

Chapter 2

Network 4 Newbies

Abstract Computer networks are the veins of the information age, protocols the language of the net.

This chapter describes the basics of networking starting with hardware going over to topology and the functionality of the most common protocols of an Ethernet/IP/TCP network up to Man-in-the-middle attacks. For all who want to rebuild or refresh their knowledge of networking.

2.1 Components

To be able to build a computer network of course you need some hardware. Depending on the kind of net you'll need cables, modems, old school acoustic in banana boxes, antennas or satellite receivers beside computers and network cards as well as router (Sect. 2.14), gateways (Sect. 2.13), firewalls Sect. 2.18, bridges (Sect. 2.15), hubs and switches.

A **hub** is just a simple box you plug network cables in and it will copy all signals to all connected ports. This property will probably lead to an explosion of network traffic. That's a reason why hubs are rarely used these days. Instead most of the time you will see **switches** building the heart of the network. The difference between a hub and a switch is a switch remembers the MAC address of the network card connected to the port and sends traffic only to the port it's destined to. MAC addresses will be explained in more detail in Sect. 2.4.

2.2 Topologies

You can cable and construct computer networks in different ways. Nowadays the most common variant is the so called **star network** (see Fig. 2.1), where all computer are connected to a central device. The disadvantage is that this device is a single point of failure and the whole network will break down if it gets lost. This disadvantage can be circumstanced by using redundant (multiple) devices.

Another possibility is to connect all computers in one long row one after the other, the so called **bus network** (see Fig. 2.2). The disadvantage of this topology is that each computer must have two network cards and depending on the destination

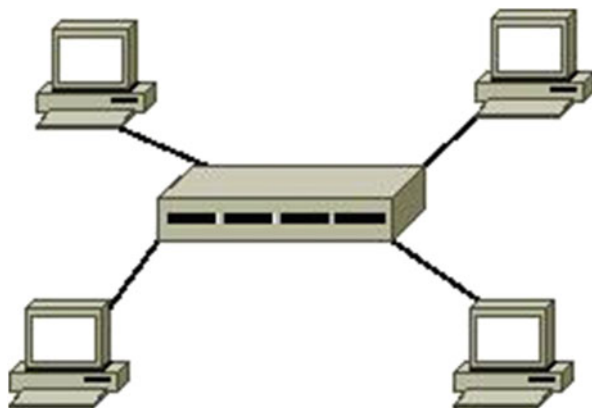


Fig. 2.1 Star network



Fig. 2.2 Bus network

the traffic gets routed through all computers of the net. If one of them fails or has too high a load the connections behind that host are lost.

The author has seen only a few bus networks this decade and all consisted of two computers directly connected to guarantee time critical or traffic intensive services like database replication, clustering of application servers or synchronization of backup servers. In all cases the reason for a bus network was to lower the load of the star network.

As last variant the **ring network** (Fig. 2.3) should be mentioned, which as the name implies connects all computers in a circle. The ring network has the same disadvantages as a bus network except that the network will only fail partly if a computer gets lost as long as the net can route the traffic the other way round. The author has not seen a productive ring network, but some wise guys whisper that it is the topology of backbones used by ISPs and large companies.

Additionally one often reads about **LAN** (Local Area Network), **WAN** (Wide Area Network) and sometimes even about **MAN** (Middle Area Network). A LAN is a local network that's most of the time limited to a building, floor or room.

In modern networks most computers are connected on a LAN over one or more switches. Multiple LANs connected over a router or VPN (see Sect. 2.17) are called MAN. If the network spreads over multiple countries or even the whole world like the internet than it is defined as a WAN.

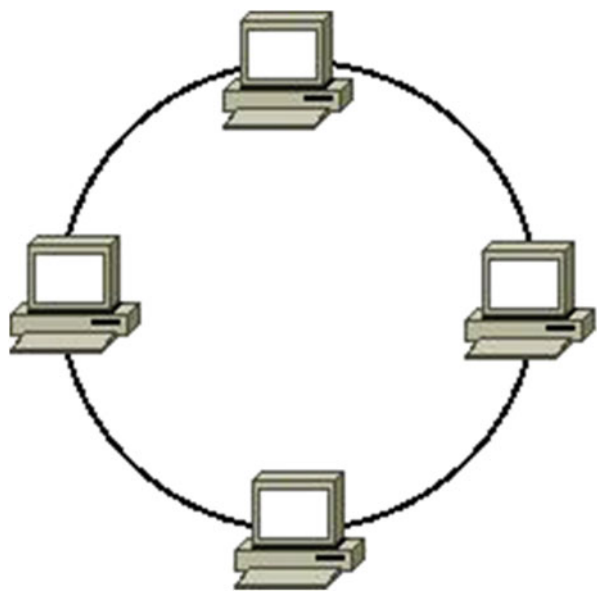


Fig. 2.3 Ring network

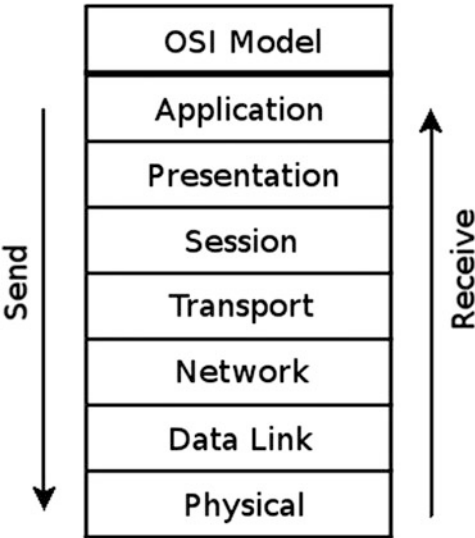


Fig. 2.4 OSI model

2.3 ISO/OSI Layer Model

According to the pure doctrine the ISO/OSI layer model, technically separates a computer network into seven layers (see Fig. 2.4).

Table 2.1 OSI layer

OSI layer	Layer name	Task
1	Physical	Cables, Antennas, etc.
2	Data-Link	Creates a point-to-point connection between two computers
3	Network	Provides for addressing of the destination system
4	Transport	Takes care that the data is received in the right order and enables retransmission on packet loss
5	Session	Used to address single applications (e.g. using ports)
6	Presentation	Conversion of data formats (e.g. byte order, compression, encryption)
7	Application	Protocols that define the real service like HTTP

Each layer has a clearly defined task and each packet passes them one after another in the operating systems kernel up to the layer it's operating on (Table 2.1).

2.4 Ethernet

Have you ever bought a “normal” network cable or card in a shop? Than the chance is nearly 100 % that you own ethernet hardware, because Ethernet is with huge margin the most used network technology today. You will see network components with different speed limits like 1, 10, 100 MBit or gigabit and an ethernet can be constructed with different cable types like coaxial (old school), twisted pair (common) or glass fiber (for data hungry guys).

Twisted pair cables can be divided into to the variations **STP** (Single Twisted Pair) and **UTP** (Unshielded Twisted Pair) as well as patch- and crossover cables.

The difference between STP and UTP cables is that the fibers of the UTP cables are unshielded and therefore they have a lower quality compared to STP cables. Nowadays new cables in a shop should all be STP.

Patch and cross cables can be separated from each other by looking at the plugs of the cable. If the colors of the fibers are in the same order than its a patch otherwise a cross cable. A **cross cable** is used to directly connect two computers, a **patch cable** is used to connect a computer to a hub or switch. Modern network cards can automatically cross the fibers so cross cables are a dying race.

Every network card in an Ethernet network has a MAC address that's world-wide unique and are used to address devices on the net. The **MAC address** consists of six two digit hexadecimal numbers, which are separated by colons (e.g. aa:bb:cc:11:22:33).

Its a common misbelief that a computer in a local TCP/IP network is reached over its IP address; in reality the MAC address is used for this purpose. Another common misunderstanding is that the MAC address cannot be spoofed. The operating system is responsible to write the MAC into the Ethernet header and systems like GNU/Linux or *BSD have possibilities in their base system to change the MAC with one command.

```
ifconfig eth0 hw ether c0:de:de:ad:be:ef
```

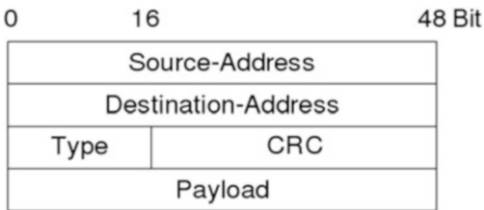


Fig. 2.5 Ethernet header

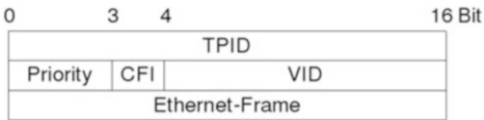


Fig. 2.6 VLAN header

Beside the source destination MAC address an Ethernet header (see Fig. 2.5) consists of a type field and a checksum. The type field defines the protocol that follows Ethernet e.g. 0x0800 for IP or 0x0806 for ARP.

Last but not least the term CSMA/CD should be explained. CSMA/CD stands for Carrier Sense Multiple Access/Collision Detect and describes how a computer sends data over an Ethernet. First of all it listens on the wire if someone is currently sending something. If that's the case it just waits a couple of random seconds and tries again. If the channel is free it sends the data over the network. Should two stations be transmitting data at the same data a collusion will result, therefore every sending station must listen afterwards to detect a collusion, than randomly wait some seconds and retransmit the data.

2.5 VLAN

A VLAN (Virtual Local Area Network) separates several networks on a logical base. Only devices on the same VLAN can see each other. VLANs were invented to define a networks structure independently from its physical hardware, to prioritize connections and to minimize broadcast traffic. They were not developed with security in mind, but its a common myth that VLANs can add to your security. Don't rely on this myth, because several ways exist to circumvent the separation of a VLAN (see Sect. 4.5).

Switches implement VLANs in two different ways: through tagging of packets using a IEEE 802.1q Header (see Fig. 2.6), that's inserted after the Ethernet header or simply defined by port. 802.1q is a newer variant, which allows the creation of a VLAN spread over several switches.

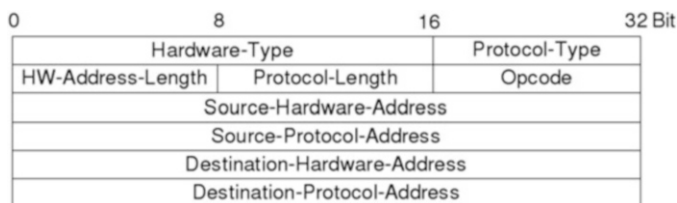


Fig. 2.7 ARP header

2.6 ARP

ARP (Address Resolution Protocol) translates between layer 2 (Ethernet) and 3 (IP). It is used to resolve MAC addresses to IP addresses. The other way round is done by RARP (Reverse Address Resolution Protocol). The structure of an ARP headers can be seen in Fig. 2.7.

Imagine a source host (192.168.2.13) tries to communicate with a destination host (192.168.2.3) for the first time than it will loudly shout over the broadcast address (see Sect. 2.7) something like the following: “Hello, here is Bob, to all, listen! I want to talk to Alice! Who has the MAC address of Alice?!”

In Ethernet speech it looks like this:

ARP, Request who-has 192.168.2.3 tell 192.168.2.13, length 28

The destination host (192.168.2.3) now shrieks up and screams “Hey that’s me!” by sending his MAC address to the requesting host (192.168.2.13).

ARP, Reply 192.168.2.3 is-at aa:bb:cc:aa:bb:cc, length 28

2.7 IP

IP like Ethernet is a connection-less protocol, that means it doesn’t know a relation between packets. It is used to define the source and destination host on layer 3, to find the (quickest) path between two communications partners by routing packets (see Sect. 2.14) and to handle errors with ICMP (Sect. 2.8). An example error is the famous host not reachable packet.

Beside that it handles fragmentation by cutting packets bigger than the MTU (Max Transmission Unit) into smaller ones. Last but not least does it implement a timeout mechanism thanks to the header TTL (Time-to-live) and such avoids endless network loops. Every host called hop a packet passes subtracts the TTL by one and if it reaches 0 it should be thrown away and the source host gets a error via ICMP.

Today there are two variants of IP IPv4 and IPv6. Both protocols differ widely and not only in size of IP addresses. IPv6 can be extended through so called optional

headers and IPv6 alone can fill a whole book. This book only covers IPv4, because its still the most common one.

An IPv4 header looks like diagram (Fig. 2.8).

First we want to see how IP network addressing works. An IPv4-address (e.g. 192.168.1.2) consists of 4 bytes divided by dots. A byte is equal to 8 bit therefore each number of an IPv4 address can be 2 expand 8 or 256 in maximum, thus it starts with a zero in reality it can not be bigger than 255.

Beside an IP address every IP network node needs a netmask (the most common one is 255.255.255.0). The netmask defines the size of the net and its used to calculate the net-start-address. The first IP of a net is called net-start-address, the last one is called broadcast-address, both cannot be used by hosts because they have a special functionality. Packets to the broadcast address are forwarded to every host on the network.

If a computer wants to communicate to another one over an IP network it first of all calculates its net-start-address with the use of its IP address and network mask. Let's say the computer has the IP 192.168.1.2. In binary that is:

11000000.10101000.00000001.00000010

A network mask of 255.255.255.0 in binary looks like:

11111111.11111111.11111111.00000000

Now one combines both addresses using a binary AND-operation that means every position, where both number are 1, stays 1, otherwise it is replaced with a 0. At the end you have the number of Fig. 2.9.

11000000.1010100.00000001.00000000

Calculated in decimal this is 192.168.1.0, the net-start-address.

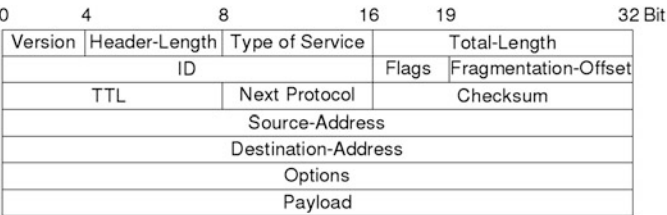


Fig. 2.8 IP-header

11000000.10101000.00000001.00000010
11111111.11111111.11111111.00000000

11000000.10101000.00000001.00000000

Fig. 2.9 Subnet-calculation

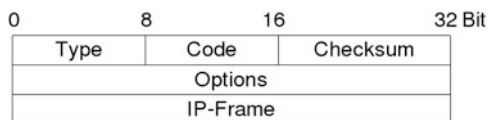


Fig. 2.10 ICMP-header

If you are not familiar with digital systems such as binary you could help yourself with a scientific calculator or a short internet search.

The netmask defines how many bits of an IP address are reserved for the net and how many for the host. In our example the first 24 bits are 1 that's the same as /24 for short, the so called CIDR block. If the complete last byte is accessible for hosts the net is classified as a class c, 2 byte make a class b, and 3 a class a otherwise the net is called a subnet.

Our example host computes the same AND-operation for the destination to obtain its net-start-address. If they differ the destination is in another network and the packet is send to the default gateway, otherwise the net is looked up in the routing table (see Sect. 2.14) and the packet is sent over the specified device or to the next router depending on its configuration.

2.8 ICMP

ICMP (Internet Control Message Protocol) is used by IP for error handling. Therefore it sets a type and a code field in its header to define the error. The header looks like in Fig. 2.10.

Most readers know the protocol for the famous ICMP echo-request packet sent by the program ping, that hopes to receive an echo-response to test if a computer is reachable and measures the network latency. Other ICMP messages include redirect-host for telling a host that there is a better router to reach his destination. The Table 2.2 lists all type and code combinations.

2.9 TCP

TCP (Transmission Control Protocol) provides session management. A new TCP session is initialized by the famous Three-Way-Handshake (see Fig. 2.13). TCP numbers all packets to ensure that they are processed in the same order they were transmitted by the source system. The destination host sends an acknowledgment to let the source know that the packet was received correctly after checking a checksum otherwise the source retransmits the packet. Last but not, least TCP addresses programs on a host by the use of ports. The port of the sending instance is called **source port** the receiving **destination port**. Commonly used application protocols

Table 2.2 ICMP codes/types

Code	Type	Name
0	0	Echo-reply
3	0	Net-unreachable
3	1	Host-unreachable
3	2	Protocol-unreachable
3	3	Port-unreachable
3	4	Fragmentation-needed
3	5	Source-route-failed
3	6	Dest-network-unknown
3	7	Dest-port-unknown
3	8	Source-host-isolated
3	9	Network-admin
3	10	Host-admin
3	11	Network-service
3	12	Host-service
3	13	Com-admin-prohibited
3	14	Host-precedence-violation
3	15	Precedence-cutoff-in-effect
4	0	Source-quench
5	0	Redirect-network
5	1	Redirect-host
5	2	Redirect-service-network
5	3	Redirect-service-host
6	0	Alternate-host-address
8	0	Echo-request
9	0	Router-advertisement
10	0	Router-selection
11	0	ttl-exceeded
11	1	Fragment-reassembly-exceeded
12	0	Pointer-error
12	1	Missing-option
12	2	Bad-length
13	0	Timestamp-request
14	0	Timestamp-reply
15	0	Info-request
16	0	Info-reply
17	0	Mask-request
18	0	Mask-reply
30	0	Traceroute-forwarded
30	1	Packet-discarded
31	0	Datagram-conversion-error
32	0	Mobile-host-redirect
		(continued)

Table 2.2 (continued)

Code	Type	Name
33	0	ipv6-where-are-you
34	0	ipv6-here-I-am
35	0	Mobile-registration-request
36	0	Mobile-registration-reply
37	0	Domain-name-request
38	0	Domain-name-reply
40	0	Bad-spi
40	1	Authentication-failed
40	2	Decompression-failed
40	3	Decryption-failed
40	4	Need-authentication
40	5	Need-authorization

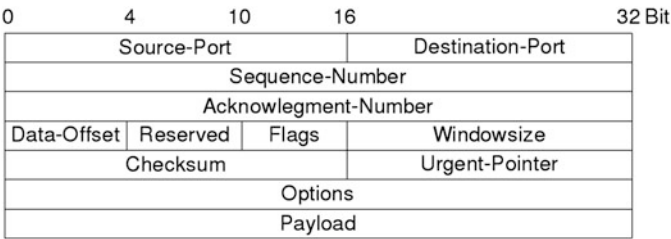


Fig. 2.11 TCP-header

like HTTP, FTP, IRC etc. have default port under 1024 e.g. a HTTP server normally listens on port 80.

A typical TCP looks like Fig. 2.11.

Beside ports one also needs to know about **TCP flags** (see Table 2.3), sequence- and acknowledgment-number and window size. Flags are used for session management to create or destroy a connection and to bid the destination to handle a packet with a higher priority.

The **Sequence-Number** is used to sort the received packets into the same order as they were send by the origin and to detect lost packets. Each packet gets an individual number that is incremented by one for every transmitted byte.

The **Acknowledgment-Number** as the name suggests acknowledges the counterpart that a packet with a certain sequence number has been received correctly. Therefore it uses the sequence number and adds one. **The Acknowledgment-number contains the next expected Sequence-Number.**

The window size defines the size of the operating systems cache of received, but not yet processed packets. A window size of zero indicates the sending station is under pressure and asks to be friendly and to slow down or even stop sending more packets until a bigger window size is received.

Table 2.3 TCP-flags

Flag	Function
SYN	Ask for a new connection
ACK	Acknowledge the receipt of a packet
RST	Cancel a connection attempt (is usually send when a host tries to connect to a closed port)
FIN	Cleanly close an established connection (must be acknowledged by the counterpart)
URG	Mark a packet as urgent
PSH	Bid the receiver to handle packet with higher priority

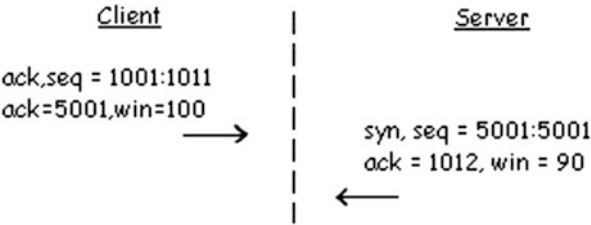


Fig. 2.12 Interaction of sequence- and acknowledgment-number

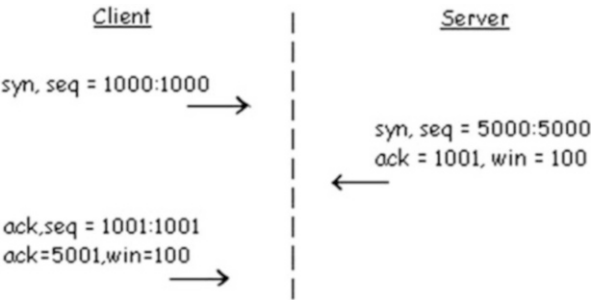


Fig. 2.13 Three-way-handshake

Beside that the window size defines the receive window. **A host accepts all packets lower than Acknowledgment-Number + Windowsize** (Fig. 2.12).

The establishment of a TCP connection is divided into three actions the **Three-Way-Handshake** (see Fig. 2.13): First of all the initiating computer sends a packet with the SYN-Flag set and to stay by our example an Initial-Sequence-Number of 1000. The Initial-Sequence-Number must be as random as possible to avoid Blind-IP-Spoofing attacks, where the attacker guesses a sequence number without being able to read the network traffic.



Fig. 2.14 UDP-header

The destination host responds with a packet where the SYN- and ACK-Flag are set. As Initial-Sequence-Number it chooses 5000 and the Acknowledgment-Number contains the Sequence-Number of the source host incremented by one (1001).

Last but not least the source host sends a final packet with set ACK- (but not SYN) flag set and uses the acknowledgment number of the SYN/ACK packet as sequence number as well as the sequence number of the previous packet plus one as acknowledgment number. This completes the Three-Way-Handshake. From now on both parties send packets with the ACK flag set. send ACK packets.

If a packets hits a closed port the destination must send a RST- Packet to be conform to RFC793. This signals the source host that the request was invalid. Lot of firewalls (see Sect. 2.18) nowadays violate this standard by either simply silently dropping the packet or even generating a bogus ICMP message. This behavior is only useful for the attacker to determine the vendor and maybe even the version of the firewall precious information for an attack.

2.10 UDP

UDP (Unified Datagram Protocol) is, like TCP, a protocol of the transport layer, but in contrast to TCP it lacks session support and is therefore classified as stateless. Further on it doesn't care about packet loss or order and only implements addressing of programs through ports. A typical UDP header can be seen in Fig. 2.14.

UDP works by the principle of "fire and forget" and is mostly used for streaming services like internet radio or television, but its also the most common used transport protocol for DNS. The advantage of UDP is the size its header adds to the packet and therefore the much higher speed.

2.11 An Example Network

An Ethernet/TCP/IP network is what you nowadays think of if you hear the term network, because it is by far the most common one. Its constructed of five layers instead of the theoretical seven layers of the ISO/OSI model. For short refreshing: **Ethernet** is on **Layer 2**, **IP** (Internet Protocol) on **Layer 3**, **TCP** (Transport Control Protocol) or **UDP** (see Sect. 2.10) on **Layer 4–6** and services like **HTTP**, **SMTP**, **FTP** on **Layer 7**.

Lets see how a HTTP packet passes all those layers one after another. In our example we want to get the index page of www.springer.com. First our computer parses the URL www.springer.com into the following components: HTTP as application protocol to be used, the hostname `www`, the domain `springer`, the Top-Level-Domain – TLD for short – (`com`) and at last the resource we try to receive in this case `/`.

Armed with these information our computer constructs the following HTTP-Header (Layer 7):

```
GET / HTTP 1.1
Host: www.springer.com
```

Next we head on to TCP (layers 4–6). It establishes a connection by the use of the Three-Way-Handshake addressing the destination port 80 (HTTP) and a random source port to connect the browser with the network.

IP (Layer 3) recognizes that it cannot use `www.springer.com` for addressing since it can only use IP addresses such as `62.50.45.35` so it makes a DNS query to resolve the IP for the hostname. We will learn more about DNS in Chap. 6. Now IP checks if the destination host is in the same network as our computer. This is not the case therefore a lookup into the routing table is necessary to retrieve the address of the next hop. There is no entry for the destination network thus the default gateway is used to send the packet to the outside world. Last but not least IP writes the address of the network card used to send the packet into the source address and our packet travels to the next layer.

On layer 2 the packet gets received by the ethernet protocol. ARP takes care about resolving the MAC address of the destination IP address and remembers them in the ARP cache this ensures it doesn't have to ask the network for every packet. Ethernet writes the MAC of the outgoing network card as source into the header and forwards the packet to the last layer (physical) in this case the driver of the network card, which will translate the packet to zeros and ones and transmit it on the medium.

2.12 Architecture

From the perspective of clients a network can have two logical structures: client/server or peer-to-peer (p2p).

A **client/server architecture** (e.g. HTTP) consists of a computer (server) that implements one or more services and another computer (client) that consumes a service.

The client sends a request and the server answers with a response if it likes the format of the request and thinks the client is authorized to ask.

In a **Peer-to-Peer-Architecture** (e.g. file sharing) all computers are equal. Everyone can admit and consume a service at the same time.

Most network connections rely on the client/server architecture.

2.13 Gateway

A gateway connects a network with one or more other networks. The most common task of a gateway is to be the so called “default gateway”, the router to whom all packets are sent, which don’t match any other local routes of a computers routing table.

Nowadays a gateway manages the connection of a local area network (LAN) with the internet and is therefore equal to a router. Some decades ago a gateway was responsible to translate between different kind of networks like Ethernet and Token-Ring.

2.14 Router

Looking at router you can differ at least two kinds: internet routers administered by your internet service provider (ISP) and home router to connect your LAN to the internet and hopefully protect you from most attacks.

Home-Router are also often called gateways, because they manage the interaction of a network with another. They receive all packets from internal hosts that should be send to some computer on the internet, write their own public IP address received from the ISP as source address into it and forwards them to the next router of the ISP.

Internet routers also forward packets, but they do so by depending on a more or less huge routing table. They don’t have a static routing table but use different protocols like RIP, OSPF and BGP to share routing information between each other and find the shortest or otherwise quickest way to the desired destination.

With the help of the command `tracert` one can determine all internet routers a packet passes between the own computer and the destination host at least if the router replies on certain packets.

```
tracert www.springer.com
tracert to www.springer.com (62.50.45.35)
 1  192.168.1.1 (192.168.1.1)  1.167 ms
 2  xdsl-31-164-168-1.adslplus.ch (31.164.168.1)
 3  * * *
 4  212.161.249.178 (212.161.249.178)
 5  equinix-zurich.interoute.net (194.42.48.74)
 6  xe-3-2-0-0.fra-006-score-1-re0.interoute.net (212.23.43.250)
 7  ae0-0.fra-006-score-2-re0.interoute.net (84.233.207.94)
 8  ae1-0.prg-001-score-1-re0.interoute.net (84.233.138.209)
 9  ae0-0.prg-001-score-2-re0.interoute.net (84.233.138.206)
10  ae2-0.ber-alb-score-2-re0.interoute.net (84.233.138.234)
11  static-62-50-34-47.irtnet.net (62.50.34.47)
12  static-62-50-45-35.irtnet.net (62.50.45.35)
```

2.15 Bridge

A bridge is a layer 2 router that's sometimes acts as a firewall.

2.16 Proxies

A proxy receives requests from a client and sends them to the destination host presuming itself would be the real source of the request. It differs to a router in acting on the layers 4–6 (TCP/UDP) till up to layer 7 (application) instead of playing on layer 3 like a router.

Most proxies additionally have the possibility to deeply understand the protocol they are working on. This way they can suppress other protocols that a client may try to speak over its port and to filter dangerous/unwanted contents like spam and malware. Furthermore a proxy could force a user to authenticate by password or smart card before he or she is allowed to use its service.

Normally a proxy must explicitly be configured by the user. A web proxy, for example, gets inserted into a browser's configuration, but a special kind of proxy exists where a router or firewall (Sect. 2.18) automatically redirects a connection through a proxy without a user realizing it. Such a proxy is called transparent proxy. Most internet service providers nowadays use such a kind of proxy at least on HTTP ports for performance reasons. The proxy caches all static web contents like images and videos on its hard disk. In some countries transparent proxies are also used to censor and observe the internet access.

Some web proxies insert a `PROXY-VIA` entry into the HTTP header and such let a user know that his connection flows over this proxies and which IP address the proxy has. The existence of this header in transparent proxy is unlikely and may be a hint for misconfiguration or a slacky sysadmin.

Interested reader could, for example, use the following script to get an overview of all HTTP information sent by its browser to every web server they use www.codekid.net/cgi-bin/env.pl

2.17 Virtual Private Networks

Virtual Private Networks (VPN) is a collection of security mechanisms, which only have in common the protection of a connection by using encryption and/or authentication. Nearly all VPNs support the possibility to secure the access to a whole network and thanks to powerful cryptology also protect against espionage and manipulation. Therefore it operates on the protocol stack either on layer 3, 4 or 7. It can be commonly said that the deeper the VPN intercepts the connection the more secure it can be, because it can prevent attacks on each layer.

Typical protocols or protocol stacks are IPsec, PPTP and OpenVPN. Mostly they are used to connect outside-agencies and to integrate roadrunner (Employees, which connect to the company network through a mobile internet connection).

2.18 Firewalls

A firewall is neither a product nor a tiny, magical box with lots of blinking LEDs even if more IT security companies try to let you think so. **A firewall is a security concept.** It serves to protect the network and computers from being attacked and is only as effective as the combination of its components.

Typical parts of a firewall are a packet filter, intrusion detection system, intrusion prevention system, log analyzer, continuous system updates, virus scanner, proxies, honeypot and/or VPNs.

A **packet filter** works on layer 3 and 4 and decides which packets shall pass, be dropped, rejected or redirected depending on its rule-set.

Intrusion detection systems can be classified into two different types: host- and network intrusion detection system. A host intrusion detection system (HIDS for short) locates successful attacks on a local computer by, for example, continuously checking all files and directories against a database of cryptographic checksums.

A network intrusion detection system (NIDS) therefore detects attacks in the network traffic and can operate on all layers at the same time. Its functionality can be compared to a virus scanner, because it searches for signatures of known attacks. Additionally it has the possibility to learn what is classified as normal traffic in a network and the anomaly detection component alarms packets that differs from it.

Attacks recognized by a NIDS can be prevented thanks to a **intrusion prevention system** (IPS). In the easiest case it just inserts the attacking IP address into a list of IPs to block and the packet filter will drop everything from them. Be careful: this isn't the best way to deal with attacks. A smart attacker could forge packets from legitimate and important systems and cut you completely from the net. Therefore it would be better to rewrite the attack packets in such a way that they cannot do any damage any more or to at least protect certain ips from being blacklisted.

A **honeypot** is a simulated server or whole simulated network of easy to crack services. Depending on its purpose it is used to keep script kiddies and crackers away from production systems, to have a prealert system and to log and analyze new cracking techniques, viruses, worm codes etc.

Last but not least the most important component: a continuous system upgrade and patch workflow! Without current security updates you will never get security at all. A firewall consists of software like a normal desktop computer.



Fig. 2.15 Man-in-the-middle attack

2.19 Man-in-the-Middle-Attacks

Man-in-the-middle attacks (Mim- or Mitm attacks for short) behave like a proxy, but on an unintentional base. Some individuals therefore consider transparent proxies of ISPs a Man-in-the-Middle attack.

All mim-attacks have in common to partly or entirely redirect the traffic of a victim to themselves and afterwards forward them to the real destination (see Fig. 2.15).

This can be realized through different techniques such as ARP-Cache-Poisoning (Sect. 4.2), DNS-Spoofing (Sect. 6.7) or ICMP Redirection (Sect. 5.10).

Not only can an attacker steal the complete traffic including sensitive data like usernames and passwords, but also drop connections at will and manipulate content to fool the victim.



<http://www.springer.com/978-3-662-44436-8>

Understanding Network Hacks

Attack and Defense with Python

Ballmann, B.

2015, XIV, 178 p. 26 illus., Hardcover

ISBN: 978-3-662-44436-8