tags. And in some cases, such as libraries and rental shops, RFID tags cannot be killed because they must survive over the lifetime of the objects they track. For these reasons, it is imperative to look beyond killing for more balanced approaches to consumer privacy.
   Rather than killing tags at the point of sale, then, why not put them to “sleep”, i.e., render them only temporarily inactive? This concept is simple, but would be difficult to manage in practice. Clearly, sleeping tags would confer no real privacy protection if any reader at all could “wake” them. Therefore, some form of access control would be needed for the waking of tags. This access control might take the form of tag specific PINs, much like those used for tag killing. To wake a sleeping tag, a reader could transmit this PIN.
   The sticking point in such a system is that the consumer would have to manage the PINs for her tags. Tags could bear their PINs in printed form, but then the consumer would need to key in or optically scan PINs in order to use them. PINs could be transmitted to the mobile phones or smartcards of consumers - or even over the Internet to their home PCs. Consumers have enough difficulty just managing passwords today, however. The nitty-gritty management of PINs for RFID tags could prove much more difficult, as could the burden of managing sleep-/wake patterns for individual tags.

A physical trigger, like the direct touch of a reader probe, might serve as an alternative means of waking tags. Such approaches, however, would negate the very benefit of RFID, namely convenient wireless management.

2. **The renaming approach**: Even if the identifier emitted by an RFID tag has no intrinsic meaning, it can still enable tracking. For this reason, merely encrypting a tag identifier does not solve the problem of privacy. An encrypted identifier is itself just a meta-identifier. It is static, and therefore subject to tracking like any other serial number. To prevent RFID-tag tracking, it is necessary that tag identifiers be suppressed, or that they change over time.
   a) *Relabeling*: Sarma, Weis, and Engels (SWE) propose the idea of effacing unique identifiers in tags at the point of sale to address the tracking problem, but retaining product-type identifiers (traditional barcode data) for later use. Inoue and Yasuura (IY) suggest that consumers be equipped to relabel tags with new identifiers, but that old tag identifiers remain subject to re-activation for later public uses, like recycling. As a physical mechanism for realizing the idea of SWE, IY also explore the idea of splitting product-type identifiers and unique identifiers across two RFID tags. By peeling off one of these two tags, a consumer can reduce the granularity of tag data. Karjoth and Moskowitz extend this idea, proposing ways that users can physically alter tags to limit their data emission and obtain physical confirmation of their changed state. As a remedy for clandestine scanning of library books, Good et al. propose the idea of relabeling RFID tags with random identifiers on checkout.
   The limitations of these approaches are clear. Effacement of unique identifiers does not eliminate the threat of clandestine inventorying. Nor does it quite eliminate the threat of tracking. Even if tags emit only product-type information, they may still be uniquely identifiable in constellations, i.e., fixed groups. Use of random identifiers in place of product codes addresses the problem of inventorying, but does not address the problem of tracking. To prevent tracking, identifiers must be refreshed on a frequent basis. This is precisely the idea in the approaches we now describe.
   b) *"Minimalist" cryptography*: While high-powered devices like readers can relabel tags for privacy, tags can alternatively relabel themselves. Juels proposes a "minimalist" system in which every tag contains a small collection of pseudonyms; it rotates these pseudonyms, releasing a different one on each reader query. An authorized reader can store the full pseudonym set for a tag in advance, and therefore identify the tag consistently. An unauthorized reader, however, that is, one without knowledge of the full pseudonym set for a tag, is unable to correlate different appearances of the same tag. To protect against an adversarial reader harvesting all pseudonyms through rapid-fire interrogation, Juels proposes that tags "throttle" their data emissions, i.e., slow their

responses when queried too quickly. As an enhancement to the basic system, valid readers can refresh tag pseudonyms.
The minimalist scheme can offer some resistance to corporate espionage, like clandestine scanning of product stocks in retail environments.

3. **Distance measurement**: The barebones resources of basic RFID tags urge exploration of privacy schemes that shy away from expensive, high-level protocols and instead exploit lower protocol layers. Fishkin, Roy, and Jiang (FRJ) demonstrate that the signal-to-noise ratio of the reader signal in an RFID system provides a rough metric of the distance between a reader and a tag. They postulate that with some additional, low-cost circuitry a tag might achieve rough measurement of the distance of an interrogating reader. FRJ propose that this distance serve as a metric for trust. A tag might, for example, release general information ("I am attached to a bottle of water") when scanned at a distance, but release more specific information, like its unique identifier, only at close range.

4. **Blocking**: Juels, Rivest, and Szydlo (JRS) propose a privacy-protecting scheme that they call blocking. Their scheme depends on the incorporation into tags of a modifiable bit called a privacy bit. A 0 privacy bit marks a tag as subject to unrestricted public scanning; a 1 bit marks a tag as "private". JRS refer to the space of identifiers with leading 1 bits as a privacy zone. A blocker tag is a special RFID tag that prevents unwanted scanning of tags mapped into the privacy zone.

   Example: To illustrate how blocking might work in practice, consider a supermarket scenario. When first created, and at all times prior to purchase - in warehouses, on trucks, and on store shelves - tags have their privacy bits set to 0. In other words, any reader may scan them. When a consumer purchases an RFID-tagged item, a point-of-sale device flips the privacy bit to a 1: It transfers the tag into the privacy zone. (This operation is much like the "kill" function in EPC tags, and may be similarly PIN-protected.) Once in the privacy zone, the tag enjoys the protection of the blocker. Supermarket bags might carry embedded blocker tags, to protect items from invasive scanning when shoppers leave the supermarket. When a shopper arrives home, she removes items from her shopping bags and puts them in the refrigerator. With no blocker tag inside, an RFID-enabled "smart" refrigerator can freely scan RFID-tagged items. The consumer gets privacy protection from the blocker when it is needed, but can still use RFID tags when desired!

   How does a blocker actually prevent undesired scanning? It exploits the anti-collision protocol that RFID readers use to communicate with tags. This protocol is known as singulation. Singulation enables RFID readers to scan multiple tags simultaneously. To ensure that tag signals do not interfere with one another during the scanning process, the reader first ascertains what tags are present, and then addresses tags individually.

Blocking is of particular interest because, in exploiting singulation, it draws on the special operating characteristics of RFID. It is therefore worth giving a little detail.

One type of RFID singulation protocol is known as tree-walking. In this protocol, l-bit tag identifiers are treated as the leaves of a binary tree of depth l, labeled as follows. The root has a null label. For a node with binary label s, the left child has label s||0, the right child has label s||1.

The reader effectively performs a depth first search of this tree to identify individual tags. Starting with the root of the tree, the reader interrogates all tags. Each tag responds with the first bit of its identifier. If the only response received by the reader is a 0 bit, then it concludes that all tag identifiers lie in the left half of the tree; in this case the reader recurses on the left half of the tree. Conversely, a concordant response of 1 causes the reader to recurse on the right half of the tree. If the tag signals collide, that is, some tags emit 0 bits and others emit 1 bits, then the reader recurses on both halves of the tree. The reader continues recursing in this manner on sub-trees; it restricts its interrogation to tags in the current subtree. This procedure eventually yields the leaves - and thus the l-bit identifiers - of responding tags.

A blocker impedes RFID scanning by simulating collisions in the singulation tree. For example, a naively designed blocker could block scanning of all tags simply by emitting both a 0 bit and 1 bit in response to every reader interrogation, and forcing the reader to traverse the whole tree. Given that a typical tag identifier is, say, 96 bits in length, such a tree has many, many billions of leaves. So such a blocker would always cause a reader to stall!

As explained, the aim of the blocker is not wanton disruption of tag scanning. Rather, the scheme is selective. Blocking here relies on designation of the leading bit of a tag identifier as the privacy bit. The blocker only disrupts the scanning process when a reader attempts to scan tags in the privacy zone, i.e., in the right half of the singulation tree. The blocker does not interfere with the normal scanning of tags with 0 privacy bits, i.e., those outside the privacy zone. Figure 2 shows how blocking might work in conjunction with tree-walking in the supermarket scenario we have sketched.

A blocker tag can be manufactured almost as cheaply as an ordinary tag. Blocking, moreover, may be adapted for use with ALOHA singulation protocols (the more common type). To prevent undesired reader stalling, JRS also propose mechanisms whereby a blocker tag can be "polite", that is, it can inform readers of its presence so that they do not attempt to scan the privacy zone.

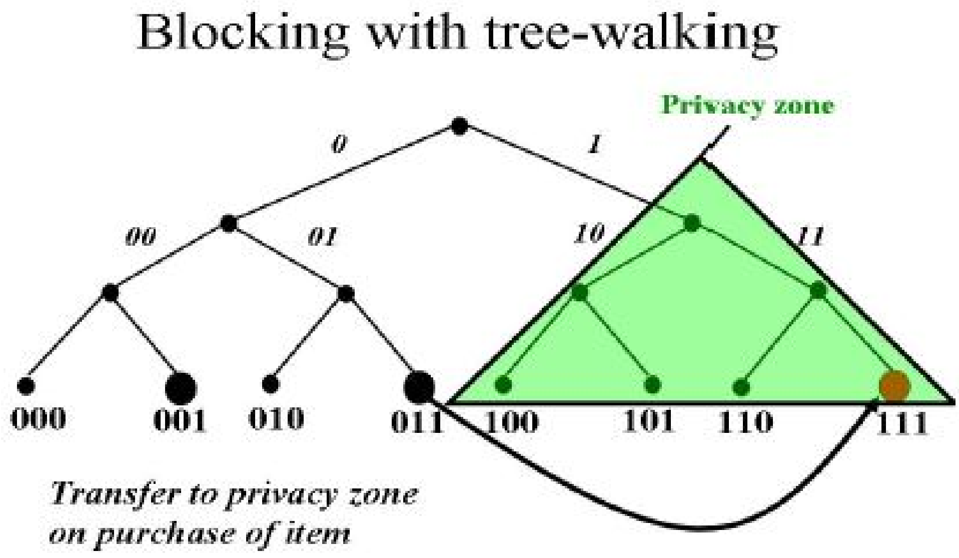

Blocking with tree-walking
Privacy zone
0
1
00
01
10
11
000
001
010
011
100
101
110
111
Transfer to privacy zone on purchase of item

Fig. 2. Illustration of how a blocker tag might work

Of course, the blocker concept has limitations. Given the unreliable transmission of RFID tags, even well-positioned blocker tags might fail. Readers might evolve, moreover, that can exploit characteristics like signal strength to filter blocker signals. On the other hand, improvements and variations are possible: A blocker might be implemented as an active device in a mobile phone, for example. Given the notoriously unpredictable behavior of RFID devices in the real world, both attacks and defenses merit careful empirical evaluation.

Authentication

EPC tags of the Class-1 Gen-2 type have no explicit anticounterfeiting features whatsoever. In principle, an attacker can simply skim the EPC from a target tag and program it into another, counterfeit tag - or simulate the target tag in another type of wireless device.
Juels shows a simple way to repurpose the kill function in EPC tags to achieve limited counterfeiting resistance. Normally, the kill PIN authenticates a reader to a tag in order to authorize the deactivation of the tag. Instead, this authentication can be reversed, and the kill PIN can instead serve to authenticate the tag to the reader. The basic protocol proposed in co-opts the ability of tags to distinguish between valid and spurious kill PINs.
Juels proposes an RFID protocol called yoking. It provides cryptographic proof that two tags have been scanned simultaneously - and evidence (although not proof) that the tags were scanned in physical proximity to one

another. A yoking protocol might, for example, allow a pharmacy to demonstrate to a government agency that it scanned an RFID-tagged medication bottle at the same time that it scanned an RFID-tagged booklet of contraindications - and thus that it furnished legally required information to consumers. One variant of this protocol is suitable for basic tags in that it requires virtually no computation, but it does require several hundred bits of storage per invocation.
Even if tags themselves do not have on-board anticounterfeiting features, they can support physical anticounterfeiting mechanisms. For example, physical one-way functions (POWFs) are small plastic objects with reflective inclusions such as tiny glass beads. Laser scanning of POWFs reveals unique, random speckle-patterns that may be translated into bit-strings; a POWF as small as 1 square centimeter can contain as many as 10^{12} static, random bits (effectively ROM). A POWF has two important anti-cloning properties: (1) Physical tampering destroys the information contained in a POWF, and (2) It is difficult to manufacture a POWF that emits a predetermined set of bits when scanned. Thus a POWF can help enforce unique identification of an object or container to which it is attached. While POWF data can be stored anywhere, an RFID tag may serve as a particularly useful carrier for POWF data. POWFs are just an attractive research concept at present. Many forms of packaging today contain special, proprietary (and secret) dyes and other physical markers of uniqueness. Basic RFID tags can equally well serve as carriers for their anti-counterfeiting data.

The problem of PIN distribution

As we have explained, both privacy and authentication features in basic tags can depend on tag-specific PINs. The kill function for Class-1 Gen-2 EPC-standard tags requires a PIN. There is also an option in the EPC standard for PIN-controlled write-access in tags. PIN distribution will almost certainly pose a major problem in the field, from the standpoints of both security and pure logistics. Once item-level tagging becomes prevalent, it will be necessary to provision point-of-sale terminals securely with the PINs for the RFID tags they are to kill. This problem, the perennial cryptographic one of key management in another guise, has seen only brief treatment in the RFID literature. Molnar, Wagner, and Soppera propose tree-based PIN distributions schemes akin to their ideas for privacy enforcement and delegation of secrets, which we summarize below; Juels proposes an extended tag authentication scheme in which readers prove their legitimacy through interaction with valid tags. The problem of secure PIN distribution merits much more investigation.

4.4.2 Symmetric-key tags

For brevity, I loose notation in this section, and assume very basic familiarity with cryptographic primitives. Recall that a cryptographic hash function h has the special property that for a random bit string M of sufficient length, it is infeasible to compute M from knowledge of the hashed value h(M) alone. Hashing involves no secret key (and is therefore only loosely called a symmetric-key function). In contrast, symmetric-key encryption, sometimes called secretkey encryption, relies upon a secret key k. With this key, a message or plain text M can be encrypted as a cipher text C = e_k[M]. Only with knowledge of k is it feasible to decrypt C and recover M.
In our discussions here we assume a centralized system, i.e., one in which readers are continuously on-line. We denote the number of tags in a system by n, and let T_i for $1 \leq i \leq n$ denote the identifier for the i^{th} tag in the system. We informally refer to this tag as T_i. We suppose that tag T_i contains in memory a distinct, random, and secret key k_i.

Cloning

In principle, symmetric-key cryptography can go far toward eliminating the problem of tag cloning. With a simple challenge-response protocol like the following, a tag T_i can authenticate itself to a reader with which it shares the key k_i:

1. The tag identifies itself by transmitting the value T_i.
2. The reader generates a random bit string R (often called a nonce) and transmits it to the tag.
3. The tag computes H = h(k_i,R), and transmits H.
4. The reader verifies that H = h(k_i,R).

4.5 Safety of sovereign documents

Despite the State Department's assurances that U.S. RFID passports are safe, some computer experts have demonstrated their ability to clone RFID passports issued by Germany and the United Kingdom. An American researcher named Chris Paget was also able to remotely scan and copy the information of two U.S. passport cards that contain a non-encrypted RFID chip, without the card owners' knowledge, by using a $ 250 scanner. Although these demonstrations indicate the potential for RFID passports to be illegally cloned, as of 2010, nearly four years after their implementation, there have not been any reported crimes of RFID hacks involving U.S. passports.

Bibliography

[1] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld.. *Physical one-way functions.*, Science, 297 2026-2030, 2002.

[2] S. Garfinkel and B. Rosenberg. *RFID Applications, Security and Privacy.*, Addison-Wesley, 2005.

[3] Ari Juels. *RFID Security and Privacy: A Reasearch Survey*, RSA-Laboratories, 2005.

[4] S. Stern. *Security trumps privacy*, Christian Science Monitor, 2001.

[5] Shannon George. *RFID Passport Dangers*, eHow, 03.08.2011.

[6] Dirk Henrici, Jochen Müller, Paul Müller. *Sicherheit und Privatsphäre in RFID-Systemen*, Technische Universität Kaiserslautern, 26.06.2011.

[7] Yohida. *Euro bank notes to embed RFID chips by 2005*, EE Times, 2001.

[8] A. Juels, R.L. Rivest, and M. Szydlo. *The blocker tag: Selective blocking of RFID tags for consumer privacy.*, 8th ACMConference on Computer and Communications Security, 2003.

[9] K. P. Fishkin, S. Roy, and B. Jiang. *Some methods for privacy in RFID communication.*, 1st European Workshop on Security in Ad-Hoc and Sensor Networks, 2004.

[10] M. R. Rieback, B. Crispo, and A. S. Tanenbaum. *Keep on blockin' in the free world: Personal access control for low-cost RFID tags*, 13th International Workshop on Security Protocols, 2005.

[11] A. Juels and S. Weis. *Defining strong privacy for RFID*, 2005.

[12] S. Garfinkel. *An RFID Bill of Rights*, Technology Review, Oct2002.

Chapter 5

Overlay Networks – Evolving the Internet

Jan Forkel

Abstract: *During the last years we have seen the rise of numerous so-called "overlay" networks in the Internet. Nowadays we know many of such overlay networks: Gnutella, Akamai, Tor, I2P, PlanetLab and 6Bone, to name only a few representatives. These, in turn, are categorized in several different classes. We can assume that most people already got in contact with at least one or two of these classes. There are voice-over-IP services offered via Skype, peer-to-peer file sharing networks associated with applications such as BitTorrent or Napster, content-delivery-caching networks implemented and run by companies like Akamai - and all of these are built up on the basic Internet.*

This paper provides a first attempt to understand the implications of such overlay networks for the Internet architecture and policy. Some representatives for overlay networks are being introduced and examined concerning the reason for their growing importance for the future Internet.

Contents

5.1 Introduction

To understand why it is worth taking a closer look at overlay networks and their future importance we can look back at the years of the very beginning of the global Internet. Today's Internet started as a government-funded research network running on top of the Public Switched Telecommunications Network (PSTN). So, in the beginning the Internet was a mostly unregulated data application that was mounted on top of the public-utility regulated telephone networks. To meet very special needs of a small research community the new Internet added functionality like a packet-switched data network to the underlaying basic infrastructure of the PSTN. As can be seen, the world wide web was an "overlay" that complemented the PSTN basic infrastructure which was already in place.

With the commercialization of the Internet in the 1980s and its emergence as a mass market platform for global communications in the 1990s, the Internet evolved into the principal platform for our global public communications infrastructure. With the Internet Protocol (IP) packet transport providing the basic transport medium for all kinds of communication, the world wide web does no longer exclusively serve research communities but also the common computer user. What was an "overlay" application has now become basic infrastructure.

The success of the Internet owes much to this common interest of *normal* users and specialists in the new technology but also the interpretability and connectivity supported by ubiquitous adoption of the IP protocols and the adherence to the end-to-end design principles that have governed Internet architecture for so long. However, the Internet's success has also posted significant problems. The incipient demand for functionality towards the Internet from amateurs and professionals nowadays became very unequal - today both groups have new needs and requirements. Besides, the persistent growth of the Internet brought heterogenous services while not everyone needs the same capabilities and complexity paired with size issues. To meet this challenges, the world wide web needs to continue to evolve. It is a process that looks like history repeating itself since there are many types and examples of overlays (see table 5.1) that arise to meet a range of purposes and needs. The future will have to show whether we can draw parallels between the Internet building up on the PSTN and present overlay networks and their relation to the IP based web. Perhaps one of the overlay networks operated nowadays will be the precursor of the future architecture of the Internet? Or they will all just be intermediate steps or even worse, the overlay networks known today will threaten the end-to-end connectivity and interpretability.

Before it is possible to answer these questions intelligently, however, it is necessary to gain a better understanding of what constitutes an overlay, the motivation for their development and use, and the potential conflicts and tensions that may arise among stakeholders. It is also a task to comprehend which implications overlay networks will result in.

Table 5.1: Examples of overlay networks

Type	Purpose	Example
Peer-to-peer	file sharing, distribution of data	Napster, Gnutella, BitTorrent
Content-delivery	content caching to reduce access delays and transport costs	Akamai, Digital Island
Routing	reduce routing delays, resilient routing overlays	Akamai, Limelight, SureRoute
Security	improved end-user security, privacy	Virtual Private Networks, onion routing (TOR, I2P), anonymous content storing (Freenet, Entropy), censorship resistant overlays (Publius, Infranet, Tangler)
Other	various	Email, VoIP (Skype), Multicast (MBone, 6Bone), Delay tolerant networks

This paper's purpose is to provide further knowledge about design, architecture and functionality of overlay networks.

Furthermore, different issues will be raised: The impact of these networks on the basic Internet infrastructure on the one hand, and some commercial and technical implications on the other hand. Thus, the balance of this paper is divided into three sections. The following chapter enlightens several technical details on the way towards a taxonomy for overlay networks, while section two illustrates this taxonomy in the context of three examples of overlays. The last sections provides a summary conclusions as well as suggestions for further research.

5.2 Towards a Taxonomy of Overlays

In this section we provide a taxonomy for thinking about overlay networks that includes examining the different motivations for emerging. This proves relevant when thinking about the prospective implications for the basic Internet infrastructure and the commercial implications. Therefore, we need to define what constitutes an overlay first.

The very short description in the introduction and the list of example overlays in table 5.1 offer a brief overview of diverse networks that appear to exist "on top" of another set of networks. They all have in common that they rely on the so-called *underlay* network for basic network functions, namely routing and forwarding. But from this point, even our first examples range from state-of-the-art distribution ways to special purpose systems that provide advanced routing up to very experimental networks.

5.2.1 What is an Overlay?

As a starting point we want to consider an overlay network as a set of distributed nodes, typically client devices or servers that are deployed on the Internet. The nodes are expected to meet the following requirements:

a) provide infrastructure to one or more applications

b) support high-level routing and forwarding tasks which are different from those that are part of the basic Internet,

c) can be operated in an organized and coherent way by third parties

The boundaries of this first definition are quite fuzzy. For that reason the definition offers different dimensions and considerations towards thinking about networks. This paper will focus on overlays that are not thought of as part of the basic Internet or provided by today's network service providers. At this point we offer one example how overlays may evolve. The email infrastructure of the Internet must be thought of as an overlay, just one that happens today to be operated by network service providers.

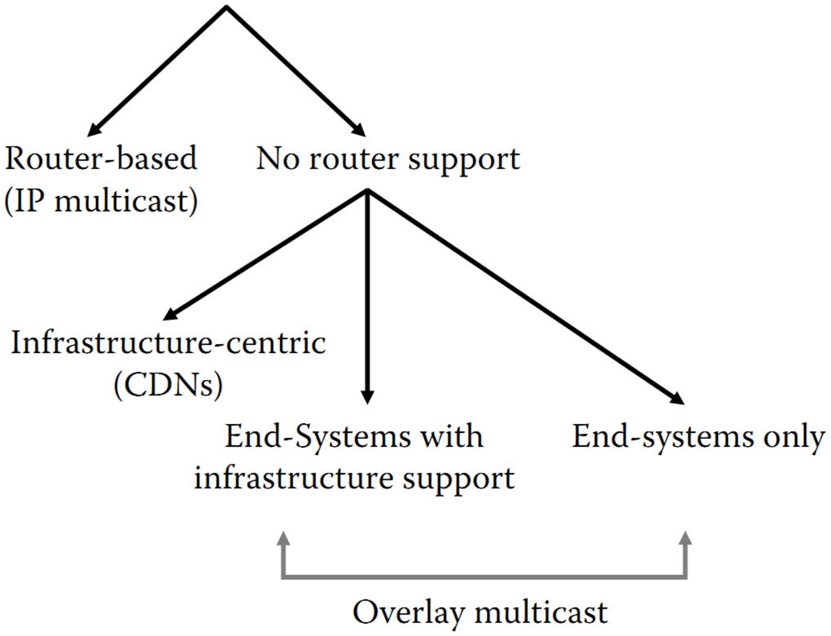

Router-based (IP multicast)
No router support
Infrastructure-centric (CDNs)
End-Systems with infrastructure support
End-systems only
Overlay multicast

Figure 5.1: a further approach towards a taxonomy according to [11]

Figure 5.1 offers another approach to sort overlay networks. Overlays can be router-based or they can be completely implemented on top of the underlay, typically TCP/IP. Router-based overlays typically employ IP multicast and IP anycast features. However, given the fact that deployment of IPv6 has not progressed according to most optimistic expectations, these extensions are not globally supported on the Internet. If the routers only provide basic unicast end-to-end communication, information networking functions need to be provided by the overlay. The two remaining categories illustrated in figure 5.1 are end-systems with and without infrastructure support. The former combines fixed infrastructure with software running in the end-systems in order to realize efficient data distribution. The latter category does not involve fixed infrastructure but rather establishes the overlay network in a decentralized manner.

5.2.2 Costs and Benefits

The very first step to find out why overlay networks emerge is to take a brief look at their costs and benefits.
As you one say money works at the impetus for all development it is highly prized that developers do not have to deploy new equipment for all the nodes across the Internet. In addition, they do not have to modify existing software or protocols; probably there have to be deployed new software on top of the existing software. The most common example is adding the IP on top of ethernet does not require modifying the ethernet protocol or its drivers. A further advantage is the robustness of overlays which allows them - due to their adaptable nature - to be tolerant against node and network failures. With a sufficient number of nodes in the overlay the network may be able to offer multiple independent paths to the same destination. At best overlay networks are able to route around faults. Another large benefit of overlay is to allow bootstrapping which refers to the development of successively more complex, faster network environments. This, again, is important because it is far too expensive to develop entirely new networking hardware and software from the very beginning. Moreover, it is not necessary to deploy the new overlay at every node. Firstly, yet every node needs or wants overlay network services all the time. Secondly, new overlay networks may be too heavyweight for some nodes if they consume too much memory, cycles or bandwidth. Equally important is the fact that new overlays may have unclear security properties, perhaps they can be used for Denial of Service (DoS) attacks.

From our point of view overlay networks have two major disadvantages, namely adding overhead and complexity. Every new overlay adds a layer in the networking stack with additional packet headers and the corresponding processing. Sometimes this extra work is redundant as the IP packet shows. It contains both ethernet header and IP addresses.
Layering does not eliminate complexity, it only manages it. So the more layers of functionality one node gets, the more possible unintended interactions happen between them.

5.2.3 Why Do Overlays Emerge?

■ support special needs

Overlays emerge for a variety of reasons. The first reason to be considered in this paper is the extra functionality offered by some overlays, beyond what is supported by the basic Internet. In the context of this discussion we claim that the basic Internet functionality is defined by the suite of core Internet protocols, namely IP, Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Domain Name System (DNS) and the Border Gateway Protocol (BGP). Any network or node must support this minimal set of basic protocols in order to be considered part of the Internet. Nonetheless, the services offered by the core Internet protocols are not always enough. Particular applications and user communities have special needs to be addressed through specialized functionalities and capabilities.
Though the original Internet architecture was designed to support unicast communication between fixed locations where the source knew the address of the destination. Whereas the mobile user has more general communication needs. Multicast, the delivery of packets to a group of destinations simultaneous in a single transmission and anycast, where packets from a single sender are routed to the topologically nearest node in a group of potential receivers all identified by the same destination address are generally used nowadays. In both examples, the source does not know the destination address which is a huge challenge for the current Internet architecture. In mobile communications it is also possible that the receiving host is not fixed or part of several networks.
Table 5.1 provides a range of functional extensions that overlays can provide. We now want to direct the attention towards customized routing functionality, mobility, Quality of Service (QoS), novel addressing, enhanced security and contend distribution.

As can be seen, overlays blur the clean principle of the Internet: On the one hand the end-to-end principle at application level in which data is no longer directly transferred from the source to the ultimate destination, and on the other hand the clean Internet architecture distinction between package forwarding and application processing. Overlays as application-specific network solutions are increasingly seen as the mechanism of choice for introducing functionality into the Internet. This is important because overlays have become a primary means for evolving the Internet architecture.

■ incrementally deploy innovations

According to the previous section overlays play an immense role in the dynamic evolution of Internet technology. It is a great advantage of the Internet's end-to-end architecture to incrementally deploy and adopt innovations so that applications can be deployed virally by a growing number of nodes without requiring modifications to the basic Internet. The Internet with its ubiquitous core protocols implements a stable platform to support communication among and across heterogenous edge-nodes. However, the very ubiquitous availability becomes a challenge when it comes to upgrading the Internet's own basic infrastructure. Coordinating the updating of all of the routers and servers that support the basic Internet represents massive undertaking, even if everyone agrees that an upgrade is needed and agrees on its nature.

On February 3, 2011 the Internet Assigned Numbers Authority (IANA), responsible for the IP address allocation, assigned the last IPv4 addresses to the five Regional Internet Registrys (RIRs). For that reason, it is time for the next big step in the Internet's history, switching to the Internet Protocol version 6, whose functionality was tested previously in projects like 6Bone or M6Bone which can be seen as overlays as well.

As this example shows overlay networks can provide a way to first experiment with new routing and architecture designs and then as a way to deploy new solutions. Likewise, new technologies like enhanced QoS or security and privacy mechanisms can be brought to users who require most and who are possibly willing to to pay for enhancements that may not be available on the general Internet yet. Over time, successful innovations will become ubiquitously adopted and, as such, de facto components of the basic Internet.

■ conflicts in stakeholder interests

Overlays may arise because of conflicts in stakeholder interests. Service-providers, costumers and policy-makers grapple with each other to gain the upper hand when it comes to decisions about numerous problems. For example at some point the basic Internet lacks of privacy. But overlays that implement ways to obscure the source, content or type of traffic might be in conflict with public policies that seek to make traffic auditable for law enforcement. Conflicts between consumers and service-providers are also inevitable. While the Internet Service Providers (ISPs) try to price discriminate differentiated services, consumers have a immense interest in network neutrality. There are conflicts within the different ISPs as well. Routing overlays that seek to improve on the basic Internet route selection process may be in conflict with policy-based routing implemented by peering ISPs in response to other non-delay-related considerations. ISPs try to manage traffic to minimize intercarrier payments. Therefore, these companies have business agreements which could be violated by overlays that try to select the best route based on global information about link delays.

5.2.4 Challenges and Limitations

Not only conflicts in stakeholder interest limit the implementation and use of new overlay technologies. Likewise, the typical underlay protocol IP does not provide universal end-to-end connectivity due to the ubiquitous nature of firewalls and Network Address Translations (NATs) devices. In computer networking, NAT is the process of modifying IP address information in IP packet headers while in transit across a traffic routing device. Due to the IP v4 address exhaustion it is common to hide an entire IP address space, usually consisting of private IP addresses, behind a single IP address. This means that special solutions are needed to overcome reachability issues. In addition, many overlay networks are oblivious to current organizational and management structures that exist in applications as well as in networks designs. Administrators have to deal with a further challenge of overlay networks. Practical deployment of an overlay requires them to have a management system because the administrator is typically removed from the actual physical device that participates in the overlay. This requires advanced techniques for detecting failed nodes or nodes that exhibit suspect behavior. This point is relatively easy to realize for a single administrative domain but when it comes to a fully operative system there are many parties involved and the management will be nontrivial. The overhead has to be mentioned again as well. An overlay network typically consists of a heterogeneous body of devices across the Internet. It is clear that the overlay network cannot be as efficient as the dedicated routers in processing packets and messages. Moreover, the overlay network may not have adequate information about the Internet topology to properly optimize routing processes.

5.3 Illustration of Different Types of Overlays

In this section we want to examine the technical and commercial challenges as well as the challenges concerning different policies posed by three different types of overlays: Content Delivery Networks (CDNs), Resilient Overlay Networks (RONs) and Security Overlays. For each example we provide a description of how the overlays operate and then identify issues raised by the growth of such overlays.

5.3.1 Content Delivery Networks

■ Introduction

The first class of network overlays we analyze is the Content Delivery Network, or Content Distribution Network. CDNs are overlay networks that dynamically cache content and services. Server farms and web proxies help to build large sides and to improve web performance but they are not sufficient for truly popular web sites that must serve content on a global scale. Therefore, generally CDN nodes are locally spread and connected over the Internet. All nodes work together to meet consumer requests in an economic way. In the background data is cached in a way that the delivery of content is performance-optimized or meets other factors. Major CDNs consist of thousands of nodes and servers. CDNs are overlay because the IP layer is responsible for delivering the packet to the appropriate destination but the decision about the source of packets is made at the application layer by the redirector, not the original requestor.

CDNs are technologically straightforward. When an application request content or services hosted by a CDN, the CDN overlay services the request from one or more of the distributed services throughout the Internet. It is a key feature of the network to select the most advantaged server respond to the request. This selection depends on multiple factors:

- load on each server (CPU idle, active connections...)
- which server is topologically nearby
- network capacity
- economic costs
- client information

■ Description

CDNs address a fundamental challenge on the Internet - how to distribute and acquire content cost-effectively while simultaneously lowering latencies experienced by end-hosts.

At a technical level CDNs consist of origin servers, geographically distributed surrogate servers, redirectors and clients. Source of the cached data is one single origin server. Content providers save their data here. The next step for the circulation is to transfer the data to so-called surrogate servers. These nodes provide exact copies of the original data. At this point the user's requests for data can be redirected by a request-routing-system to the single replica servers. This step is the crux of shared data distribution. At the moment a client requests data the request-routing-system selects one out of the appropriate surrogate servers to respond. Figure 5.2 clarifies this configuration.

Using a tree structure for these kinds of overlay has three virtues. The contend distribution can be scalded up to as many nodes as needed to prevent bottlenecks. Furthermore, each client gets good performance by fetching data from a nearby server instead of a distant one. Finally, with this layout the total load that is placed on the network is kept at a minimum. The idea of

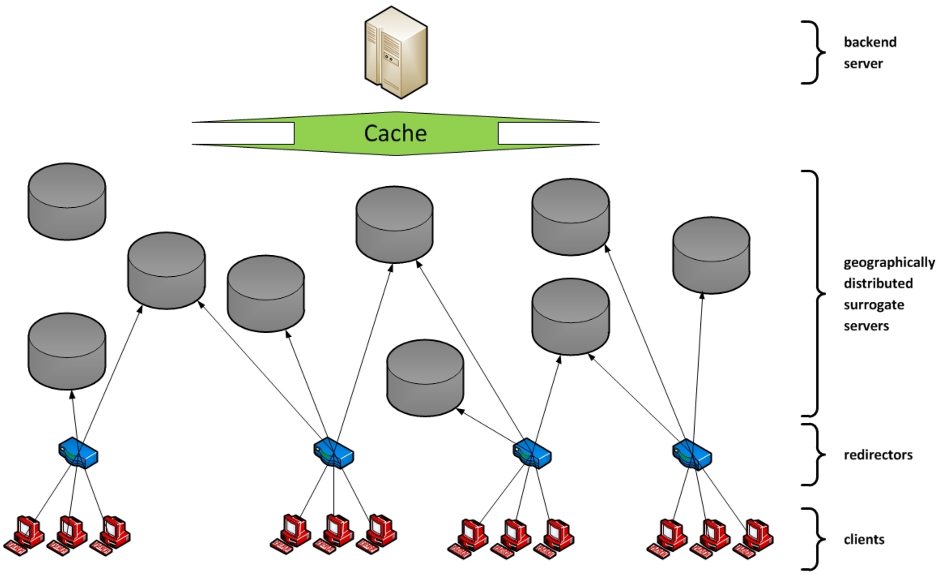


Figure 5.2: Content Distribution Network

using a distribution tree is straightforward. What is less simple is how to organize the client to use this tree if the clients are distributed all over the world as figure 5.3 shows.

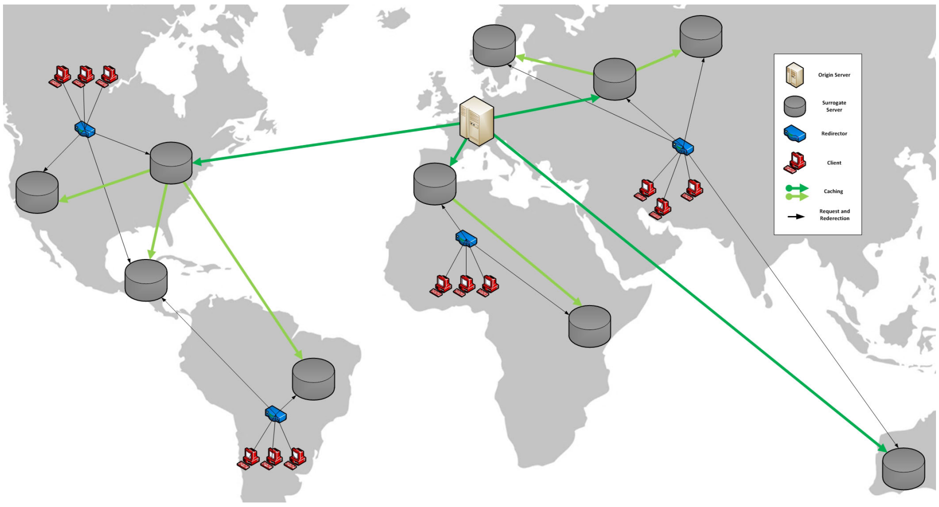


Figure 5.3: distribution all over the world

A simple way of sharing the network load and following the tree is to use the surrogate servers as mirrors and let the users decide which mirror they prefer. This technique is commonly used for large software packets provided on different servers at different locations. According to [2] the better approach uses DNS and is called DNS redirection. Suppose a client wants to fetch data from a server with the Uniform Resource Locator (URL) *http://www.unibw.de/inf/data.tar.gz*. To fetch the data the client will use DNS to resolve *http://www.unibw.de/* to an IP address. This DNS lookup proceeds in the usual manner. By using DNS protocol the client learns the IP address of the name server for *unibw.de*, then contacts the name server to ask it to resolve *www.unibw.de*. Then the name server is run by the CDN. Instead of returning the same IP address for each request it will look at the IP address of the client making the request and return different answers. The answer will be the IP address of the CDN node that is nearest the client.

CDNs fall into three different categories: commercial, cooperative and peer-to-peer based overlays. Beginning in 1998 *Akamai* [1] was the first commercial provider of CDNs to use DNS for content distribution. Currently commercial providers claim to serve a high amount of web content from their thousands of surrogate servers. Cooperative CDNs such as CoralCDN and OpenCDN seek to offer similar benefits to non-commercial users because they depend on infrastructure that is provided voluntarily. Finally, many peer-to-peer overlays function as content delivery networks. The idea of p2p networks is that many computers come together and pool their resources to form a content delivery system. This differs a lot from the system described above and could be subject of further research.

■ technical implications of CDNs

Content delivery networks have a lot of technical implications. As might have been sensed CDNs shift traffic patterns in the Internet. Data hosted by a German server requested by an Australian user must no longer be transported the whole way if the server in Germany has surrogate servers near Australia. This accelerates data distribution tremendously. It also benefits ISPs on the receiving end, content publishers which do not have to pay the expensive traffic costs arising from redundant requests and of course the consumers. Due to this shifted traffic CDNs affect where infrastructure investments are likely to occur in the future. One can possibly say that this results in larger caching capacities to serve traffic locally and that investments in wide area end-to-end capacity could be reduced. Additionally, once in place, CDNs cost money to support. Therefore such services may require additional payments.

Those services and content that are stored on CDNs may benefit from better performance and lower-cost access because of the efficiency benefits of caching. Hence, in the future CDNs may contribute to the creation of a two-class Internet. This has been a new issue in the recent debate over network neutrality. CDN operating companies could offer better treatment to certain content for a fee, or ISPs could prefer content requested from a CDN and deliver this data preferably to independent stored data.

As has been stated before CDNs dynamically change the communication pair by redirecting communication to different destinations. Luckily CDNs respect the clean end-to-end architectural distinction between packet forwarding and application processing. This is possible because the IP layer is responsible for delivering the packet to the appropriate destination but

[1] See http://www.akamai.de/index.html for further information.

the decision about the destination is made at the application layer by the redirector, not the original requestor.

5.3.2 Resilient Overlay Networks

■ Introduction

The Internet consists of thousands of autonomous systems with their nodes and multiple and redundant paths between each other. If one of these interlinks fails or even by default sometimes indirect paths may offer better performance for data transport between two servers. The idea behind Resilient Overlay Networks (RONs) is to use alternative paths to improve performance and to route around network faults. Figure 5.4 illustrates how overlay technology can be used to route around defects. In this example there is a problem with the normal path between A and C across the Internet. Now, the overlay can use a so-called detour path through B to send traffic to C. Of course this will result in some networking overhead but can be used to maintain communication between A and C. For that reason RONs reduce routing delays and costs, as well as they offer resiliency paired with flexibility.

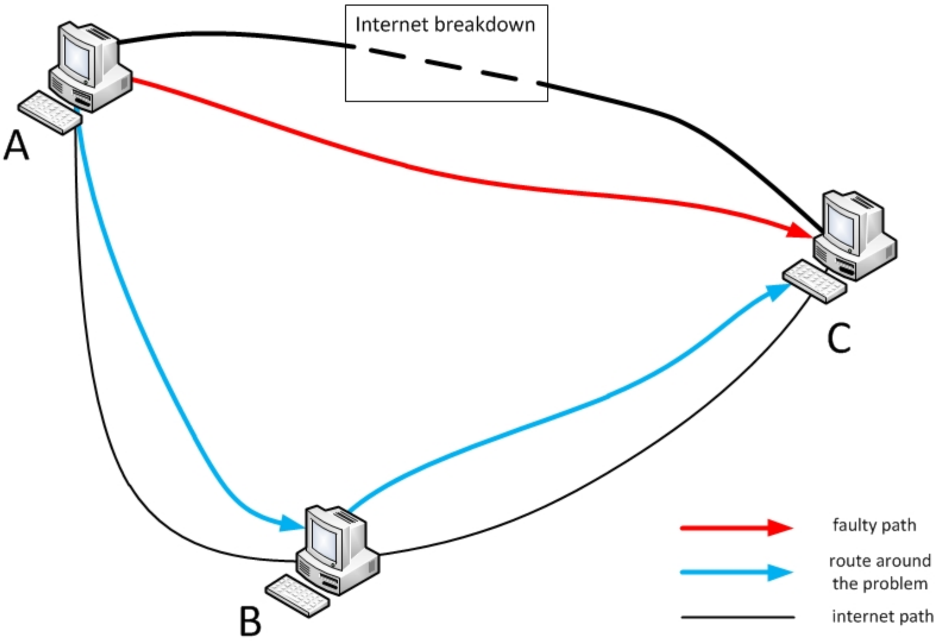


Figure 5.4: resiliency using overlay techniques

These efforts are necessary due to the steadiness of Internet routing. Internet connectivity failures, unfortunately, are not rare. Every network failure affects the availability of service delivery across Wide Area Networks (WANs). Paxson [7] states that “significant routing pathologies” prevent pairs of hosts from communicating 1.5% to 3.3% of the time, and more recent measurements [17] suggest that availability has not significantly improved. Chandra et al. [3] use probes to confirm that failure durations are heavy-tailed and report that 5% of detected failures last more than 2,75 hours, even up to 27,75 hours. Labovitz et al. [4] examine route availability by studying routing table update logs. They detect that only 25% to 35% of routes had an availability higher than 99.99% and that 10% of routes were available less than 95% of the time. Also 60% of the failures are repaired in an half hour or less, and the remaining failures exhibit a heavy-tailed distribution. These results are qualitatively consistent with our end-to-end analysis and provide additional evidence that connectivity failures may significantly reduce WAN service availability.

As is shown in this figures Internet scalability pays a price with slow recovery while RONs recovers fast by limiting size of overlay and exploiting redundancy in the underlying Internet. Therefore, the overlay measures all links between its nodes. It also computes all path properties and determines the very best route out of direct and indirect ones. Finally, it forwards the traffic over the selected path.

■ policy implications of RONs

Like everything else Resilient Overlay Networks may be misused. RONs allows users or administrators to define the types of traffic that is allowed on particular network links. Furthermore, it is possible to define separate routing policies for exclusive cliques and everyone else. So, again, it is a question of net neutrality; thanks to RONs it is easy to define which is less sensitive to latency and congestion in the backbone because of rerouting, while other content may be stuck on the information highway. For that reason there is again the thread of fragmenting the market which may lead to reduced scope and scale economies, factors that fueled the explosive growth of the Internet. Also RONs are related to the changing relationship between ISPs and their costumers. As long as ISPs retain control over routing decisions within the network there is little call for the technical routing mechanisms to resolve the “tussle” between the choice of the ISPs and those of end users. Therefore, RONs stand in direct competition with ISP because they provide a service that is generally provided by the ISPs. Every routing overlay gives the users an input into the routing decision. However, nowadays there is no coordinated way to resolve conflicting objectives between the various parties. Instead RONs simply allow end users to override the ISP in certain situations.

Table 5.2: examples of prominent security overlays

type	purpose	example
onion routing overlays	enable pseudo-anonymous communications over the Internet	TOR, I2P
anonymous content storage and retrieval	protect the identity of author, publishers and content providers when they store, query, and download content from the Internet	Freenet, Entropy
censorship resistant overlays	attempt to make it very difficult for powerful adversaries to remove content or pollute the overlay network with distracting material	Publius, Infranet, Tangler
deniability of knowledge	allow a sender to authenticate a message for a receiver, in a way that the receiver cannot convince a third party that such authentication took place	Off-the-Record (OTR) messaging, BitTorrent

5.3.3 Onion Routing

■ Introduction

The final class of overlay networks we want to discuss in this paper are security overlays. In table 5.2 some examples of prominent security overlays are provided. The aim of these networks is to provide different forms of communication protection, anonymity, censorship resistance or deniability of the knowledge of traffic. This is a particularly interesting class of overlays because even if the volume of traffic on these overlays may not be large, the policy and social implications can be significant. As has been mentioned in the previous section some overlays change the routing and caching behavior of communication and content on the Internet. The same applies to security overlays. But in contrast to the CDNs performance enhancement is not the intention but some aspects of end-user security.

As this class of networks tends to make the Internet opaque to regulation, it easily frustrates policy makers objectives. Some exponents of this network class use encryption to hide the content of communication while others try to hide the whole network. Due to clever use of cryptographic techniques and system engineering these networks provide provable properties about how hard they are to break.

However, the beneficial use of this type of overlay network is significant. Today most of these overlays base up on techniques originally developed by the United States Navy, like onion routing which was developed to hide the true origin of packets on an IP network [2]. This governmental interest on the one hand and the advantages for end-users on the other hand show the extreme currentness and importance of this topic.

[2] See http://www.onion-router.net/ for further information.

■ Onion Routing with Tor

Onion routing is an infrastructure for private communication over a public network. It provides anonymous connections that are strongly resistant to both eavesdropping and traffic analysis. Therefore, onion routing anonymous connections are bidirectional and near real-time.

Tor was originally designed, implemented and deployed as a third-generation onion routing project of the U.S. Naval Research Laboratory. It was originally developed with the U.S. Navy in mind for the primary purpose of protecting government communications. Today it is used every day for a wide variety of purposes by normal people, the military, journalists, law enforcement officers, activists and others.

According to the Tor project web page [3] Tor is a network of virtual tunnels that allows people and groups to improve their privacy and security on the Internet. They also mention a lot of examples people use its services for:

- keep websites from tracking user positions
- connect to news sides even if they are blocked by ISPs
- use instant messaging services securely
- socially sensitive communication
- prevent eavesdropping
- journalists communicate more safely with whistleblowers and dissidents
- surveilling web sites without leaving private IPs

But how does the Tor network work? Volunteers run Tor software to allow their computers to become Tor nodes. Tor nodes pass Internet traffic between each other securely and anonymously. When the traffic reaches the final destination it does so through a normal Internet connection.
To surf anonymously the user's client connects to the Tor network. During the startup the client software fetches a list of available and usable Tor servers from a directory server (figure 5.5 step 1). This list is digitally signed so that every Tor proxy can obtain an authentic directory. With the received directory the client software is able to select a random route through the Tor servers to the destination. As a next step the client negotiates an encrypted connection with the first Tor server (figure 5.5 step 2). Then the new server on his part extends the chain by selecting a random second Tor server and opens a further encrypted connection. Afterwards, the second server sets up a connection to a third server in the same way (figure 5.5 steps 3 and 4).
Due to this method every server knows his predecessor and successor. Moreover, all connections consist of at least three Tor servers. The Tor developers

[3] See https://www.torproject.org/about/overview.html.en

decided to use three servers to incur as high anonymity as possible with an appreciable delay. The success depends on at least one trustworthy Tor server, and the initial and end point must not be observed by attackers. Upon opening the connection data can be transported across the servers. The last Tor server within the chain therefore operates as exit node (figure 5.5 step 5). Behind this point the packets could no longer be encrypted. Thus, it is highly recommended to cipher the data itself. To provide further security the trunking scheme specified above is going to be established new every ten minutes.

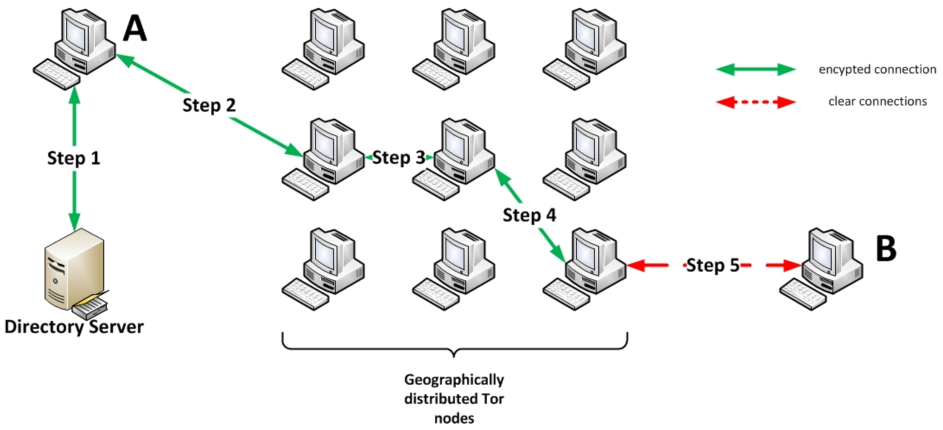


Figure 5.5: negotiation of a Tor connection

As can be seen in the example of Tor onion routing networks are generic transport or network layer overlays capable of providing anonymity to any application. Care must still be taken as applications may leak the identity of an anonymous host in other ways that are protected by the overlay network. Ensuring anonymity is therefore still a non-trivial task for most users of overlays.

■ Censorship Resistance

Out of the 40 countries studied by the OpenNet Initiative [4] in 2006, 26 censored the Internet in some way. The types of material censored varied depending on the country, e.g.:

[4] See http://opennet.net/

- human rights (blocked in China)
- religion (blocked in Saudi Arabia, UAE, Iran, Bahrain)
- pornography (blocked in Saudi Arabia, UAE, Iran, Bahrain, Singapore, Burma)
- other issues censored include:
  - military and militant web sites
  - sex education
  - alcohol and drugs
  - music
  - gay and lesbian web sites
  - news

As this figures show powerful adversaries try to remove certain types of content from the Internet or make them inaccessible. There are different architectures and different ways to try to resist censorship. The mechanism used by Tor and described in the previous section helps to make it difficult to locate users downloading censored content. Systems like Freenet [5] help to provide content without revealing the storage location of the data.

Another general strategy to avoid censorship is to automatically cache content in many locations and preferably in many different legal jurisdictions. For that reason it is much more frustrating trying to remove content in a legal way by making the organization interested in removing content. Different jurisdictions make this much more complex for each file they want removed. Security overlays and CDNs alleviate this method.

■ Providing Deniability

In computer networks deniability often refers to a situation where a person can deny transmitting or storing data even when it is proven to come from his computer. Normally, this is done by setting the computer to relay certain types of broadcasts automatically in such a way that the original transmitter of a file is indistinguishable from those who are merely relaying it. That way, the person who first transmitted the file can claim that his computer had merely relayed it from elsewhere, and this claim cannot be disproven without a complete decrypted log of all network connections to and from that person's computer. This property, while being independently useful in certain circumstances, also contributes to an overlay's censorship resistance by providing a defense that no intent existed to host illegal content.

To make it possible to deny knowledge of stored content it is possible to store encrypted data but not the encryption keys on the same node. Each

[5] See http://freenetproject.org/

node can therefore plausibly assert that they do not know the content of the files on their system. In such systems the individual node can normally not choose the content they host. This approach is taken in the Publius [12] and PAST [15] systems.

According to the Publius web site [6] Publius is a web publishing system that is highly resistant to censorship and provides publishers with a high degree of anonymity. The system consists of publishers who post content on the web, servers who host random-looking content, and retrievers who browse Publius content on the web. Therefore, Publius content is encrypted by the publisher and spread over some of the web servers. The publisher takes the encryption key and splits it into shares. Furthermore, each server receives the encrypted Publius content and one of the shares. At this point, the server has no idea what it is hosting, it simply stores some random looking data. To browse content a retriever must get the encrypted Publius content from some server and all of the shares. This very brief description shows why Publius is an outstanding example for deniability of knowledge.

■ Impact of Security Overlays

Content delivery networks carry huge amounts of traffic, quite contrary to security overlays. But the impact of these special overlays providing anonymity, censorship resistance and deniability is significant. It is not in question, for better or worse, they do certainly complicate notions of identity and responsibility on the Internet. Thus, they make the job of law enforcement and even national security more difficult.

The debatable networks provide technically justifiable excuses for most network traffic or digital content on a computer. They fundamentally change the notion of identity. While the binding between an IP address and an end-user was never absolute these networks completely break correspondence, even for IPv6 where the owner's Media Access Control (MAC) address could be part of the static IP.

But all the mentioned benefits on the one hand, there are drawbacks on the other hand. The encryption, spreading, and deniability of knowledge leads to a tension with the interests of law enforcement. In cases where a criminal act was committed using these networks, law enforcement is left with a few ways of determining the culpable parties. In case all the evidence is digitally stored in different places, and all the digital evidence is anonymized, crimes become much more difficult to solve. Further tussle occures between enforcement of copyright law and freedom of speech. This is a conflict of ideologies that is difficult if not impossible to resolve.

Due to these reasons it is not likely that the major ISPs will offer anonymity, censorship resistance or deniability enhancing services in the future. For

[6] See http://www.cs.nyu.edu/ waldman/publius/

that reason the commercial impact seems to be limited in the near term. The reason for this is possibly the fact that commercial demand for these security overlays is fairly limited in most cases. Anonymity, censorship resistance and deniability are not services that are generally required by most of the consumer population. For ISPs inside the United States of America it may also be impossible to offer such services given the regulatory requirements already in place. The purpose of the Communications Assistance for Law Enforcement Act (CALEA) is to enhance the ability of law enforcement and intelligence agencies to conduct electronic surveillance by requiring that telecommunications carriers and manufacturers of telecommunication equipment modify and design their equipment, facilities and services to ensure that they have built-in surveillance capabilities, allowing federal agencies to monitor all telephone and broadband internet traffic in real-time.

Another major barrier to adoption for end-users of a commercial offering is how to purchase such services while maintaining anonymity. Most users willing to pay for enhanced security features do not trust in the company offering the financial service. Relying on the company as a trustworthy third party conflicts with the anonymity aims of costumers.

Furthermore, security overlays also have a technical impact. We want to mention two examples. Firstly, they represent a significant challenge to the field of computer forensics, already struggling to attribute network activity or content found on a server of an individual. Establishing evidence in a court of law is difficult when a user can claim and prove technical deniability. Secondly, overlays also represent an interesting challenge to notions of the geographic origin of traffic. Many services on the Internet deliver different content depending on the geographical origin of the request. Users in Germany are often not allowed to stream movies from providers within the United States. Another example was [7] google delivering very different hits while searching for “Tiananmen Square” from inside China or Germany.

5.4 Conclusion

The Internet emerged as an overlay on the telephone system and triggered a massive shift in the structure of the telecommunication industry, with economic, policy and social implications. We believe that overlay systems on top of the internet may signal yet another shift. From our point the future has to show how far-reaching and dramatic this change will be. Sometimes change happens overnight. However, the emergence and rise of powerful and commonly used new network overlays takes some time.

Overlays exist for several reasons and they are all able to evolve the internet, which this paper has tried to illustarte. Every single motivation behind the

[7] On 13 January 2010 google stopped censoring itself in China. See http://www.guardian.co.uk/technology/2010/jan/12/google-china-ends-censorship.

overlay examples provided may force open the Internet as we know it. To evolve the internet, to develop new strategies and mind blowing possibilities - these are general reasons why people develop overlay networks. However, every project has its very own motivation. One reason for overlays is that specialized groups of users have specialized niche requirements. Sometimes a single program running on the local computer cannot fulfill these requirements but some functions distributed across the internet. In the context of going backwards in case of not going forwards another motivation is that overlays can allow the early deployment of new applications. This opens the internet as a test range for net generation applications. A final reason for overlays is that they capture an intrinsic tension between the interest of different parties. As we focused on the impact of security overlays in the last part of this paper, overlays that allow to communicate anonymously are a perfect example.

Today's Internet has little borders in terms of geographical distribution. And those existing are about to be torn down by onion routing overlays like Tor. There is almost no way to keep content back from a group of users either. The most social communities are not limited to special groups of people. Normally everyone who wants to take part in a community is allowed to do so. It is a great feature that anyone can talk to anyone thanks to the Internet. Overlays may be a means to build "gated communities" in the cyberspace, where like-minded participants agree to talk only among themselves while others are excluded. This scenario is also thinkable in terms of research when only a few laboratories gain access to a set of services connected by an overlay, whereas others depend on a reduced set of possibilities. Whether this happens or not and what it might mean for the future of the Internet, should be a topic of further observation and discussion.

An overlay may serve several goals simultaneously and may evolve over time. For example today's niche of content delivery overlays, as has been shown as a part of the second section, may evolve into basic infrastructure over time. In conclusion we can say that overlays are still a source of disruptive innovations and that there is no end or final direction to foresee.

Bibliography

[1] A. Pathan and R. Buyya. *"A Taxonomy and Survey of Content Delivery Networks"*, Grid Computing and Distributed Systems (GRIDS) Laboratory, University of Melbourne 2006.

[2] A. Tanenbaum, D. Wetherall. *"Computer Networks"*, 5th edition, Boston 2011.

[3] B. Chandra, M. Dahlin, L. Gao, A. Nayate. *"End-To-End WAN Service Availability"*, IEEE/ACM Transactions on networking, Vol. 11, No. 2, April 2003.

[4] C. Labovitz, A. Ahuja, F. Jahanian. *"Experimental study of Internet stability and backbone failures"*, International Symposium on Fault-Tolerant Computing (FTCS), June 1999.

[5] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, B. Weihl. *"Globally Distributed Content Delivery"*, IEEE Internet Computing, September - October 2002.

[6] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, S. Surana. *"Internet Indirection Infrastructure"*, University of California, Berkeley 2002.

[7] V. Paxson. *"Measurements and analysis of end-to-end Internet dynamics"*, University of California, Berkeley 1997.

[8] M. Di Raimondo, R. Gennaro. *"New Approaches for Deniable Authentication"*, CCS'05, November 07 - 11, Alexandria 2005.

[9] D. Doval, D. O'Mahony. *"Overlay Networks a Scalable Alternative for P2P"*, IEEE Internet Computing, Juliy - August 2003.

[10] D. Clark, B. Lehr, S. Bauer, P. Faratin, R. Sami, J. Wroclawski. *"Overlay Networks and the Future of the Internet"*, Communicatiosn & Strategies, no. 63, 3rd quarter 2006.

[11] S. Tarkoma. *"Overlay Networks - Toward Informations Networking"*, Auerbach Publications, Boca Raton 2010.

[12] M. Waldman, A. Rubin, L. Cranor. *"Publius: A robust, tamper-evident, censorship-resistant web publishing system"*, 9 $^{\text{th}}$ USENIX Security Symposium, August 2000.

[13] D. Andersen, H. Balakrishnan, F. Kaashoek, R. Morris. *"Resilient Overlay Networks"*, Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge 2001.

[14] A. Parekh. *"Routing on Overlay Networks"*, University of California, Berkeley, October 2002.

[15] A. Rowstron, P. Druschel. *"Storage management and caching in PAST, a large-scale, persistent peer-to-peer storage utility"*, 18 th Symposium on Operating Systems Principles, November 2001.

[16] M. Lemley, L. Lessig. *"The End of End-to-End: Preserving the Architecture of the Internet in the Broadband Era"*, University of California, Berkeley 2000.

[17] Y. Zhang, V. Paxson, S. Shenkar. *"The stationarity of Internet path properties: Routing, loss, and throughput"*, AT&T Center for Internet Research, International Computer Science Institute (ICSI), Berkeley 2000.

[18] R. Dingledine, N. Mathewson, P. Syverson. *"Tor: The Second-Generation Onion Router"*, 13 th USENIX Security Symposium, August 2004.

Chapter 6

Cloud Computing and Infrastructure as a Service - Why Standardization Is Needed and How It Can Be Achieved

Markus Rothmann

Cloud computing is a rather new computing paradigm that is acknowledged as the future of providing IT resources by many experts. Although its idea's potential and lots of already implemented services show great promise there are some problems that prevent the concept from being widely adopted.
This paper will, after looking at what cloud computing and cloud infrastructure services are and how they work, explore which aspects of this emerging paradigm lack definitions and standards. Furthermore, it will analyze how these lacks hamper the adoption of cloud services and look at the current situation. Finally, this paper states benefits of standardization in these fields and presents possible ways to achieve such standardization throughout the cloud industry.

Contents

6.1 Introduction

6.1.1 Definition of Cloud Computing

Infrastructure as a Service (IaaS) is the basic layer of a set of services offered under the superordinate concept of cloud computing. To find a valid definition of IaaS or cloud infrastructure services we first need to grasp the functionality of the greater idea. The National Institute of Standards and Technology (NIST) provides following definition:

> ”Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models.” [1]

The five essential characteristics are on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service which will be discussed further in section 6.1.3. There is no specification on how these attributes are realized. Therefore, cloud computing is no technology but rather a paradigm, moreover the name for the improvement and combination of two established concepts.
One of them is utility computing which describes business models where providers supply IT services and bill according to the client’s actual usage (on-demand, elasticity, measured service). This concept can be compared to the consumption of electricity where customers only pay what they use opposed to for example flat rates which generate constant monthly charges.
Grid computing on the other hand is the idea of connecting loosely coupled computers to a virtual supercomputer with virtual being the key word here (resource pooling, network access).

6.1.2 Functionality of Cloud Services

The development in the areas of distributed systems, virtualization and data transfer allows providers to merge these two concepts to deliver high-performance IT services. Because IaaS is the basic implementation of the cloud computing paradigm it is possible to explain how it works with the help of cloud infrastructure services. At the beginning all is based upon real hardware which can range from a small data center up to large server farms consisting of great numbers of racks. These physical computers are connected by a network and virtualized to form one large unit. Like several hard drives can be united to work as one with more capacity, computing power can also

be combined. The result is a large resource pool which can then be split into a wanted number of virtual machines of desired power. Amazon and Microsoft for example use the Xen Hypervisor and the Hyper-V respectively [2][3] where hypervisor is the name of software that allows the operation of multiple virtual machines on physical computers. This virtualization software is necessary for allocating resources to virtual machines or billing because it also monitors which client uses which resources. Therefore, the resource pool and the physical computers are separated by a virtualization layer. To stay with the simple hard drive example, two exemplars of the same size could be virtualized to form three virtual hard drives with a different storage capacity each. The virtualization software maps the I/O-requests from the virtual discs to the real ones. Although there is much more management involved when virtualizing whole computers, the hypervisors' basic concept also works for virtual machines, virtual servers or, a name often used in cloud computing, instances.
Cloud infrastructure services can be divided into two major classes, storage and computing services. Computing services supply just this - computing power - through the above mentioned instances which can be requested and used with different values of attributes like processing power or RAM. These instances can then be configured and used for example as web servers or for computing complex calculations that would take up a lot of time without high performance computing. Storage services on the other hand provide disc capacity, useable for backup or archiving reasons.

6.1.3 Characteristics of Cloud Services

The above mentioned features cloud services provide by the NIST's definition are not the only characteristics of the paradigm. Because of the virtualization layer it is possible to easily replace or repair faulty hardware without customers noticing the breakdown. When a physical machine has a defect its tasks are transferred to different machines. For that same reason the resource pool can be increased simply by adding more real machines to be virtualized without the need to shut down the whole data center. Because virtual servers can be started, stopped and resized quickly by the hypervisor the clients' instances can be altered on demand within minutes of the request. In contrast to leased data centers, requests to change operational details of cloud services do not require actions on the providers' part but can be conducted in self-service by customers through several access methods (cf. 6.2.3). The process of adjusting instances in number and power is the desired elasticity and is called scaling. Vertical scaling describes increasing the resources an instance uses and therefore provides while horizontal scaling means creating more instances. Scaling allows to deal with spikes in user load for web servers or rare needs like the above mentioned complex calculations that would otherwise cause a slowdown in any form. All these characteristics are linked to the cloud computing's virtualization aspect. The outsourcing and utility computing aspects provide benefits like the pay-as-you-go model

and less operating costs for IT personal and infrastructure. Customers also need not care about maintenance or upgrades of their IT. The most interesting features of cloud computing arise from the combination of these aspects. Cloud infrastructure services offer a level of reliability otherwise not attainable. Storage providers save redundant copies of files to prevent data loss. While instances may crash due to hard- or software failures they can easily be rebooted from images. Data produced by instances is usually stored to cloud storage and therefore not affected. Furthermore, the scaling opportunities create an impression of unlimited resources. It is practically impossible to use up all of a provider's resource pool. Because cloud infrastructure services run on remote data centers and powerful instances may generate large amounts of data it is important that providers and customers have network access with rather large bandwidth.
Figure 6.1 shows some relationships between characteristics. Each one is roughly positioned according to how much it is related to grid or utility computing (left/right) as well as whether it represents an economical or a performance-related benefit (top/bottom). Arrows depict that characteristics result from one another with virtualization (grid computing) and outsourcing (utility computing) being the base principles of their computing paradigm rather than a characteristic. Finally, the colored areas indicate a characteristic's affiliation to the earlier mentioned five essential ones of cloud computing (italic, cf. 6.1.1) although broad network access is not represented.

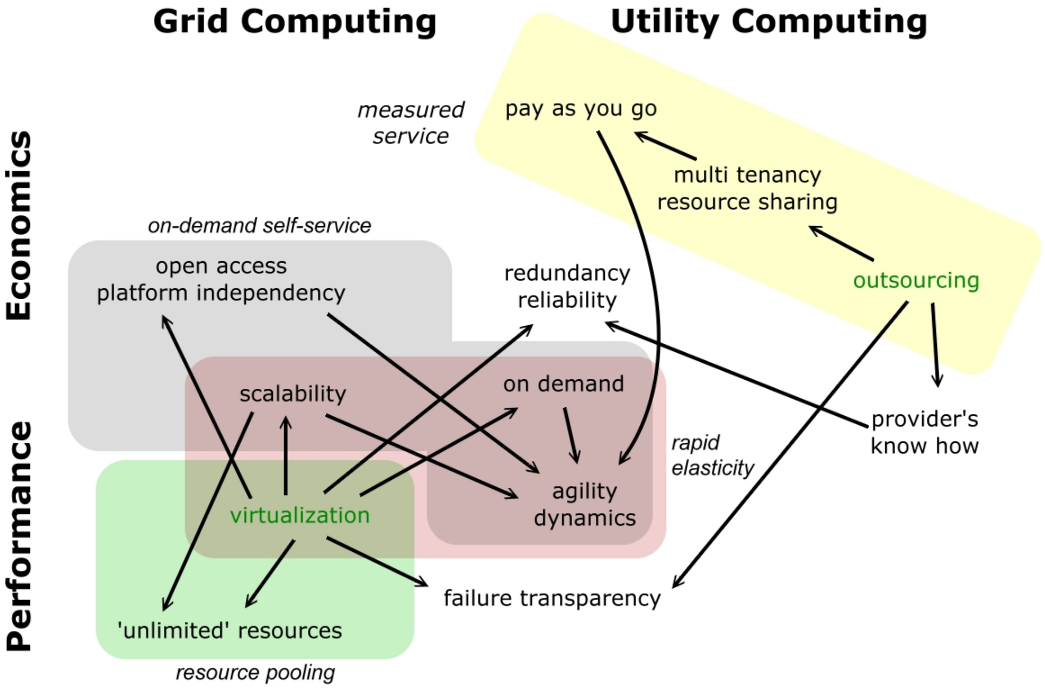


Figure 6.1: Relationships between cloud computing characteristics

Most of the above explanations were made with the help of providers and customers. One implementation of cloud infrastructure services certainly involves companies like Amazon (S3 for storage/EC2 for compute) [4], Google (Storage for Developers/-) [5], Microsoft (Windows Azure for both) [6] or Rackspace (Cloud Files/Cloud Servers) [7] setting up data centers and offering these services. But as mentioned above the virtualization concept does not depend on numerous physical hardware but can also be realized on a smaller amount of computers. There are developers that offer the necessary software to run cloud infrastructure services on any hardware. Examples for storage cloud software are NetApp StorageGRID [8] and EMC Atmos [9] as well as open source projects like ownCloud [10] or OpenStack Storage [11]. NetApp [12] as well as Permabit [13] also offer particularly suited hardware. While these products are primarily designed for storage solutions there also is software for computing clouds such as OpenStack Compute [14], Eucalyptus [15], Nimbus [16], OpenNebula [17], Nimbula [18] und Enomaly [19], just to name a few. Complete packages consisting of both hard- and software can be obtained from IBM (CloudBurst on Power Systems) [20] or HP (CloudStart) [21]. Such or any other combination of hard- and software that provides cloud services to certain set of customers is called private cloud whereas services of the before named providers are available to everybody and therefore are in the so-called public cloud. To access and utilize these services customers have to register with a provider and can then use resources according to their needs. More on the different deployment models or types of clouds can be found in the respective excursus following figure 6.2.

6.1.4 Definition of Infrastructure as a Service

Therefore, after clarifying the term of cloud computing and asserting that cloud infrastructure services are an implementation of cloud computing, a definition of IaaS can be derived from the one given in the beginning:

> Infrastructure as a Service is a subset of services following the cloud computing paradigm. These cloud infrastructure services span processing power, disc space, network connectivity as well as other fundamental computing resources and provide the following characteristics: on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service. Customers do not have to manage the underlying hardware.

6.2 Standardizable Fields of Cloud Infrastructure Services

Standards are important for technologies in various ways. William Thomson, better known as Lord Kelvin, a mathematical physicist, engineer and founder of the absolute temperature scale, once said, ”If you can't measure it, you can't improve it” [22]. The following sections will discuss the lack of standards in IaaS and partially cloud computing altogether. They will determine fields of cloud infrastructure services that need standardization, explain why standards are needed and try to find reasons for the lacking in that particular area. Furthermore, we will take a look at the current situation and how it might develop in the future.

6.2.1 Definitions and Terms

A first and very basic one of these fields is the definition of cloud services collectively. Each one of the above described ideas of cloud computing for itself can be found in other IT paradigms. Virtualization and hypervisors were used and developed as early as in the '60s amongst others by IBM [23, p.24]. The idea of utility computing can also be traced back to the '60s where John McCarthy said, ”If computers of the kind I have advocated become the computers of the future, then computing may someday be organized as a public utility just as the telephone system is a public utility” [24]. On demand services, pay-as-you-go billing and time sharing are concepts used throughout the industry since the '90s [25]. Nowadays, dedicated and shared hosting services let customers lease hardware they can use while not needing to maintain it. Virtual private servers are a related form of hosting services where hardware is managed by providers that use virtualization to run several users' servers on the same physical computers. So how does cloud computing differ from all these services? Is it legitimate to give a buzz word name to an at most combination of already in use concepts? At the beginning, the more people or committees got involved with the term the more definitions or mandatory attributes cloud services received (cf. [26]). And because there might not be the one definition of cloud computing (and IaaS respectively) there is a lot of confusion on what it actually is. Without declaring clear requirements it is hard to decide whether a service is entitled to be called a cloud service. This problem of providers falsely labeling new or renaming old products is called cloud washing, derived from the term green washing [27][28]. Today there is a consent about characteristics a service needs to have to not be ruled not a cloud service from the start (accessibility via internet, scalability and on demand usage, pay-as-you-go metering, no upfront payment, cf. [29] and definition in section 6.1.1). Because cloud computing is a paradigm rather than a technology it will remain difficult to tell whether an IT infrastructure service is IaaS or not for there are no defined techniques required. This is a major problem because many of the potential customers are only held back from switching to cloud services by the uncertainty what cloud computing really means. The obvious reason for this area of problems is the young age of the paradigm. Providers may unintentionally or deliberately interpret the word cloud differently and just as with all new concepts or technologies it takes time for people and businesses to learn about and gain trust in these new ideas before eventually adopting them. Over time there will be more and more success stories and best practices making cloud computing a trustworthy alternative to traditional IT. Although the understanding of the terms, procedures, risks and rewards only contributes partially to the predictions about usage of cloud services, one of them is shown in figure 6.2.

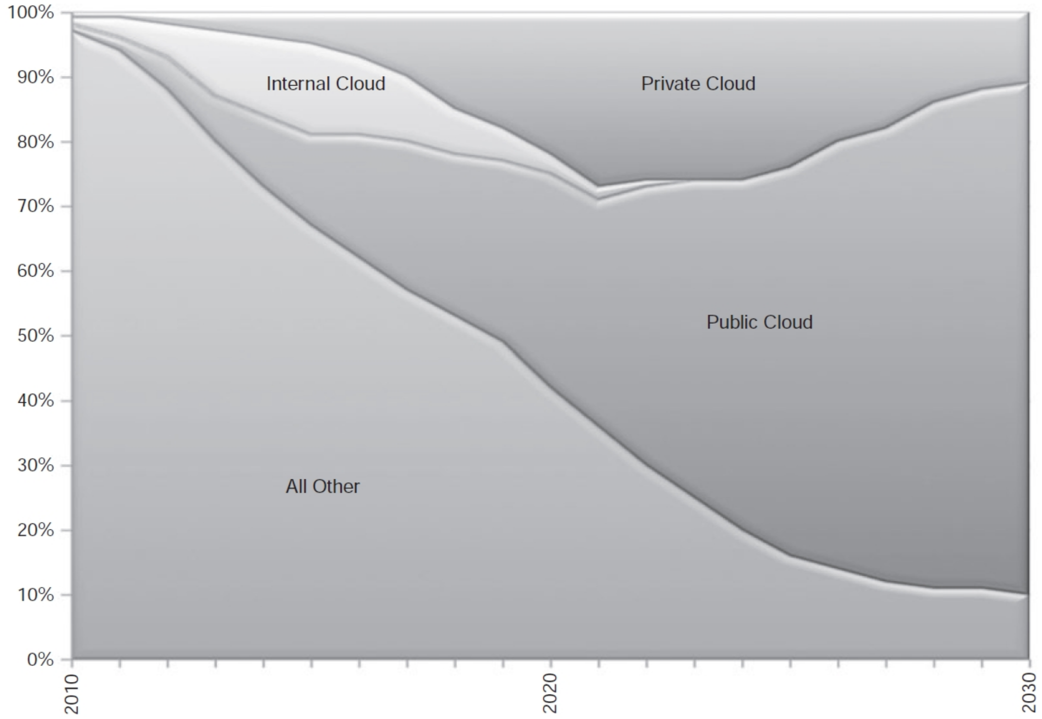


Figure 6.2: An example for predictions regarding the development of adoption of cloud services [23, p.197]

Excursus: Types of Clouds

To fully understand the diagram we will take a brief look at the types of clouds or as the NIST's definition calls them, deployment models.

The *public cloud* offers its services to whoever needs and wants them. Public cloud infrastructure services for example are Amazon EC2 and S3 as well as Rackspace Cloud Servers and Cloud Files. Any internet user can register for an account with one of those providers. Amazon for example only requires a credit card number for billing and a telephone number for calling and verifying the customer's identity.

A *private cloud* on the other hand has a restricted set of users. Its hard- and software is usually owned by its user and located within the user's network as well as behind the user's firewall, which is the actual definition of a internal cloud. Since most of the private clouds nowadays are internal clouds, the terms tend to be used interchangeable.

The third kind of cloud is a mixture of both public and private cloud services. A *hybrid cloud* describes the scenario where a user runs a private cloud but instead of solely relying on its services adds capacities of the public cloud on demand. This approach is more secure than ralying on public services alone because the private area can be used for confidential data while less endangered data can be worked with using public services. This leveraging of public services compensates the private cloud's disadvantages like limited resources and higher cost.

The prediction's authors expect a steady rise of cloud based computing as opposed to traditional solutions like data centers ("all others"). They also estimate that the share of private clouds in the cloud-based area will decline after the next ten years making public cloud services the by far most important computing method afterwards.

6.2.2 Cloud Service Contracts

The second field of lacking standards or maturity is contracting for cloud services (cf. [30]). Like in most IT services the most important criterion in cloud infrastructure services is uptime or availability. Providers have very different approaches to compensating downtime which is set out in service level agreements (SLAs). Amazon's SLA for EC2 for example states that "If the Annual Uptime Percentage for a customer drops below 99.95% for the Service Year, that customer is eligible to receive a Service Credit equal to 10% of their bill [...]" [31]. Rackspace deducts percentages from the monthly charges up to 100% for each 30 minutes of network downtime (5%), for each 30 minutes of data center infrastructure downtime (5%), other failures for its Cloud Servers and more [32]. GoGrid offers a "10,000% Service Credit" for all failures which means "a credit equivalent to one hundred (100) times Customer's fees for the impacted Service feature for the duration of the Failure. (For example, where applicable: a Failure lasting seven (7) hours would result in credit of seven hundred (700) hours of free service for the feature in question" [33].

Especially the last SLA seems very appealing but different providers exclude different reasons for downtime from their liability or calculate their uptime advantageous. Amazon's uptime is averaged over a 365 day period where if a customer has not used the service this long the remaining time is allowed for 100% up. GoGrid on the other hand excludes the following downtime from being compensated: "(1) downtime during scheduled maintenance or Emergency Maintenance [...] periods; (2) outages caused by acts or omissions of

Customer [...] (3) outages caused by hackers, sabotage, viruses, worms, or other third party wrongful actions; (4) DNS issues outside of GoGrid's control; (5) outages resulting from Internet anomalies outside of GoGrid's control; (6) outages resulting from fires, explosions, or force majeure; (7) outages to the Customer Portal, and (8) failures during a beta". Noticing that there is virtually nothing left to cause failures the "10,000% Service Credit" appears to be nothing but a advertising slogan. SLAs and contracts certainly cannot be standardized but the development of intra-industry best practices and honest approaches to liability and compensation would prevent customers falling for or noticing deceptive ones and raise the overall trust in cloud services. Gartner released a report on the topic that addresses more associated issues when contracting for cloud services [34]. One is that contracts are immature, "lack descriptions of cloud service providers' responsibilities and do not meet the general legal, regulatory and commercial contracting requirements of most enterprise organizations". Furthermore, Gartner found that many providers web link parts of the terms of use or service specifications. This referring enables changes in the pointed to parts of contracts without the customer noticing. The report concludes that contract conditions generally favor the providers. Contracts may even "contain clauses disclaiming responsibility for keeping the customer's data confidential, secure or even intact. Other clauses reserve the right to terminate accounts for a variety of reasons including apparent lack of use of the service or simply because the provider has decided to [or has to] discontinue the service" [35].

The reason for this diversity in approaches again is the young age of cloud services where providers will put their services on the market quickly. These premature and unfair contract conditions create a bit of a vicious circle. As long as customers are not too firm with what is best for their way into the cloud, providers will exploit that ignorance. But on the other hand, customers will stay cautious about that way if service level agreements don't stop being disadvantageous. Although there are already a lot of providers moving to the cloud market, there will be more of them in the future and large providers will continuously grow and take over smaller ones. Not before the market is saturated will providers have to adjust their conditions to submit to competition.

6.2.3 API Access to Cloud Services

The third area of missing standardization is the most practical one. While the prior two where based on a lack of certainty, knowledge and experience as well as the complexity and lacking maturity of the legal situation, the next field is concerned with technical possibilities to access cloud infrastructure services. To understand the issue at hand we need to take a look at access methods for cloud services [36, Ch. 2.2].

Excursus: Methods to Access Cloud Services

GUI-based Access

Almost every cloud service provider offers the possibility to access and control cloud services via a graphical user interface (GUI) that allows users to apply well-known point-and-click control. This functionality is necessary because customers cannot be expected to be tech-savvy enough or to be willing to utilize any of the later explained, advanced access methods. The key feature is the GUI and can be implemented on the providers' web sites (web GUIs), officially supplied client-side programs or independently developed third party programs. While sophisticated programs can feature advanced operations, providers' GUIs tend to be simple and straightforward or at least well-structured and therefore intuitive. They usually offer context menus, drag-and-drop and other functionalities commonly used in file explorers like Windows Explorer, Finder for Mac or Thunar/Dolphin for Linux. These GUI-based programs and web applications itself are based on API requests which are explained later on. Furthermore, third parties' programs can support cloud infrastructure services of various providers by utilizing each ones' API.

Remote Administration

Remote administration describes access methods that manage services remotely which for cloud computing usually is the case by definition. One well-established method is command line access which descends from traditional data centers where one terminal is used to manage many servers. Unlike GUI-based control this method does not offer a supporting graphical layer but accesses single servers via plain text inputs to save resources. Cloud infrastructure services can be accessed this way just as well but instead of a terminal controlling a server there are command line tools for cloud services that connect to the services. Users usually need to authenticate their rights to access the services and then can request information on the services' status, change configurations or initiate other actions. Data sent and received is often transferred through secure connections like SSL which needs to be set up if necessary.

Away from command line tools there is the method of remote desktop administration or shared desktop which is only applicable for computing services rather than storage services. It enables the user to remotely control an instance which is comparable to running and controlling a virtual machine on a computer with the difference that the virtual server runs on the provider's hardware. As example, the Windows Azure service's Management Portal, Microsoft's GUI-based web application, offers "Remote Desktop functionality [that] enables customers to connect to a running instance of their application or service in order to monitor activity and troubleshoot common problems" [37].

API Requests

The remaining three access methods describe ways for programmers to embed cloud infrastructure services. If leverages properly, these methods let

applications communicate with cloud services. These methods are used to develop and implement the above described GUI-based programs and web applications as well as command line tools.
An Application Programming Interface (API) is a set of specifications that allows communication between software. By correctly implementing the provided rules and examples for this communication, developers can let programs phrase so-called requests to prompt actions or, as the name indicates, request information. Programmers need to ensure that applications phrase syntactically and semantically correct requests. This request construction requires much effort for even a single wrong space can ruin a request. Other difficulties with this kind of access method will be picked up later.

Programming Language Libraries
Because manual phrasing of API requests is rather complex there are frameworks for programming languages called libraries that do that. Developers of these libraries define a set of language-specific but generic methods and phrase service-specific requests from the parameters given. Aside from this "encryption" most libraries also handle the sending of requests as well as the receiving and decryption of responses. To utilize such libraries, programmers of applications simply import a library to their development environment and code towards its API rather than the actual service's API which is far less complex. A setback of this access method is that there are many programming languages and even more cloud infrastructure services and technically each combination requires its own library. Some providers offer libraries for their services but most libraries are developed by independent programmer communities. Several groups and projects try to combine multiple services in so-called multi cloud libraries which on the one hand is a relief for applications that target various services but on the other hand means unnecessary ballast for programs that only utilize a sole service.

Abstraction Layer
While a program language library already embodies a layer of abstraction between generic requests and the service-specific API, there is a way to shift this abstraction layer completely out of the software development environment. Applications can be coded to send generic requests to this abstraction layer like using a web service or a remote procedure call. The transmitted information must include the type of request (what to do), the targeted provider's and its service's name or ID and, if not already saved, the user's account's credentials. The abstraction layer is a broker, possibly in form of software on a server, that handles requests and responses and that can be managed autonomously no matter how many applications depend on it.

In cases of change in a provider's service's API the programs using the last three described access methods are affected differently. When manually generating correct API requests large parts of the application's code may need reworking. The advantage of libraries is that the code still works but the appropriate library needs to be updated and the program needs to be compiled again. When using an external abstraction layer the once released software needs no revising at all because changes and updates are made to the abstraction layer. To visualize the various access methods they are depicted in figure 6.3.

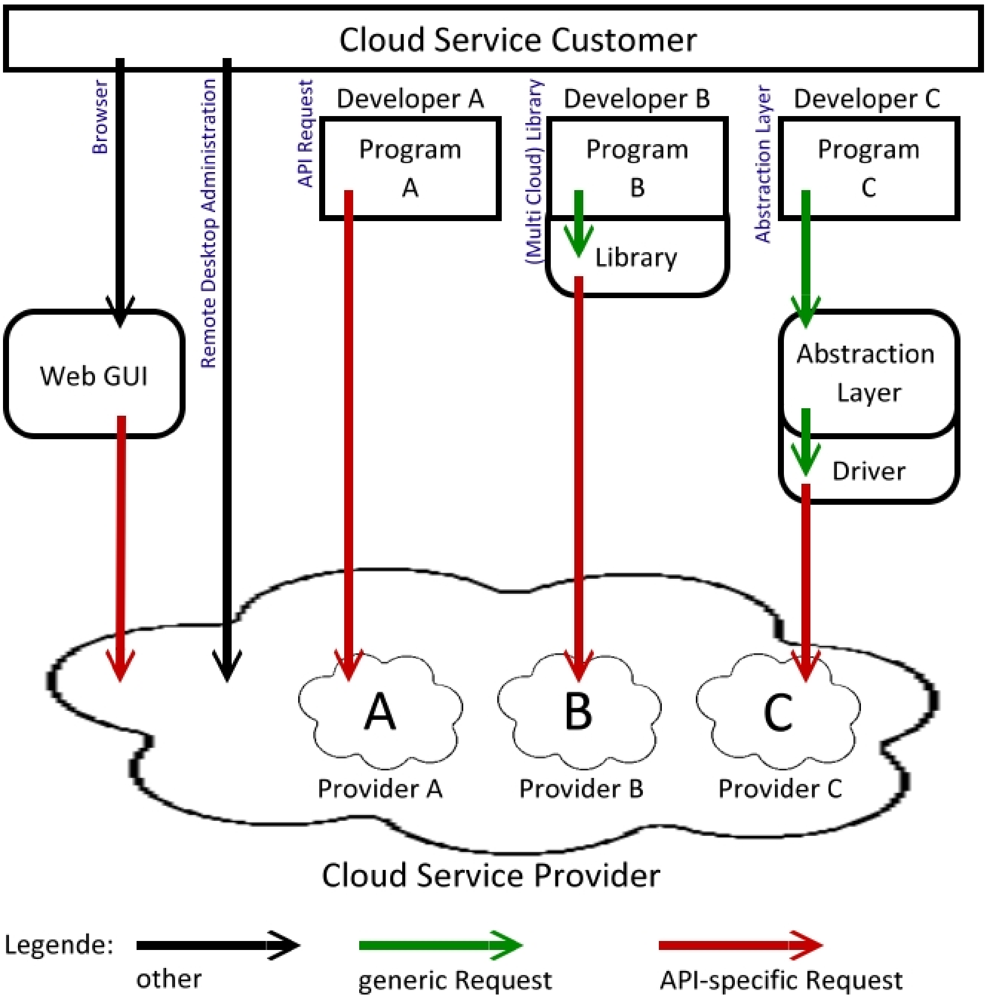


Figure 6.3: Methods to access cloud infrastructure services [36, Fig. 2.10]

Phrasing API Requests

The foundation of all access methods are the services' APIs. GUIs, command line tools, libraries and abstraction layer software only mediate between users and the providers' APIs. Several problems arise from every API being different.

A rather less fatal issue is that different APIs have varying vocabularies. A request that intents the same command is often defined differently, requires other parameters or just a different order of the given data. Without going to deep into the requests shown below it is obvious that while being structured alike the requests use different notions. Both are query requests to stop an instance of a computing service's virtual server.

```
https://ec2.amazonaws.com/
?Action=StopInstances
&InstanceId.1=<InstanceID>
&<AuthParams>
```

Request 1: Amazon EC2 [38]

```
http://provisioning.reliacloud.com:8080/client/api
?command=StopVirtualMachine
&id=<InstanceID>
&apiKey=<api_key>
&signature=<HmacSHA1-hashed-value>
```

Request 2: ReliaCloud Cloud Servers [39]

The endpoint address (red) describes the target of the request and is defined in the API specification. The key-value-pair that constitutes the requested command is in blue. The API's approach to submit the ID of the to-be-stopped instance is marked orange while grey describes authentication parameters like credentials, signatures or the request's hash value. In programming during the request generation, it is not too complicated to use different strings depending on which provider's service is supposed to be accessed. Another problem arises when not only the terms differ but also their order as a third request shows.

```
https://api.gogrid.com/api/grid/server/power
?id=<InstanceID>
&power=off
&api_key=<api_key>
&sig=<MD5 Signature>
&v=<API Version>
```

Request 3: GoGrid Cloud Servers [40]

The third issue with proprietary APIs is that the various providers offer different functions and therefore some API calls of one service cannot be mapped to another one's API because it does not support the feature. As an example for storage services, not every provider offers a API request to rename files or list all containers associated with a user's account [41, p. 26].

Transfering API Requests

Still, not only the phrasing of the requests harbors potential problems but also the transmitting of API calls is handled differently from one provider to another. Right now there are two major concepts to dispatch API requests.

One uses the SOAP standard while the other follows the REST architecture. Briefly speaking, SOAP is an XML-based communication protocol for applications and machines. A SOAP message has the below depicted basic structure [42].

```
<?xml version="version"?>
<env:Envelope>
  <env:Header>
    Header Elements
  </env:Header>
  <env:Body>
    Body Elements
  </env:Body>
</env:Envelope>
```

This SOAP envelope can be sent via Remote Procedure Call (RPC) or over the Hypertext Transfer Protocol (HTTP) where cloud infrastructure services only use HTTP.

While SOAP is a standard, REST is an architecture. So-called RESTful requests are also transmitted via HTTP but hypothetically any network protocol would do. The term Representational State Transfer originates from Roy Fieldings' dissertation in which he proposes how hypermedia systems should interact [43, Ch. 5]. It suggests server-client architecture, stateless communication, cacheable responses, uniform interfaces and layered system structures. One of the architecture's main ideas regarding cloud infrastructure services is that communication should be simple and leverage existing possibilities rather than adding new layers. While SOAP-based communication uses XML and SOAP via HTTP, RESTful APIs employ the HTTP's implemented methods like GET, PUT, POST and DELETE for requests. By using predefined methods APIs become more alike from the start in contrast to SOAP-based APIs where only the formatting is given but the methods themself need to be defined individually by developers.

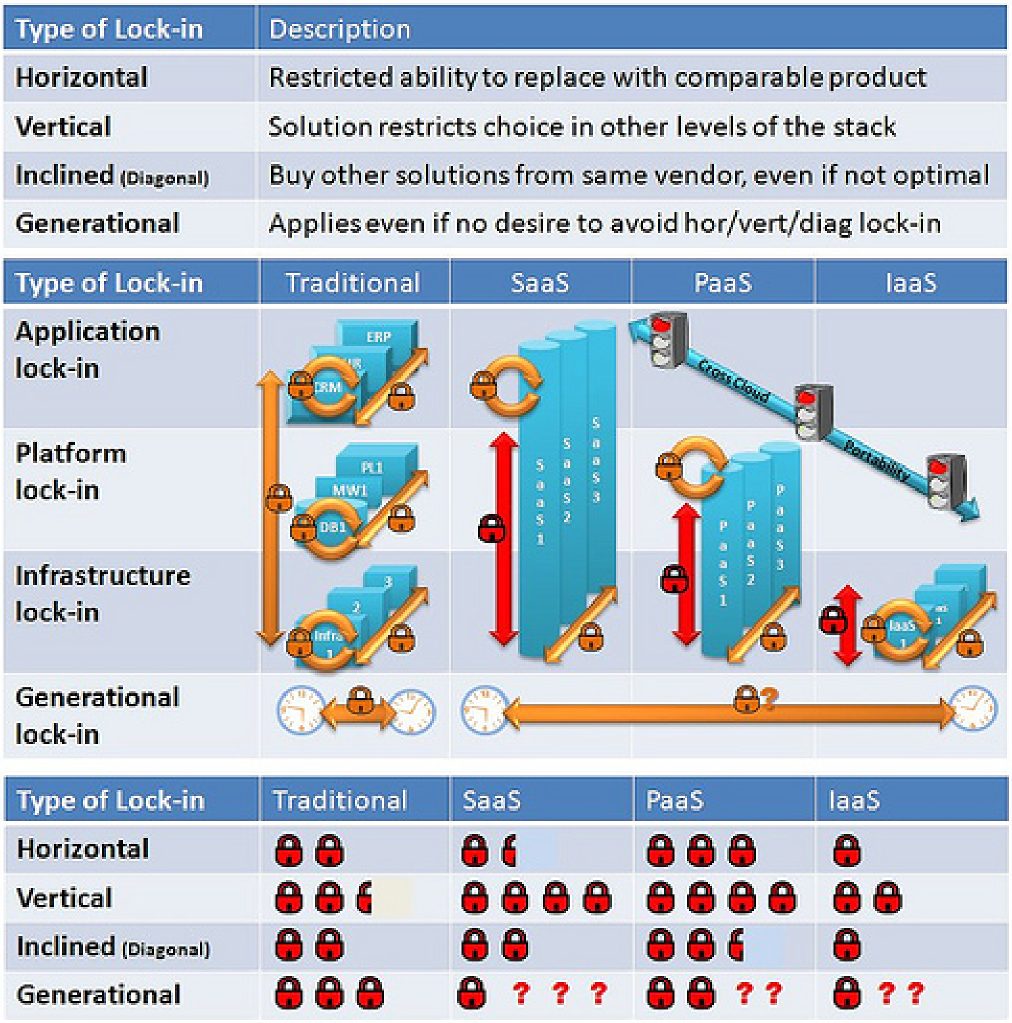
RESTful APIs are more suitable for managing cloud services for several reasons that will not be explored further in this paper. But browsing through different services' API documentations confirms that "eighty-five percent of the market is turning to REST while SOAP is fading away" [44]. The initial development of SOAP-based APIs can be explained by looking at SOAPs origins for it plays an important role in web service and RPC communication. It is an established standard many programmers are firm with and was a valid candidate to become the cloud's communication protocol. SOAP-based APIs are still available because when trends indicated that RESTful APIs are more suitable a part of the customers already deployed cloud infrastructure services using SOAP. Some users also prefer SOAP-based to RESTful APIs because of their experience with XML and SOAP but new providers often only offer RESTful APIs.

6.3 Benefits from Using Standards

The disparity in the above described field of accessing and managing cloud infrastructure services generates several problems which we will look at by listing what benefits standards will have.
The first positive effects is the decrease of potential customers' already mentioned uncertainty because they will not have to wonder which transmitting concept is better and whose API is the best to work with.
The second effect is reduced training cost for employees which no longer need to learn about different systems [27]. Whoever works with a cloud provider's service these days may be familiar with the concept of cloud computing but has to adjust in some way if switched to another service.
If services function similarly and use the same interfaces they can be exchanged more easily which creates competition on the market. Today there are many slightly varying cloud infrastructure services which makes customers base the decision which to choose at least partially on the protocols and standards used. A uniformity in this area would let customers set more value on fees, performance, reliability or other attributes.
Standardized APIs and functions of cloud infrastructure services will also simplify developers' work when programming applications for different services. Once a program is written it can be used with all services it is designed for no matter its provider. To simulate this benefit there are the earlier described attempts to conceal differences between APIs. If a programming language library can connect to several services it is called a Multi Cloud Library (MCL). Developers can call methods just like with the other libraries and simply add another parameter for the desired service. If APIs were standardized, not only would MCLs become dispensable but services could receive a common endpoint address. Which service the request needs to be sent to could be determined by the credentials alone when each provider has a unique range of account IDs. Furthermore, not only employees and programmers will benefit from standardization but also clouds will be able to interact. Hybrid cloud users can combine their own and any public services using the same managing software and behavior. This cloud interoperability may be one of the more far away goals to accomplish.

Excursus: Lock-Ins

The probably most important benefit of creating cloud interoperability is the avoidance of lock-ins. The Linux Information Project defines a lock-in or vendor lock-in as "the situation in which customers are dependent on a single manufacturer or supplier for some product (i.e., a good or service), or products, and cannot move to another vendor without substantial costs and/or inconvenience" [45]. In cloud computing today there are several kinds of lock-ins.
Horizontal lock-ins describe problems with replacing a service with a comparable one which involves issues already discussed like not being able to use an



Type of Lock-in	Description
Horizontal	Restricted ability to replace with comparable product
Vertical	Solution restricts choice in other levels of the stack
Inclined (Diagonal)	Buy other solutions from same vendor, even if not optimal
Generational	Applies even if no desire to avoid hor/vert/diag lock-in

Type of Lock-in	Traditional	SaaS	PaaS	IaaS
Application lock-in				
Platform lock-in				
Infrastructure lock-in				
Generational lock-in				

Type of Lock-in	Traditional	SaaS	PaaS	IaaS
Horizontal	🔒 🔒	🔒 🔒	🔒 🔒 🔒	🔒
Vertical	🔒 🔒 🔒	🔒 🔒 🔒 🔒	🔒 🔒 🔒 🔒	🔒 🔒
Inclined (Diagonal)	🔒 🔒	🔒 🔒	🔒 🔒 🔒	🔒
Generational	🔒 🔒 🔒	🔒 ? ? ?	🔒 🔒 ? ?	🔒 ? ?

Figure 6.4: Description and estimated influence of different types of lock-ins on cloud services [46]

application with another provider's service because of different phrasing of requests or because of different sets of functions. But other factors also prevent switching providers. Services may use different techniques for identical tasks. In cloud storage there are various ways to ensure data integrity using a hash checksum. Zetta for example uses the SHA-1 algorithm [47] while Windows Azure employs MD5 [48] which makes rewriting of a program's hashing section necessary. If using a library to access cloud infrastructure services there might not be a library for the combination of the desired programming language with the new service. The last example of a horizontal lock-in are problems with retrieving data when changing providers. Reasons may be proprietary data structures of the old service which makes it difficult to reuse data if it can be extracted at all. Applications on Salesforce.com (SaaS) are one of the cases where this issue can arise because customer data runs on Salesforce's distinct database and middleware [49].
A *vertical* lock-in on the other hand means that because of the decision to use a specific service the choice of underlying or continuative services is restricted.

A single provider's infrastructure services are most likely connectable. The Amazon computing and storage services (EC2, S3) can be used together easily because their features and APIs are concerted. Using Amazon EC2 with for example Rackspace Cloud Files will cost effort to link these services.

A consequence of vertical lock-ins are *inclined* lock-ins which describe the tendency that customers use several services of a single provider even though they are not ideal just to avoid the above mentioned complications when combining different providers' services. Other reasons for such decisions exist but are not relevant to standards or interoperability.

These three types' influence on cloud services are depicted in figure 6.4 which also presents generational lock-ins, a type not germane to this paper.

Once again, most of these issues can be avoided if API access is standardized and functions as well as functionalities converge.

6.4 Ways to Achieve Standards

6.4.1 Standardization through Following the Market Leader

There are several approaches and predictions how to achieve standards. One is the often proven observation that standards come from market leaders [23, p. 202]. The leader in cloud infrastructure services is Amazon and the mentioned prediction already commenced to an extent. Google Storage for Developers, Mezeo's Cloud Storage Platform and Dunkel, all cloud storage services, as well as Eucalyptus, a private cloud software, offer APIs that are compatible with the one Amazon provides for its S3 [50]. There are pros and cons for this development. As mentioned before, for a new competitor to be geared to the market leader makes sense because customers that worked towards Amazon's API can quickly switch providers without much effort [51]. By following that trend it would be just a matter of time until the leader's archetype becomes an official standard. But several factors oppose this development. While Amazon undoubtedly is the market leader, it is difficult to tell by how much because Amazon's trailblazer reputation and media coverage plays a big role in public perception. Cloud providers' market shares are hardly measurable today and there are many of them [52]. An often recited analogy for the current situation in cloud computing are the early railroad and postal systems with their different track sizes and addressing concepts. But following the most influential company on the choice of a specific railroad track or whether to use four or five cipher postal codes is far easier than complying with a cloud API. Matching comparable functions is simple but what about features not included in either Amazon's or the new provider's service? It is difficult to create a compatible API in the first place let alone keep it that way for APIs change continuously [53]. The bottom line is that adopting Amazon's APIs as de facto standards is very complicated and would also somewhat hamper further development of cloud infrastructure services because new providers may be discouraged from implementing new features that are not covered by Amazon's APIs. The wish for standards should not interfere with the cloud's advancement and because it is difficult to do so, assimilating to the market leader might not be the solution for the cloud [54]. The very fact that standardization on the API level is not possible yet is the reason for the already described concepts to conceal differences in API access (cf. 6.2.3).

6.4.2 Industry Consortia

The second approach is to create standards by consciously setting up consortia and working groups with the support of interested parties in form of (consulting) members to develop standards from scratch. These groups can

feed on its members’ input and already made experiences in their respective fields of profession. Additionally, the more members participate in the creation of a standard, the larger the circle of companies certain or at least likely to adopt that standard is.
To add some examples for such projects we will first look at the Storage Networking Industry Association’s (SNIA) Cloud Storage Initiative (CSI) [55]. This association of companies connected to storage networking products formed its Cloud Storage Technical Work Group to create an API for cloud storage services which is based on REST and JSON. The first version of the Cloud Data Management Interface (CDMI) was released on April 12th 2010 [56]. Since then several certification and development committees have designed reference implementations based on it and it has been cited on various cloud roadmaps [57].

Organization: Storage Networking Industry Association (SNIA) Working Group: Cloud Storage Technical Work Group (TWG) Project: Cloud Data Management Interface (CDMI)
Scope: ”The Cloud Data Management Interface defines the functional interface that applications will use to create, retrieve, update and delete data elements from the Cloud. As part of this interface the client will be able to discover the capabilities of the cloud storage offering and use this interface to manage containers and the data that is placed in them. In addition, metadata can be set on containers and their contained data elements through this interface.” [58]
Status: v1.0 released April 12, 2010 [56]

Another group that is the Distributed Management Task Force (DMTF), ”the industry organization leading the development of management standards and the promotion of interoperability for enterprise and Internet environments” [59], which also cooperates with the SNIA [59][60]. Its Cloud Management Working Group’s (CMWG) scope is to develop, validate and promote standardized methods - which the CDMI is a vital part of - throughout the cloud infrastructure services industry [61]. It primarily focuses on resource management aspects like SLAs as well as policies for utilization, monitoring and auditing.

Organization: Distributed Management Task Force (DMTF)
Working Group: Cloud Management Working Group (CMWG)

Scope: ”The CMWG will develop a set of prescriptive specifications that deliver architectural semantics as well as implementation details to achieve interoperable management of clouds between service requestors/developers and providers. This WG will propose a resource model that at minimum captures the key artifacts identified in the Use Cases and Interactions for Managing Clouds document produced by the Open Cloud Incubator.” [62, WG Scope & Charter]

Status: ”The CMWG is expected to complete public drafts of the Cloud Service Management Model and one or more Cloud Management Interface Specifications during 2011 and finalized within 2012.” [62, WG Timeline]

The third example of industry-driven standardization is the OpenStack Cloud Software developed by contributors of several well-known companies like Rackspace, Dell, AMD, Intel, Citrix and NASA [11]. Its goal is to provide an open source cloud operating system that allows everybody to run a network of arbitrary size as cloud. The open source aspect plays an important role in OpenStack's secondary goal to create standards. As discussed before, having many members ensures the usage of the developed software. But even more interesting is the thought that free of charge software will attract many users. Although the CDMI standard for example is also free to adopt and use, it is not more but an interface that needs to be implemented from both a service's user and provider. The OpenStack software in contrast sets in at the very bottom so that even just curious people can try it out and use it. The software is available under the Apache 2.0 license. Besides the initial computing and storage projects the software now also features an image service handling the management and usage of images of virtual machines [63]. If the project keeps developing at the current rate it might soon render commercial cloud infrastructure software futile.

Organization: OpenStack Cloud Software
Projects: OpenStack Object Storage (Swift), OpenStack Compute (Nova), OpenStack Image Registry/Delivery (Glance)

Scope: ”Our goal is to produce the ubiquitous Open Source cloud computing platform that will meet the needs of public and private cloud providers regardless of size, by being simple to implement and massively scalable.” [64]
”We strongly believe that an open development model is the only way to foster badly-needed cloud standards, remove the fear of proprietary lock-in for cloud customers, and create a large ecosystem that spans cloud providers.” [11]

Status: Swift - released in Oct 2010, currently v1.3.0 [63][65]
Nova - released in Oct 2010, currently v2011.2 [63][66]
Glance - released in Apr 2011, currently v2011.2 [63][67]
Next release planned for Q3 2011 [68]

6.4.3 Standard Developing Organizations

Adopting a single provider's way (first approach) or devoting to standards created under the influence of a set of companies (second approach) may make some people feel uneasy because the participating vendors' interests could be put above the ones of the industry and the customers. While this concern is most likely causeless there are independent organizations to ensure unbiased decisions regarding standards. Examples of such associations - which often work internationally and across various fields of technologies - are the Internet Engineering Task Force (IETF), the Institute of Electrical and Electronics Engineers (IEEE), the International Organization for Standardization (ISO) and its member the American National Standards Institute (ANSI). Until now we looked at several areas of lacking standards such as APIs, definitions or contracting. Now is a good time to contemplate an essential difference amongst them. Standards can be divided into two basic groups, namely prescriptive and evaluative ones [26].

Prescriptive Standards

Prescriptive standards describe details of protocols, interfaces, formats or phrasing. Familiar examples concerning cloud infrastructure services are HTTP and TCP/IP for data transportation as well as XML and JSON for data formatting. Of the above mentioned organizations the IEFT and the IEEE deal with prescriptive standards. In April 2011 the IEEE started work on two cloud projects. The Cloud Profiles Working Group (CPWG) will try to define profiles regarding different use cases and nuances of cloud infrastructure services. Instead of developing maybe one set of standardized methods for computing and storage services, the work group looks for related activities and tries to consolidate them into a number of guidelines. These profiles' outlines could be more appealing to comply with because they attend to subtypes' distinctive features and are easier to adjust to. Because the profiles will be generated for various aspects like interfaces, conventions or formats it might be possible for service providers to combine single profiles of different levels. Customers on the other hand will - after determining which specific kind of service they need - have much less potential providers to look at and could far more easily switch providers within a profile.

The Intercloud Working Group (ICWG) will research the field of cloud-to-cloud communication. Determining common and best practice topologies, protocols, functionalities and governance methods for these so-called interclouds is a field the IEEE found not addressed enough yet [69]. While the intercloud scenario basically describes the concept of cloud infrastructure service providers reassigning resources and transferring workload amongst each other to ensure performance and availability of their services [70, p. 4], the IEEE's focus lies on creating interoperability implicitly [69]. Vint Cerf, one of the developers of TCP/IP who is recognized as one of "the fathers of the internet", sees the situation of intercloud computing at the internet's

stage before protocols were adopted to allow unrestricted interoperability of participants. According to Cerf, intercloud computing could have adopted standardized interfaces and protocols in five years [71].
In the end, these interpretations of the IEEE's approach with their working groups on cloud computing are not more than just that - interpretations. Despite a call for participation almost a whole year prior to the actual start of the working groups [72] the IEEE only just started to engage the cloud. The quoted scopes in the project overviews below are the only official statements concerning the projects' goals and results should not be expected too soon as the process of discussing and defining standards usually takes several years.

Organization: Electrical and Electronics Engineers (IEEE)
Working Group: Cloud Profiles Working Group (CPWG/2301 WG)
Project: Guide for Cloud Portability and Interoperability Profiles (CPIP/P2301)

Scope: "This guide advises cloud computing ecosystem participants (cloud vendors, service providers, and users) of standards-based choices in areas such as application interfaces, portability interfaces, management interfaces, interoperability interfaces, file formats, and operation conventions. This guide groups these choices into multiple logical profiles, which are organized to address different cloud personalities." [73]

Status: Working Group announced on April 4th 2011

Organization: Electrical and Electronics Engineers (IEEE)
Working Group: Intercloud Working Group (ICWG/2302 WG)
Project: Standard for Intercloud Interoperability and Federation (SIIF/P2302)

Scope: "This standard defines topology, functions, and governance for cloud-to-cloud interoperability and federation. Topological elements include clouds, roots, exchanges (which mediate governance between clouds), and gateways (which mediate data exchange between clouds). Functional elements include name spaces, presence, messaging, resource ontologies (including standardized units of measurement), and trust infrastructure. Governance elements include registration, geo-independence, trust anchor, and potentially compliance and audit. The standard does not address intra-cloud (within cloud) operation, as this is cloud implementation-specific, nor does it address proprietary hybrid-cloud implementations. " [74]

Status: Working Group announced on April 4th 2011

Evaluative Standards

Evaluative standards analyze how well things are done and decide on benchmarks and best practices. The CDMI for example is on its way to ISO and

ANSI ratification [75], both organizations creating guidelines and reviewing other companies' compliance with them. Certified companies or workflows usually benefit from the fact that potential customers know about the requirements for such a standardized certification which attests a certain level of quality.

To create such standards for the cloud the ISO 27002 standard for ”Information technology – Security techniques – Code of practice for information security controls” [76] is currently worked on to be adapted to cloud computing [26]. Furthermore, the ISO's Joint Technical Committee 1 - which deals with information technology - established a new Subcommittee 38 in late 2009 with one part of it being the Study Group on Cloud Computing (SGCC). One of the SGCC's declared goals is to provide taxonomy and terminology for cloud computing. The group's activities also involve observing and aligning with other organizations' development to check for missed points of interest and overlaps [77][78].

Organization: International Organization for Standardization (ISO)
Working Group: JTC 1/SC 38 - Study Group on Cloud Computing (SGCC)

Scope: Terms of Reference:
1. Provide a taxonomy, terminology and value proposition for Cloud Computing.
2. Assess the current state of standardization in Cloud Computing within JTC 1 and in other SDOs and consortia beginning with document JTC 1 N 9687
3. Document standardization market/business/user requirements and the challenges to be addressed. [...]
5. Hold open meetings to gather requirements as needed from a wide range of interested organizations. [...] [79, p. 5]

Status: April 2011 - publishing of first draft of SGCC Report [79, p. 8]
August 2011 - planned due date [78]

Figure 6.5 shows the above mentioned organizations developing standards and some of their members depicted by the colored shadows.

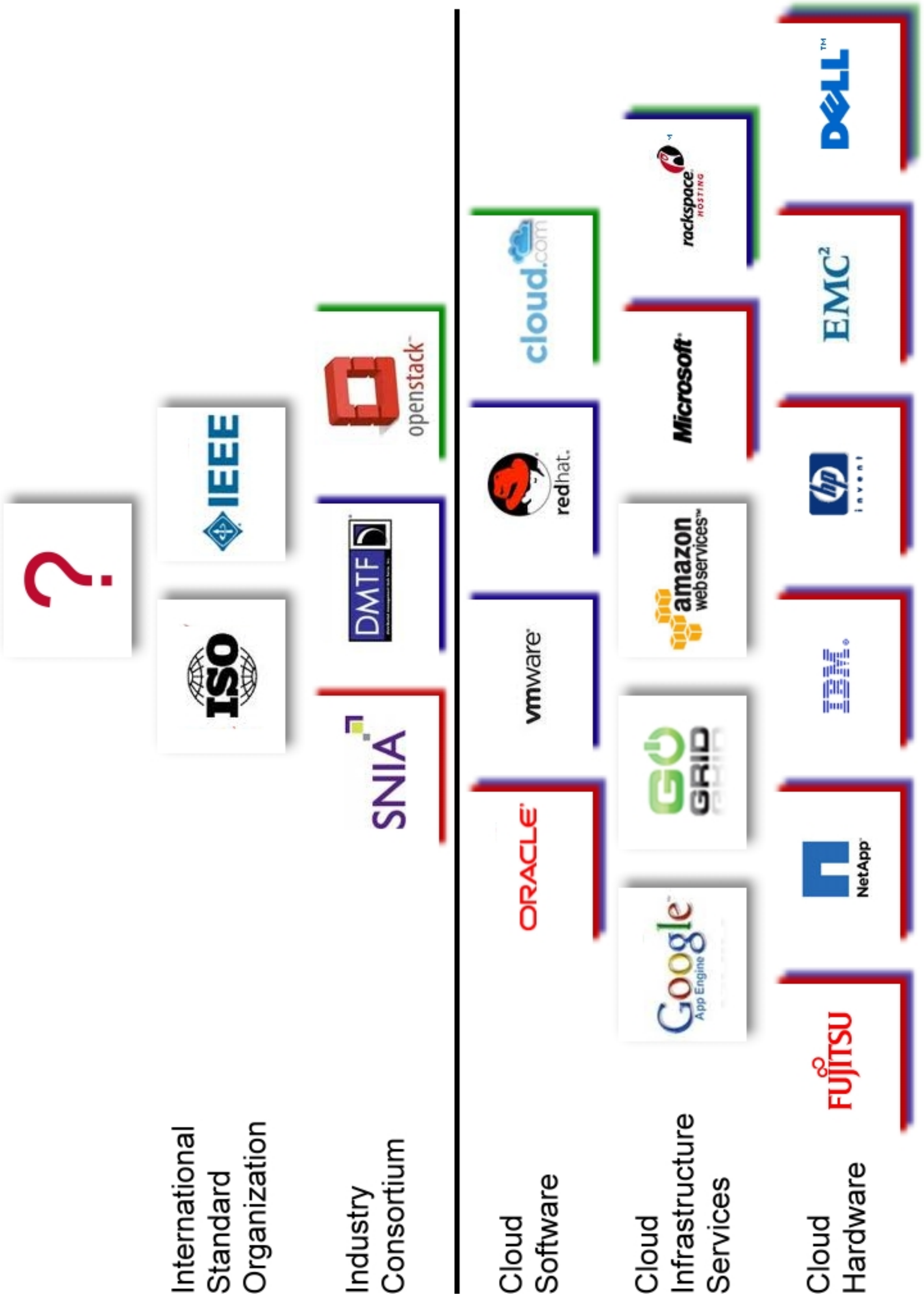


Figure 6.5: Organizations developing standards and some of their members

6.5 Conclusion

As depicted by the few above mentioned examples there are various organizations working on providing standards, taxonomies, definitions and best practices to advance the acceptance and adoption of cloud infrastructure services and cloud computing altogether [80][81]. The presented fields of lacking standards - definitions, legal concerns as well as interfaces and protocols - reflect in three important issues with cloud computing today.

Security is the most named concern when it comes to moving to the cloud [82][83]. The thought of critical data streaming through the internet outside the companies firewall-protected network scares many decision makers. Nonetheless, of the three issues we will look at this is least solvable through standardization because security in the cloud is a matter of secure and reliable protocols rather than having everybody using the same protocol. This of course does not necessarily mean that the industry cannot use the same convincing security model once one is developed. Therefore, providers and customers would benefit from a strong and reliable standard.

A second issue is portability. The earlier explained problem of vendor lock-ins is mainly based on the companies' proprietary data structures and APIs. With standardized protocols, APIs and formats it will be possible to extract data from a provider and transfer it to a new service.

This lacking conformity also prevents the intercloud concept from working well. Unity at least in basic fields of cloud service interaction is a desirable and promising step towards cloud interoperability.

In conclusion it is fairly easy to observe that developing and promoting standards will severely advance cloud computing as potential customers will gain trust in the model and others already in the cloud will have less obstacles to overcome. The industry is well on its way to create and advertize the desired standards but it will take a few years until the industry will decide on a set of mutual standards. Until then, cloud service users will have to work around differences or rely on abstraction layers to undertake these workarounds.

Bibliography

[1] *Peter Mell, Tim Grance*. The NIST Definition of Cloud Computing, version 15. Technical position, NIST, Oct 2009
http://www.alsbridge.com/methodology/pdf/NIST-definition-of-cloud-computing-v15.pdf

[2] *Keith Ward*. More Azure Hypervisor Details. Online, Nov 2008.
http://virtualizationreview.com/blogs/mental-ward/2008/11/more-azure-hypervisor-details.aspx

[3] *Kenneth Hess*. Top 10 Virtualization Technology Companies. Online, Apr 2010.
http://www.serverwatch.com/trends/article.php/3877576/Top-10-Virtualization-Technology-Companies.htm

[4] Amazon Web Services. Online, 2011.
https://aws.amazon.com/products/

[5] Google Storage for Developers - Google Code. Online, 2011.
https://code.google.com/apis/storage/

[6] Windows Azure Platform Features. Online, 2011.
https://www.microsoft.com/windowsazure/features/

[7] Cloud Computing, Cloud Hosting & Online Storage by Rackspace Hosting. Mosso is now the Rackspace Cloud. Online, 2011.
http://www.rackspace.com/cloud/cloud_hosting_products/

[8] StorageGRID - Object Storage Software - Products. Online, 2011.
http://www.netapp.com/us/products/storage-software/storagegrid/

[9] EMC Atmos - Cloud Storage, Cloud Services. Online, 2011.
http://www.emc.com/storage/atmos/atmos.htm

[10] *Falko Benthin*. ownCloud 1.1 mit neuem Plugin-System. Online, Nov 2010.
http://www.pro-linux.de/news/1/16441/owncloud-11-mit-neuem-plugin-system.html

[11] OpenStack Open Source Cloud Computing Software. Online, 2011.
http://openstack.org/index.php

[12] NetApp - NetApp Storage Systems - Storage Systems - Products. Online, 2011.
http://www.netapp.com/us/products/storage-systems/

[13] Cloud Storage Solutions by Permabit - Cloud Storage Specs. Online, 2011.
http://www.permabit.com/products/cloud-storage-specs.asp

[14] OpenStack Compute. Online, 2011.
http://www.openstack.org/projects/compute/

[15] Developers Callout - Eucalyptus Community. Online, 2011.
http://open.eucalyptus.com/

[16] Nimbus. Online, 2011.
http://www.nimbusproject.org/

[17] .:: OpenNebula: The Open Source Toolkit for Cloud Computing ::. Online, 2011.
http://www.opennebula.org/start

[18] Nimbula. Online, 2011.
http://nimbula.com/

[19] Enomaly: Elastic / Cloud Computing Platform: Home. Online, 2010.
http://www.enomaly.com/

[20] IBM CloudBurst on Power Systems. Online, 2011.
http://www-03.ibm.com/systems/power/solutions/cloud/cloudburst/index.html

[21] Cloud Consulting Services - HP IT-Services. Online, 2011.
http://www8.hp.com/de/de/services/services-detail.html?compURI=tcm:245-600107&pageTitle=Cloud-Consulting-Services?jumpid=ex_r61_us/en/large/tsg/go_smbcat20#

[22] *Sourya Biswas*. Cloud Computing Standards: How Important Are They? Online, Jan 2011.
http://www.cloudtweaks.com/2011/01/cloud-computing-standards-how-important-are-they/

[23] *Jothy Rosenberg, Arthur Mateos*. Cloud at Your Service - The when, how, and why of enterprise cloud computing. Book, Manning, 2011.
http://www.3keys.ch/MySL/WPF/Cloud/Manning%20The%20Cloud%20at%20Your%20Service.pdf

[24] *Simson Garfinkel*. Architects of the information society: 35 years of the Laboratory for Computer Science at MIT. Book, MIT Press, 1999.

[25] *Dan Farber*. On-demand computing: What are the odds? Online, Nov 2002.
http://www.zdnet.com/news/on-demand-computing-what-are-the-odds/296135

[26] *Nathaniel Borenstein, James Blake*. Cloud Computing Standards - Where's the Beef? Internet Computing, IEEE (Volume: 15 , Issue: 3), pages: 74-78, doi: 10.1109/MIC.2011.58, 2011
http://ieeexplore.ieee.org/stamp/
stamp.jsp?tp=&arnumber=5755603&tag=1

[27] *Abzetdin Adamov, Murat Erguvan*. The truth about cloud computing as new paradigm in IT. International Conference on Application of Information and Communication Technologies 2009, pages: 1-3, doi: 10.1109/ICAICT.2009.5372585, Dec 2009

[28] *Ernest de Leon*. 'Cloud Washing' Hits Feverish Pitch as Enterprises Migrate to the Cloud. Online. Apr 2011. http://cloudcomputing.sys-con.com/node/1785828

[29] *Mike Spink*. Will Cloud Computing Rain on your Sales Performance? Presentation, p. 10, Aug 2010.
http://www.gartner.com/it/content/1409300/1409319/
august_17_wil_cloud_computing_rain_mspink.pdf

[30] *Simon Bradshaw, Christopher Millard, Ian Walden*. Contracts for Clouds: Comparison and Analysis of the Terms and Conditions of Cloud Computing Services. Legal Studies Research Paper, Queen Mary University, London, Sep 2010.
http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1662374

[31] Amazon EC2 SLA. Online, Oct 2008.
https://aws.amazon.com/ec2-sla/#

[32] Rackspace Cloud Legal - SLA. Online, Jun 2009.
http://www.rackspace.com/cloud/legal/sla/

[33] Service Level Agreement (SLA) : GoGrid Cloud Hosting. Online, Jun 2011.
http://www.gogrid.com/legal/sla.php

[34] *Gartner*. Four risky issues when contracting for cloud services. Online, Feb 2011.
http://wistechnology.com/articles/8349/

[35] *Christopher Millard*. Cloud computing contracts and services: What's really happening? Insight from the Cloud Legal Project. Online, Mar 2011.
http://blogs.computerworlduk.com/cloud-vision/2011/03/cloud-computing-contracts-and-services-whats-really-happening/index.htm

[36] *Markus Rothmann*. Implementierungen von Cloud Computing Diensten. Bachelor Thesis, Universitaet der Bundeswehr, München, 2011.

[37] Just Released: Windows Azure SDK 1.3 and the new Windows Azure Management Portal. Online, Nov 2010.
http://blogs.msdn.com/b/windowsazure/archive/2010/11/29/just-released-windows-azure-sdk-1-3-and-the-new-windows-azure-management-portal.aspx

[38] *Amazon*. Amazon Elastic Compute Cloud API Reference (API Version 2010-11-15). Manual, Nov 2010.
http://docs.amazonwebservices.com/AWSEC2/2010-11-15/APIReference/index.html?ApiReference-query-StartInstances.html

[39] *ReliaCloud*. ReliaCloud API Reference Guide, edition 1.0. Manual, Dec 2009.
http://www.reliacloud.com/cloudservers/api/reliacloudAPIReferenceGuide.pdf

[40] API:grid.server.power - GoGrid. Online, 2010.
https://wiki.gogrid.com/wiki/index.php/API:grid.server.power

[41] *Doug Tidwell*. An Overview of the Simple Cloud API. Presentation, 2010.
http://static.zend.com/topics/An-Overview-of-the-Simple-Cloud-API.pdf

[42] *Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen, Anish Karmarkar, Yves Lafon*. SOAP Version 1.2 Part 1: Messaging Framework (Second Edition). Technical Report, W3C, Apr 2007.
http://www.w3.org/TR/soap12-part1/

[43] *Roy Thomas Fielding*. Architectural Styles and the Design of Network-based Software Architectures. Doctoral Dissertation, University of California, Irvine, 2000.
http://www.ics.uci.edu/˜fielding/pubs/dissertation/rest_arch_style.htm

[44] *Jerome Wendt*. Archiving data to cloud storage: How to choose the right cloud storage provider. Online, Aug 2009.
http://searchcloudstorage.techtarget.com/tip/Archiving-data-to-cloud-storage-How-to-choose-the-right-cloud-storage-provider

[45] Vendor Lock-in Definition. Online, Apr 2006.
http://www.linfo.org/vendor_lockin.html

[46] *Gregor Petri*. Vendor lock-in and cloud computing. Online, Jul 2010.
http://www.itsmportal.com/columns/vendor-lock-and-cloud-computing

[47] *Jeff Bell*. Hosting Primary, Unstructured Enterprise Data in the Cloud - Part 4: Comprehensive data integrity/protection. Online, Dec 2009. http://info.zetta.net/blog-series-hosting-primary-unstructured-enterprise-data-in-the-cloud-%E2%80%93-part-4-comprehensive-data-integrityprotection/

[48] *Rinat Abdullin*. Use MD5 Hashing for your Windows Azure Blob Operations. Online, Nov 2010. http://abdullin.com/journal/2010/11/8/use-md5-hashing-for-your-windows-azure-blob-operations.html

[49] *Bridget Botelho*. How to spot cloud service provider lock-in and stay nimble. Online, Oct 2010. http://searchcloudcomputing.techtarget.com/news/1522741/How-to-spot-cloud-service-provider-lock-in-and-stay-nimble

[50] *Jerry Huang*. Amazon S3 Compatible Cloud Storage - If you are the leader, you have supporters. Online, Apr 2011. http://cloudcomputing.sys-con.com/node/1804116

[51] Why an S3 compatible API? A rebuttal. Online, Nov 2010. http://smestorage.com/blog/?p=1206

[52] *Jonathan Lambert*. Cloud Computing Interoperability & The Amazon API Model. Online, Mar 2009 http://www.workhabit.com/labs/cloud-computing-interoperability-amazon-api-model

[53] *Jerry Huang*. Why Amazon Compatible API Is Not a Good Idea - Can you really be S3 compatible? Online, Nov 2010. http://cloudcomputing.sys-con.com/node/1624048

[54] *Lydia Leong*. Are multiple cloud APIs bad? Online, Aug 2009. http://blogs.gartner.com/lydia_leong/2009/08/27/are-multiple-cloud-apis-bad/

[55] Cloud Storage Initiative - Storage Networking Industry Association. Online, 2011. http://www.snia.org/forums/csi

[56] *Storage Networking Industry Association*. Cloud Data Management Interface Version 1.0. Technical position, SNIA, Apr 2010. http://snia.cloudfour.com/sites/default/files/ CDMI_SNIA_Architecture_v1.0.pdf

[57] *Engelbert Hörmannsdorfer*. SNIAs Cloud-Storage-Schnittstelle CDMI setzt sich durch. Online, May 2011. http://www.speicherguide.de/Magazin/StorageNews/tabid/114/ articleType/ArticleView/articleId/14051/SNIAs-Cloud-Storage-Schnittstelle-CDMI-setzt-sich-durch.aspx

[58] Cloud Data Management Interface (CDMI) - Storage Networking Industry Association. Online, 2011.
http://www.snia.org/cdmi

[59] Distributed Management Task Force (DMTF) - Storage Networking Industry Association. Online, 2011.
http://www.snia.org/about/alliances/dmtf

[60] SNIA/DMTF Work Register Version 1.3. Online, Mar 2010.
http://www.dmtf.org/sites/default/files/DMTF-SNIA_Work_Register1_3.pdf

[61] *George Chetcuti*. Cloud Standardization Bodies. Online, Apr 2011.
http://blogs.windowsecurity.com/chetcuti/2011/04/07/cloud-standardization-bodies/

[62] DMTF Cloud Management WG Charter Ver: 0.1.1. Online, Aug 2010.
http://dmtf.org/sites/default/files/CloudManagementWGCharter.pdf

[63] Releases - Wiki. Online, May 2011.
http://wiki.openstack.org/Releases

[64] Q&A. Online, 2011.
http://openstack.org/projects/openstack-faq/

[65] timeline : OpenStack Object Storage (swift). Online, 2011.
https://launchpad.net/swift/+series

[66] timeline : OpenStack Compute (nova). Online, 2011.
https://launchpad.net/nova/+series

[67] timeline : Glance. Online, 2011.
https://launchpad.net/glance/+series

[68] DiabloReleaseSchedule - Wiki. Online, May 2011.
http://wiki.openstack.org/DiabloReleaseSchedule

[69] *Rutrell Yasin*. IEEE Initiates Cloud Portability and Interop Specs. Online, May 2011.
http://redmondmag.com/articles/2011/04/05/ieee-initiates-cloud-portability.aspx

[70] *Global Inter-Cloud Technology Forum*. Use Cases and Functional Requirements for Inter-Cloud Computing. White Paper, Aug 2010.

[71] *Liam Tung*. IEEE moves on Cerf's "Intercloud". Online, Apr 2011.
http://www.crn.com.au/News/253490,ieee-moves-on-cerfs-intercloud.aspx

[72] *Alan Weissberger*. ITU Cloud Computing Focus group and IEEE Cloud Computing Standards Study Group- will they fill the standards void? Online, May 2010.
http://community.comsoc.org/blogs/ajwdct/itu-cloud-computing-focus-group-and-ieee-cloud-computing-standards-study-group-will-the

[73] IEEE SA - P2301 - Guide for Cloud Portability and Interoperability Profiles (CPIP). Online, 2011.
http://standards.ieee.org/develop/project/2301.html

[74] IEEE SA - P2302 - Standard for Intercloud Interoperability and Federation (SIIF). Online, 2011.
http://standards.ieee.org/develop/project/2302.html

[75] *Storage Networking Industry Association*. SNIA Cloud Storage TWG - Cloud Data Management Interface (CDMI). Presentation (Cloud Minisummit 2009), Jun 2009.

[76] ISO/IEC 27002:2005 - Information technology – Security techniques – Code of practice for information security management. Standard, Apr 2008.
http://www.iso.org/iso/iso_catalogue/catalogue_tc/
catalogue_detail.htm?csnumber=50297

[77] Draft Agenda for ISO/IEC JTC 1 SC 38 SGCC Open Meeting. Feb 2011.
http://www.dmtf.org/sites/default/files/
SGCC_Open_Meeting_Draft_Agenda_r4.doc

[78] Draft Study Group on Cloud Computing Report V.2. Request for Comments, May 2011.
http://dmtf.org/sites/default/files/ISO-IECJTC1-
SC38_N0282_Draft_Study_Group_on_Cloud_Computing_.pdf

[79] *Seungyun Lee*. ISO/IEC JTC 1 SC 38 SGCC. Presentation, May 2011.
http://dmtf.org/sites/default/files/JTC1_SC38_SGCC_r1.pdf

[80] *ITU Telecommunication Standardization Bureau*. Activities in Cloud Computing Standardization. Repository, May 2010
http://www.itu.int/dms_pub/itu-t/oth/49/01/
T49010000020002PDFE.pdf

[81] CloudStandards. Online, Jun 2011.
http://cloud-standards.org/wiki/index.php?title=Main_Page

[82] *Frank Gens*. IT Cloud Services User Survey, pt.2: Top Benefits & Challenges. Online, Oct 2008.
http://blogs.idc.com/ie/?p=210

[83] *Jeffrey Burt*. AMD Survey: Cloud Computing Is Maturing. Online, Jun 2011
http://mobile.eweek.com/c/a/Cloud-Computing/AMD-Survey-Cloud-Computing-is-Maturing-692623/?kc=rss

Chapter 7

The NewArch Project

Andreas Markus Brandl

The following document presents a summary of the research and progress made by the New Arch Project, a collaborative effort to evaluate the adequacy of today's Internet to meet the demands of tomorrow.
One of its goals was to rethink every architectural decision that shaped the Internet and to evaluate the possibility to develop a "Future Generation Internet Architecture" from scratch.
The project's basic question was the following:
"How would we make the main architectural decisions if we could now design the Internet on the basis of our knowledge today?"
From 2000 till 2003 the project participants devoted themselves to the tasks of reviewing current requirements to Internet architecture, examining the principles of the original Internet, identifying main architectural points of consideration, exploring specific issues and phrasing recommendations for further work.

Contents

7.1 Introduction

More than three decades ago the technological roots of today's Internet were developed. Decisions were made out of a vast pool of alternatives, according to momentary requirements and the at-that-time understanding of what is needed and even possible to accomplish.

It is no secret that these requirements have changed tremendously since the late 1970s, when the Internet technology was developed. Back then no one could imagine connecting more than 2 billion people or conceive the huge traffic the Internet needs to deal with in these days.

But the Internet has changed. Many small modifications have been performed in order to improve or alter functionality, generality, adaptability, or robustness, but nearly all were short-term changes to fix specific "problems" without thinking about the long-term effect or what the future network should be. But in the absence of a long-term technical roadmap, the coherence of the original architecture continues to dissolve and the Internet's effectiveness decreases or, at the worst, it no longer meets the demands placed on it by applications.

To provide a remedy, it would be necessary to set long-range directions or, in other words, to develop a new "Future Generation Internet Architecture" to guide the Internet's evolution. This leads to the NewArch Project, which was unique in its range, long-term scale and abstraction, looking at both the requirements for a future Internet and at the main aspects needed to establish a basic architecture.

Inspired by the original DARPA Internet research program, the NewArch Project used top-down protocol design, as well as design and prototyping to fulfill their tasks.

Due to limited resources, the creation of an entire new architecture was impossible, so the project was only able to perform detailed design and prototyping in selected areas. Nevertheless, the participants chose to redo the steps to develop high-level architectural abstractions by examining changes of requirements and failures of the original architecture, consulting experts in specific areas, developing new architectural principles and proposals for design principles and implementing proof-of-concept prototypes as appropriate.

The plan of work contained at first to review the current requirements for an Internet architecture, secondly to take a look at the fundamentals of the original Internet with present knowledge. Thirdly, the project participants worked out the main architectural points for consideration, followed by detailed exploration of the specific demands. The last point was to form recommendations for further work as a project result.

The work was not meant to lead to rejection of the Internet, but to explore and examine the fundamentals of it, in order to point out possible spots where there is room for ideas and improvement. Conducting research in the fields of architecture, routing, congestion control and more, the project even brought forth some prototypes of protocols to extensively test them in multiple scenarios.

7.2 Network Architecture

First of all, it is essential to know what a “network architecture” is. In computer communications, this term is generally used to describe a chosen set of principles for technical design of protocols and mechanisms. Chosen, because the decisions were deliberately made out of many alternatives, carefully selected by people knowing what requirements there are to be met. The architecture provides a guide to be able to standardize network protocols and algorithms, and its purpose is to support and make available coherence and consistency in order to ensure the compliance with the conditions set for the architecture. Architecture defines what the network is for and how it accomplishes its function. On the one hand it imposes constraints on what “rules” must be obeyed, but on the other hand it establishes freedom for the architect, if he stays inside the space created by the rules. An architecture must be clear on what is and is not specified, specifying only what is necessary and being plain on where there are no constraints.
The term “network architecture” was introduced during the Internet research phase, where a “design philosophy” emerged, to accompany the design of algorithms and protocols for the Internet protocol suite. The set of high-level design principles provided by a network architecture guides the engineering of its protocols and algorithms and typically specifies everything relevant to the creation of protocols. The architecture’s role is to guarantee the consistency and coherency of the resulting technical design, in other words that the pieces will fit together gently, and that it satisfies the requirements on network function.
Architecture represents a more general concept than a specific technical design. Whereas a technical design may evolve and change due to certain changes in requirements, its architecture may very well stay the same. An architecture is meant to last relatively long and be applicable to more generations of the technology. A famous example for that fact would be IPv4 and IPv6, different versions but both conform to the same Internet architecture.

7.2.1 Modularity

Modularity and abstraction are used in Computer Science to provide help in comprehending complex coherences. Modularity breaks a system into parts, whereas abstraction is a refinement of modularity. Abstraction makes complex mechanisms to small modules with simple interfaces, so that the complexity can no longer be seen. A popular structure is the layered model, where a lower layer mechanism provides a higher layer mechanism with functions. An example for that would be the OSI seven-layer reference model for networks.

But not every system aspect can fit into a layered model, e.g. system performance cannot be put into a layer. You could only express it as a vertical functional slice on top of horizontal layers, so even if the primary description of a system is in layers, there will also be vertical slices that also need to be regarded.

However, physical distribution for example cannot be described by slices either. Distribution has two planes, the system topology and the physical location, which are entirely different in general, e.g. two devices next to each other on a desk can be many hops away from each other in network topology.

A great advantage of modularity is block building. Architecture can be seen as a world of building blocks, modules and abstractions, which can be created, combined or re-used. Reusable parts reduce the design effort, nevertheless it is the most artful part of architecture to design a single module with a single interface to serve multiple purposes. The Internet for example is a general-purpose infrastructure as well. Its services can be used by multiple applications, this is also one of the Internet's primary requirements.

The modularity does not only affect the design, it also shapes the implementation. The modularity applies at design, at implementation and at deployment time. Parts that were undefined at design time, have undefined dependencies at implementation time. This leaves room for ideas and competition.

And that is where commerce comes into play. The architecture defines a marketplace, what products and parts can be produced, sold and operated. Only if certain protocols or mechanisms are specified, interoperability among devices can be guaranteed. The industry also tried to influence decisions in that sector, the term "critical interfaces" describes these architectural proposals.

Nevertheless architecture is also responsible for the development of a system. Architecture can support or inhibit the evolvement of a system, modularity and abstract interfaces tend to encourage change, since modules can be altered or replaced as long as the interfaces remain.

7.2.2 Interoperability

A computer network's basic job is to provide and assure interconnection and interoperability between different nodes. Everything must be specified to fulfill the task of interoperation in the desired way. Simply put, interoperation is the ability of a set of computing elements to interact successfully when connected in a specified way. For the consumer's point of view this might be enough, however, at a more detailed level, it is useful to ask why interoperation was achieved. The answer to this question leads to possibilities to maintain interoperation. Network services, called protocols, as already mentioned in the previous section, are the most usual means to achieve interoperation, namely by agreeing to a common definition of those protocols. Certain protocols are designed to bridge differences of diverse lower layers and permit translation between many services or technologies used on layers below. At the layer and above only common standards create interoperation, while below, the provided layer translation is used. A layer that links the layers above and below is called spanning layer. Real interoperation is achieved by defining and using effective spanning layers.

In the Internet protocol suite the Internet Protocol, or IP, plays the role of a spanning layer. IP supports the forwarding of simple text mail for example, depending on the Transport Control Protocol which uses the services of the IP protocol, which provides a uniform interface, whatever network technology is used. The IP spanning layer defines a basic set of services carefully designed to allow a wide range of network technologies below itself. So the spanning layer can be described as the foundation for interoperation, due to the fact that it is not relevant to the interoperation of mail how the IP is implemented. The IP defines a highly successful spanning layer, its functions and semantics are well specified. Its goal is to support many different applications above, and a wide range of network technologies below, so it could be symbolized as an hourglass, where the narrow part is the IP itself, acting as the single point of agreement with wide parts above and below.

Other examples for interoperation would be NTSC video delivery and Asynchronous Transfer Mode. All these services have this "narrow point" in common, where a variety of services are connected. If you want to compare these approaches to interoperation, you need to consider what range of technologies is supported above and below. The Internet must always support a broad range of applications because it connects computers, and computers are used for multiple purposes.

The image of the hourglass conveys that there can only be one spanning layer, but in the Internet protocol suite the IP is not the only one. SMTP is written to be independent of TCP, any reliable byte stream can be used to deliver email. This illustrates that in the real world, multiple points of narrowing can be existent in the same architecture.

But how can you find the perfect place for a spanning layer? Spanning layers must allow conversion among different feature sets, defining what is needed end to end and globally. Some sort of destination address must be provided as well, in contrast to the addressing in lower layers. Basically, a spanning layer is a specification of the end-to-end service. It controls the conversion and rules what must be preserved and what can be changed or ignored.

Spanning layers create a framework for comprehension and asserting interoperation. Two applications can interoperate if:

- they are based on common definitions at the application layer
- they use supporting services in a consistent manner
- they are based on a common spanning layer

- the range of network technologies below are within the spanning layer's reach

Spanning and conversion are key to network evolution. If spanning layers are defined to operate above and below other spanning layers, it is easy to create a new spanning layer or a new application, simply by using the already existing interfaces.

7.2.3 Function

The Internet has but one real function, to transport bits, thereby being oblivious to the meaning of the data it transports. Because of that, the only things it can do are transformations that do not alter the stored information, like encryption or data analysis. But knowledge about the transported data could increase efficiency and allow priority setting.

The specification of the Internet is minimalistic, e.g. performance is left out completely, due to the fact that packets can always be lost, reordered or corrupted. The semantics of IP are really weak, so the Internet packets can be carried by any reliable bitstream.

Today this principle of weak semantics is controverted by many people.

Negative effects like latency for example, cannot be removed once the system's architecture has been set. Perhaps it would have been better to generate stronger semantics to find more accurate solutions for the problems that arose. Another fact is that the Internet is only used for a small spectrum of services, many services it could provide are not and will perhaps never be used, so focusing on the services that have been put into practice seems the logical consequence. Nevertheless, having tighter semantic specifications could also make developing components cheaper.

Furthermore, not only the principle of minimal semantics, but also the principle of oblivious transport and the end-to-end principle, stating that the end hosts ought to implement the functions rather than intermediary nodes, are under attack. A tussle of interests among subscribers, providers and society results, which cannot be resolved easily.

The NewArch Project's proposal would be the following:

When an application becomes popular, more and more players will want to get involved. When they do, complexity raises, reliability or predictability probably decrease and the application evolves away from the original vision. Designers should have the goal of keeping the net open to new applications, because new applications and their change drive the net further on.

7.3 Project Results

These following subsections present some architectural fields the NewArch project analyzed in detail, as well as the architectural guidelines that resulted from this analysis.

7.3.1 FARA

The NewArch Project developed and prototyped a new architectural model called FARA, short for ”Forwarding directive, Association, and Rendezvous Architecture”. Using top-down architectural design, a network architecture should be designed in two stages. FARA represents the first stage, a high-level model with maximum universality, but adequate to a set of assumptions. Later, in the second stage, a complete architecture is to be created as an instantiation of the general model created in stage 1. One particular instance out of many possible was designed and prototyped from FARA and called M-FARA in order to illustrate the implications for mobility and addressing domains in FARA.

In the FARA model, packet exchange between entities substitutes the abstraction of host-to-host communication. An entity is a very abstract concept, allowing a variety of implementations.

To avoid the overloading of the IP address as both network destination for packet delivery and identifier for communication entities was a main objective in designing FARA. Instead, the use of the IP as destination identity is replaced by the notion of an association, each packet carries an association ID (AId) enabling the redirection to a particular association. The use of the IP address for packet routing is replaced by the notion of a forwarding directive (FD). Each packet carries a destination FD, providing sufficient information for forwarding and delivery, a source FD may be carried as well, to allow return packets. All forwarding actions to reach the destination entity are driven by the FD, the AId is then used to locate the association state. By separating the two roles of the current IP address into AId and FD, no single global address space is needed any more.

In the rendezvous process the FD of a destination entity and the information for the initial packet is returned. An entity contains state, allowing associations to persist over multiple packets, and has the ability to move independently within the network. The advantage of separating association and forwarding directive is a “generalized mobility”, routes can be changed at will, entities can move and certain routes can be preferred over others, with defined mechanisms of course, which are not specified in the FARA model.

For the purpose of starting associations, a rendezvous framework is needed. It consists of two parts, one part to discover an entity, returning a pair (FD, rendezvous string), similar to a DNS lookup, and a second part to initiate associations. The FD is needed to get the packet to the right entity, whereas the rendezvous string (RS) is used by that entity in order to know how to process the first packet and to start the association. The RS bears resemblance to the URL, which directs a query to a web page and can be carrying all sorts of dynamic information. The RS is meant to have the same functionality, but in a more general way. It provides instructions how to generate an RS to reach the desired entity and may require the sender to do calculations or attach local information. Both phases, discovery and initiation, are held very general and extensible to allow a variety of possibilities, and a global name space is not mandatory here as well.

However, separating the association identity from the forwarding directive raises serious security issues. As the FC can be easily spoofed, verification may be necessary to ensure that there is no intruder in the packet exchange. The FARA model delegates this problem's solution to the entities, which have many possible means of relief.

One instantiation out of a wide variety of specific architectures that could be derived from the FARA model is the M-FARA architecture. It was designed to show and exercise the mobility and addressing generality aspects of FARA, including mechanisms for addressing, forwarding, FD management and security, but due to lack of funding the issues of rendezvous and directory service were not included. In M-FARA, FDs are updated by a system of mobility agents, which have the role of rendezvous points and third parties in communication. For each entity, there is an M-agent, which is informed by the entity whenever it moves. For security, it uses the nonce system, where a pair of tokens is exchanged during association creation, which serve as credentials. If an entity moves, the two ends re-authenticate each other. A prototype of M-FARA was built and demonstrated, showing the ability to move between two addressing realms with IPv4 and IPv6 addresses.

7.3.2 Role-based architecture

In traditional network architectures, communication functions are organized into protocol layers and the metadata into protocol headers, but that leads to severe limitations and problems, including the following:

- Certain problems cannot be solved without violating the layers. Even in the base Internet architecture implicit layer violations are included.
- Layer violations introduce implicit functional dependencies, where originally modular separation had prevailed. These dependencies provoke unexpected feature interactions and as a result loss of extensibility.
- Designers often insert new functionality in between existing layers, to avoid extensive work on changing inter-layer interfaces or working implementations. This often leads to new layer violations.
- The fast proliferation of middle boxes like firewalls or NAT boxes represent a serious challenge to the architecture. These devices require

control data that can only be gained by special-case protocols, which operate out of band from the data.

As you can see, layering has serious limitations. This led to a fundamentally different approach, a non-layered architecture, called role-based architecture (RBA).

Communication in RBA is organized by using functional units called roles. A role represents a communication building block performing functions in order to forward or process packets. A role's inputs and outputs are application data payloads and controlling metadata that is required. Roles are meant to be well defined functional building blocks and can be compared to classes is object-oriented programming. An example for a role would be a "Forward role", representing the action of a router. A relatively few of well-known roles must be defined and standardized, however, the number of special-purpose or locally defined roles will probably be much greater. RBA could allow re-modularization of current protocols into smaller functional units. Further differences from a layered architecture are:

- There is no more packet header, but a "container" for variable-sized blocks of metadata, which can be modified in any order by the modular protocol units. The metadata, called role data, is divided into role-specific headers (RSHs), so control information can be carried with the data flow.
- New rules for controlling processing order and access to metadata are required. By controlling the association between program and data, interactions between the different protocol modules can be explicitly controlled.

In an idealized RBA model, all data in a packet is role data, modularized by RSHs. Because an end system does not always know what nodes a sent packet will encounter, the set of RSHs in a header may vary depending on the services requested by the client. The role abstraction is designed to be independent of the choice of nodes, enabling flexible network engineering.

Various compromises with traditional layering can be made in applying the RBA concept:

- A completely layer-free architecture could be established by replacing all protocol functions from the present link layer to the present application layer as well as all middlebox functions by roles or sets of roles.
- RBA could be applied only above a particular layer, keeping layering below that layer and trading flexibility against efficiency.
- An RBA subset might retain the link layer as a distinct layer, to match certain technological restraints at the link-layer protocols, imposed by industry groups.

- RBA could be performed only in end systems and middleboxes, in order to solve the efficiency issues with RBA by keeping the IP layer.

- RBA could also be only applied as an application layer application, leaving the transport layer untouched.

The NewArch Project concluded that it would require substantial effort to understand the potential of RBA, and that significant detailed design work would be the object of future research efforts.

7.3.3 NIRA

In today's Internet users cannot pick what routes their packets should take. Once the packet has entered the network, users cannot exercise any control over packet routing. A better alternative would give the user the ability to pick the packet route, opening up a new market. Enhanced Quality of Service (QoS), available end-to-end at an appropriate price would certainly find customers, be it to avoid "insecure" paths or to prioritize paying users.
The New Internet Routing Architecture, or short NIRA, is designed to exactly give the user the abilities outlined above. The user is able to choose domain-level routes, sequences of routers a packet traverses. A provider-rooted hierarchical addressing scheme is used for discovering routes and representing them efficiently. Top-level providers obtain unique prefixes and allocate subdivisions of these to customers. By this mechanism, a pair (source address, destination address) is able to uniquely identify a provider-level route. However, the user does not need to know the entire topology, only his way to the top-level provider. The Topology Information Propagation Protocol (TIPP) is central to NIRA, it propagates address allocation information as well as topology information to the user. What is also essential to NIRA is the Name-To-Route Resolution Service (NRRS), a service similar to DNS. The NRRS tells the user the address and topology information of the user he/she wants to communicate. Combining the information from the NRRS and the TIPP, the user can choose an initial route to contact another user. To handle route failure, TIPP notifies a user if a route failed. If the specified route is unavailable, the router will try to send a control message back to the sender of the packet. If that is not possible, the user can easily timeout the packets and quickly switch routes if there is no answer. A user can also specify arbitrary routes in the packet header, but providers are able to install policy filters to prevent illegitimate route usage.
NIRA is compatible with IPv6, uses the IPv6 header format and NIRA router's forwarding algorithm examines both the source and the destination address in order to effectively prevent source address spoofing.

7.3.4 XCP Congestion Control

The NewArch Project followed a new approach to congestion control, allowing individual flows to obtain large end-to-end throughput. Current TCP-based congestion control cannot provide a large per-flow end-to-end bandwidth-delay product, what becomes a serious limitation when more users access the Internet by high bandwidth link technologies. Two characteristics of TCP create this limitation. Firstly, TCP increases by a constant of 1 packet/round-trip time (RTT), therefore it cannot gain much spare bandwidth in little time. Secondly, the TCP's throughput is inversely proportional to the packet drop rate, so a very high throughput can almost never be achieved.

The project developed the eXplicit Control Protocol (XCP) to find a solution for TCP's weaknesses. XCP outperforms TCP in conventional environments and stays efficient, fair and stable as the bandwidth or the RTT increases. XCP routers inform the senders about the congestion at the bottleneck, and can scale to any number of flows. Many simulations prove that XCP improves the overall performance substantially, reduces the drop rate, increases utilization and decreases queuing delay. In further work it was even extended to provide QoS, providing guaranteed bandwidth service and fairness, without increasing the complexity of the routers.

7.3.5 Regions

Regions are first class objects in the network architecture. They were introduced because routing information of a whole network like the Internet is not manageable by a single instance. Internet packet routing has a two-tiered design containing routing among Autonomous Systems (ASs) and routing within each AS. Routing inside of and between ASs often differ intentionally, to address different performance problems.

To define a region is to define what characteristics distinguish each region. This may be topological or physical proximity or administrative control, for an AS, a definition may include IP address ranges, the internal routing algorithm or the administrative controller. If a user wants to be part of a region, he/she must conform to the description of that region.

Advantages from introducing generalized regions are that it allows discovery and exchange of information about individual resources through the region abstraction, significantly improving performance. Another advantage is that adaption frameworks can be designed and built for them, allowing internal structural reorganizing to adapt to changing conditions.

7.3.6 Subscription Systems

Subscription systems are mechanisms to inform subscribers in the fastest way possible about the arrival of relevant information on the Internet, after signing up for notifications on topics that are important to them. When relevant information on a specific topic arrives, it is routed to the associated subscribers. Notifications can include weather reports, current news, stock market quotes or any other news or data. Existing subscription systems can be divided into three categories:

a) **unicast systems**: Notifications are transmitted directly to subscribers via the Internet's existing forwarding mechanisms.

b) **single-identifier multicast systems**: Messages are sent through discrete message channels to which subscribers with same interests may subscribe.

c) **content-based multicast systems**: Messages are forwarded based on their text content.

No one knows which system will be most adequate to meet the Internet users' needs, however none of these systems may be able to inform millions of users with the requested information. Unicast systems seem to be inappropriate for a huge number of users, because they send numerous copies of the same message to network routers. Single-identifier multicast systems cannot handle complex subscription categories efficiently, so they do not seem to be adequate either. And content-based systems are too slow in processing notification messages, so they offer no clear solution as well.

Due to these deficiencies, the NewArch Project developed a new approach to Internet subscription systems, the Fast, Flexible Forwarding System (F3). It is designed for large-scale, complex applications and uses distributed multicast mechanisms. The F3 has two important features:

- All messages pass through preprocessors, which identify message information which is relevant to forwarding decisions and attach this information to the message headers. The messages then enter the F3 network and are forwarded based on the preprocessed headers.

- F3 routers store subscription information using a data-structure called content graphs to represent the relationships among subscription topics. By using the content graphs, the routers can determine efficiently what relationships there are between subscriptions and notifications.

F3 was simulated and its performance was compared to unicast, single-identifier, and content-based systems in a variety of scenarios. It supports the same interfaces as other subscription systems, at a significantly lower cost, representing an auspicious development in the field of Internet Subscription Systems.

7.4 New Perspectives

Members of the NewArch Project spent many hours in discussing network architectural designs and principles, where some specific results have been presented in the previous section. Furthermore, the discussions led to less concrete results, like gains in conceptual understanding and suchlike, which are described in this section.

7.4.1 Trust

Trust is an important issue when designing mechanisms for systems such as the Internet. But first of all, what is trust?

Humans use past experience, explicit information, the nature of a relationship, the role in which the other person has to be trusted and so on to evaluate or assess the trustworthiness of other people, it is a matter of judgment and emotional reaction. But a system cannot always judge what is best for its users, it can only do what it is designed to do. A user can have confidence in a system, if it is designed to do what it is supposed to do, but systems cannot be "trustworthy". The Internet is implemented and operated by people, so trust can only be based on experience with it or the assumption that the people who built it have acted in the users' best interests. Constraint, on the other hand, is somewhat opposing to trust, it normally implies that there is no or not much trust.

In order to make people trust a complex system like the Internet, past experience, explicit advice and trust in the creators is very important. Nevertheless, real people are responsible for all actions, so a system should behave the way real people do as well, it should seek constraints when trust is missing, an relax them when there is enough trust. Firewalls for example are constraints, which have a huge impact, they changed the Internet to a world where "that what is not permitted is prohibited".

Trust has a great influence on people. In a trusting environment it is more likely to find innovation and originality, so the Internet should reflect that idea. As in the real world people themselves should decide who to trust and who not to trust, the Internet of tomorrow needs a delivery model that the NewArch Project called "trust-modulated transparency". This model implies that when all communicating nodes wish, all data flow is transparent and unconstrained, but either side ought to be able to require constraints to limit the risk of damage or unexpected behavior.

What is most important, interacting parties must definitely know to whom they are talking to, or at least what role he/she plays (employee of a certain corporation, etc.). Accountability is another important dimension. If you can be held accountable for your actions, you will most certainly behave as you should. Identity theft, on the contrary, destroys the basic fabric of trust completely, so a system must be able to manage and convey identity and prevent spoofing or masquerading as another person. With a single mandatory global identity framework, it would be possible to confidently identity a person, but it should also be possible to explicitly choose to be anonymous or to create an identity which is untraceable to a real person, using a pseudonym for example. Pseudonyms can also provide a sense of trust based on past behavior. But it is vital, that you cannot hide that you are anonymous or that you are using a pseudonym, so decisions can be made based on the willingness of another party to reveal itself.

There are different approaches to verification when sending or receiving messages. One possibility is that only suitable participants in the network are allowed to send messages. The idea is to create a network with only trustworthy people and to exclude untrustworthy people, so the network can operate with few internal constraints. The problem hereby is that when the network grows bigger and bigger, it is more likely that there are insiders who are not entirely trustworthy, or it gets harder to find a uniform standard for trust. Another possibility is to make decisions about trust close to the receiver, keeping away unwelcome traffic from the receivers. In this way, trust is a local matter, an no longer a global one.

If you take email as an example, the main problems are spam and virus attacks. Approaches to solve this problem are to limit what senders can do, or to treat mail from known senders in a different way than mails from unknown senders, by sending an automatically generated mail back to unknown senders with the request to perform a single computation as part of the reply. Trust-related control makes sense for spam control and deciding whether to trust the contents of the message or not.

For applications, there are certain principles related to identity:

- Applications will need to deliver information about identity.
- There ought to be a standard format for identity, usable by any application.
- Users should be free to use any means to identify that is compliant to the standard format.
- There ought to be a range of systems available for identity assertion.
- There should be different levels of trust according to the potential to do harm.

The NewArch Project came to the proposal that the initial packet of any interaction ought to carry identity information, so a receiver knows to whom he is talking to before investing any resources. To prevent low-level attacks:

- a robust mechanism for first packet identity must be designed that prevents a receiver from being flooded by requests that require excessive action before verifying identity.

- a guard machine, taking on the risk of overload should be able to make decisions based on first packet identity.

In order to protect the Internet itself, there must be a mechanism for accountability between Autonomous Systems, which is robust enough to prevent flooding. It may be necessary to interpose a service between the end node and the untrusted foreign party, to solve problems like wasted resources and preventing low level security attacks. All firewalls and application-specific devices protecting hosts must decide their actions based on a trust specification provided by the end node, in order to implement a consistent design for trust.

7.4.2 Application Architecture

The members of the NewArch Project formed a list of techniques and goals for application designers:

- Sort out the motivations of possible participants in the application.
- Consider the importance of user empowerment: Let the users decide what services they want to use and who they want to communicate with.
- Use encryption to control what can be seen or changed and what is hidden.
- Design for the life cycle of an application: When an application grows up, the design can be changed.
- Include full application entity information: This allows servers to participate; encryption allows hiding the information from unwanted guests.
- Include a concept of identity: Allow the receiver to control how restrictive he/she wants to be and what he/she accepts.

Useful common services for applications would be the following:

- a service to manage identity
- a naming service for different sorts of entities
- encryption layers

7.4.3 Mobility

Mobility, in Internet architecture, refers to an end system or network changing its point of attachment dynamically, resulting by physical movement of the end system, renumbering of a network segment, getting a new temporary IP address from a DHCP server or switching the Internet Service Provider. This "generalized mobility" implies extra complexity and is handled as an exception. Due to the fact that it is expected that in future still the majority of Internet service points will be statically connected, the current approach of treating mobility as an exception is probably the best solution.

7.5 Requirements

7.5.1 Original Requirements

To understand Internet architecture, you need some information about the surroundings and requirements for its development. The network was developed to interconnect different networks whenever possible. The fact that the military was involved in the research explains a lot of these conditions.
A compact summary of the requirements follows.

1) **Internetworking**: Whole networks can and need to be connected.

2) **Robustness**: Communication must be upheld whenever possible.

3) **Heterogeneity**: A variety of networks must be accommodated by the Internet architecture.

4) **Distributed management**: The Internet architecture must be able to distribute its resources.

5) **Cost**: It must be cost effective.

6) **Ease of Attachment**: Host attachment must be permitted with low level effort.

7) **Accountability**: The resources used must be accountable.

7.5.2 Today's Requirements

It is vital to work out a set of goals and requirements when creating an architecture, in order to guide the development. The technical requirements, as presented in the previous section, have changed dramatically since they were articulated, and they will go on changing still. Forming a new requirement list based on the changing requirements for the Internet and the ways the Internet technology has drifted from the original architecture is necessary from time to time. This iterative process of reexamination must be redone whenever requirements change, tend to become more or less important, or when architecture limits certain applications and stops the network from evolving further.

Another important point is that due to the transition of the Internet from research project and research network towards mainstream infrastructure for everyone the requirements must be set much broader. Therefore only few of the requirements will be truly global, many of the requirements will take less effect in certain parts of the network, or even none at all. This makes the elaboration of a single hierarchical list of requirements, as in the previous subsection, more than hard. A new Internet architecture must face up to a multi-ordered requirements set that defines how, where, when and with what importance to meet certain requirements. Constructing such a "meta-requirement" will certainly be one of the most challenging aspects of designing a new architecture.

The commercialization of the Internet has sounded the bell for many new requirements, consequently tomorrow's architecture must also regard the needs and concerns of commercial providers, like traffic planning, regulation and payment for network usage or usage of special services.

Here are some important new requirements that may matter for a new architecture

- **Mobility**: Flexible, efficient, highly-dynamic mobility should be provided by Internet architecture.

- **Policy-driven Auto-Configuration**: Auto-configuration of end systems and routers should be provided, so that the end-user can configure those systems without expert knowledge.

- **Highly time-variable resources**: Resources that are highly variable over short time-scales should be supported. This could be arranged by installing switched backbone links or mobile devices that can switch the transmission medium.

- **Allocation of Capacity**: The ability to allocate capacity among users and applications must be given to users and/or network administrators. Today, allocation occurs as a result of congestion control, when all traffic slows down. But this sort of "fairness" is not always the right model. Commercially, capacity is generally diverted based on the willingness to pay, this should also be possible. For emergencies the capacity must be allocated based on priority of task. The administrator should be able to request resources from the network, which should be able to inform if the request can or cannot be met, and should the occasion arise, why it cannot be met.

- **Extremely long propagation delays**: The ability to deal with extremely long propagation delays arises mainly in the proposed Interplanetary Internet. Both delay itself and delay-bandwidth interactions complicate the architecture of a network.

The technical requirements have partially been dealt with, but there are meaningful non-technical aspects that influence the Internet design as well. Commercial drivers want to make a profit from the Internet and therefore want to influence architecture to their advantage. Legal and public policy increasingly tends to influence the Internet as well. Examples would be intellectual property law, encryption export law, police surveillance, privacy, free speech, telecommunications laws, charging and taxation. Those are all aspects of national law, varying from country to country. But the Internet is worldwide, so in different regions different regulations must be in effect. Being aware of those issues, the NewArch Project proposed to concentrate on the technical requirements, and to resolve the other requirements in detail later.

7.6 Architectural Principles

Attempting to break the Internet Architecture (IA) down into logically independent components, the NewArch Project formed a list of Architectural principles. These principles were formed at first, then reexamined, adapted and changed for the purpose of defining principles for a possible New Internet Architecture (NIA). The following section describes the project's results in this area.

7.6.1 List of Architectural Principles

The list is ordered by the principle's relevance to the architecture, starting with the most important. Furthermore, the list was divided into two parts, primary and secondary principles, where the primary principles are essential to construct the general architectural framework, and the primary principles fill in certain details.

Primary Principles

1. Multiplexing: only Packets
2. Transparency
3. Universal Connectivity
4. Immediate Delivery
5. End-to-End
6. Loose Semantics
7. Subnet Heterogeneity
8. Common Bearer Service
9. Connectionless Network
10. Global Addressing
11. Protocol Layering

Secondary Principles

12. Distributed Control
13. Global Routing Computation
14. Regions
15. Mobility
16. Security
17. Resource Allocation
18. Minimum Dependency

7.6.2 Current Internet Architecture

A more detailed clarification of the particular primary principles follows. In each case, the specific principle is displayed first, in bold letters, and then it is described.

1. Multiplexing: only packets
“The current Internet uses variable-length packets as the universal approach to multiplexing independent data streams.”
As the Internet is based on packet switching, this has a huge impact on the rest of the architecture. An alternative to packets would be messages for example, but this was explicitly rejected by the Internet’s designers, because packets pave the way for error detection, error recovery and multiplexing in a universal way.

2. Transparency
“In the absence of transmission errors, user data that is delivered to the intended receiver is delivered without modification.”
”What comes out is what goes in!” This is a basic principle of the original Internet, which is threatened or violated with increasing regularity today.

3. Universal connectivity:
“Universal connectivity is the normal state; a host can send data directly to any other host, except when explicitly prohibited by a third party, e.g. by a firewall.”

4. Immediate delivery
(a)“Connectivity is continuous.
(b)“In the absence of failures or overload, data is delivered immediately.
In packet delivery there should be no indefinite delays. In the original architecture, intermitted connectivity was excluded.

5. End-to-End principles

(a)"Generality: the network is build with no knowledge of, or support for, any specific application or class of applications."
The Internet should support as many applications as possible, therefore no restrictions should be made.

(b)"Robustness: The end nodes are responsible for communication functions that can be entirely accomplished by those end nodes."
This principle resulted from the high costs of computer circuitry in the 1970s and from the following principle (c).

(c)"Fatesharing: State that is specific to a particular data flow between communication end nodes is maintained in those end nodes. In case of a crash, the loss of such state will be coincided with loss of the communicating applications."
The end nodes are responsible for data delivery, buffering and format conversion. The network does not have mechanisms for these purposes.

6. Loose semantics

(a)"The IA contains no careful definition of the end-to-end semantics of data carriage in the Internet. The transparency principle implies that what goes in comes out; deviation from this is ad hoc."

(b)"The Internet has no model of its own performance. A host that needs to determine the performance of a path must measure it for itself."
This very vague definition was not a mistake, but it was one main aspect that led to the Internet's success, allowing the net to evolve, to adapt to change and thereby satisfy new service requirements.

7. Subnet heterogeneity

"The Internet makes minimal assumptions about subnet functionality, in order to operate over the broadest possible range of subnet technologies."
This principle should allow the Internet to operate on diverse subnet technologies. But "minimal assumptions", as mentioned in the principle, does actually mean a lot in the current Internet architecture. Naturally, the subnet technology must support the transport of packets. However, it must also provide high reliability, requirements for Quality of Service and a mechanism to match IP addresses into link-layer addresses.

8. Common bearer service

(a)"The Internet provides a connectionless service, end to end."
A standard end-to-end network-layer service called "common bearer service" is defined by the architecture. This service is connectionless, so no setup is required from a host to send packets.

(b)"The common bearer service provides at least the minimal common service, best-effort."
The term "best-effort" means packets may be lost, duplicated or reordered,

the end systems must take care of how to assure reliable delivery.
(c)"There is no access protocol for end systems (hosts).

9. Connectionless network mechanism
"The (inter-)network packet delivery mechanism is connectionless."
In the Internet, routers are used to forward packets connectionless using the best-effort service. But connectionless end-to-end does not necessarily mean connectionless within the network, too.

10. Global addressing
(a)"The Internet uses a single global address space to identify the network attachment point(s) of each node. Packet forwarding decisions are based upon these addresses."
Originally, there was one single global address space. But today, NAT boxes are used almost everywhere, so this principle is already violated.
(b)"IP addresses are overloaded as end-system identifiers."

11. Protocol layering
Internet protocols are defined using layered abstraction. Layering is realized using a last-on, first-off "stack" of protocol headers on each packet."
Layering is a powerful tool to build complex protocol systems, providing modularity, abstraction and information hiding. However, the strict layering model is also often violated today, by introducing sub-layers to create functionality which could not have been realized by the "conventional" layering model.

7.6.3 New Internet Architecture

The New Internet Architecture (NIA) is a proposal for a future Internet architecture, which concluded the NewArch Project's study on Internet architecture. The results are described below.

to 1: *unchanged*
The NewArch Project considered other possibilities than packets, but allowing more than one representation is complex and does not give greatly improved performance or functionality.

to 2: "The network will continue to support totally transparency at the wish of both ends, but it may interpose constraints when requested by the end points."
In order to provide security, it is partially necessary to introduce services in middleboxes and to abandon the strict principle of transparency. Transparency should be modulated by trust.

to 3: "A new Internet should permit two parties that want to

communicate to do so."
"The architecture should include a component that can filter traffic based on criteria specified by end nodes and administrators, including possibly DDoS suppression."
If some parties seek to block communication, the network ought to permit and support this.

to 4: ***unchanged***

to 5: "To the extent possible, mechanisms that provide constraints to protect untrusting applications should be designed so that an application that assumes a transparent network can continue to work unmodified, unless the constraints are application-specific and specified explicitly as part of the application design."
As the end-to-end principles provide an accurate allocation of responsibility in an architecture, they should not be abandoned, but adapted to change. Therefore constraints were introduced, with the purpose of restricting actions to prevent harm.

to 6: a) ***unchanged***
b) **"The network should contain tools to measure its performance and make this information available to end nodes."**

to 7: The members of the NewArch Project concluded that the subnet technologies should probably be tailored even further to the Internet as a means to provide enhanced functionality.

to 8: ***unchanged***

to 9: "The architecture should support a spectrum of contexts for forwarding packets. The network may detect when the establishment of additional context would improve the efficiency of the forwarding process and adapt accordingly."
"Explicit/path/source routing should be a sanctioned forwarding mechanism at the regional level."
For the purpose of allowing various possibilities to forward context and to explore how to make forwarding faster, the principle has been formed this way.

to 10: "Network addresses imply nothing about the identity of the entity at the end point. Entities can change their network attachment addresses without disrupting on-going communication.
"There are high-level name spaces to allow nodes to name each other."
Separating the two functions of an IP address (locator at the network layer, identifier at the transport layer) should be supported by the architecture.

to 11: Alternatives to layering (e.g. Role-Based Architecture) were explored during the project and could probably provide remedy to layer related problems.

Bibliography

[1] Developing a Next-Generation Internet Architecture, Robert Braden, David Clark, Scott Shenker, John Wroclawski, *Introductory Paper*, July 2000.

[2] Final Technical Report New Arch: Future Generation Internet Architecture, David Clark, Karen Sollins, John Wroclawski, Dina Katabi, Joanna Kulik, Xiaowei Yang, 1999.

[3] FARA: Reorganizing the Addressing Architecture, D. Clark, R. Braden, A. Falk , V. Pingali, ACM SIGCOMM 2003 FDNA Workshop, Karlsruhe, August 2003.

[4] From Protocol Stack to Protocol Heap - Role-Based Architecture, R. Braden ,T. Faber, M. Handley, HotNets-I, Princeton, NJ, October 2002.

[5] NIRA: A New Internet Routing Architecture, Xiaowai Yang, ACM SIGCOMM FDNA 2003 Workshop, Karlsruhe, August 2003.

[6] Congestion Control for High Bandwidth-Delay Product Networks, D. Katabi, M. Handley, C. Rohrs, ACM SIGCOMM 2002, August 2002.

[7] Designing for Scale and Differentiation, K. Sollins, ACM SIGCOMM FDNA 2003 Workshop, Karlsruhe, August 2003.

[8] Tussle in Cyberspace: Defining Tomorrow's Internet, D. Clark, J. Wroclawski, K. Sollins, R. Braden, ACM SIGCOMM 2002, August 2002.