UCLouvain

École polytechnique de Louvain

LINFO1341

Computer networks : information transfer

Olivier Bonaventure

Dylan Goffinet

2021-2022

Contents

1 Terminology **1**

2 Layers **2**
- 2.1 Physical Layer . . . 2
- 2.2 Datalink Layer . . . 2
- 2.3 Network Layer . . . 3
- 2.4 Transport Layer . . . 3
- 2.5 Application . . . 3

3 Protocols **4**
- 3.1 Datalink Layer . . . 4
  - 3.1.1 STP . . . 4
- 3.2 Network Layer . . . 5
  - 3.2.1 IPv6 . . . 5
  - 3.2.2 ICMPv6 . . . 6
  - 3.2.3 Intradomain routing . . . 7
  - 3.2.4 Interdomain routing . . . 8
- 3.3 Transport Layer . . . 8
  - 3.3.1 UDP . . . 8
  - 3.3.2 TCP . . . 8
- 3.4 Application Layer . . . 11
  - 3.4.1 DNS . . . 11
  - 3.4.2 Mails . . . 11
  - 3.4.3 HyperText Transfer Protocol (HTTP) . . . 12
  - 3.4.4 Secure Shell (SSH) . . . 12
  - 3.4.5 Transport Layer Security (TLS) 13

4 Tips **14**
- 4.1 Window size . . . 14
  - 4.1.1 Maximum window size with go-back-n and selective repeat . 14
- 4.2 HTTP . . . 14
  - 4.2.1 Role of Header's fields and what if they are deleted ? . . . 14
- 4.3 IPv6 Routing . . . 14
- 4.4 Link State . . . 14

1 | Terminology

- ADSL : Asymmetric Digital Subscriber Line (high download speed but low upload speed).
- ARP : Address Resolution Protocol.
- BPDU : Bridge Protocol Data Unit.
- CRC : Cyclical Redundancy Check.
- DNS : Domain Name System. Naming system for associating various information (such as an IP address) with a domain name.
- HTTP : HyperText Transfer Protocol.
- IEEE : Institute of Electrical and Electronics Engineers.
- LAN : Local Area Network.
- MSL : Maximum Segment Life.
- MTU : Maximum Transmission Unit Maximum size of a protocol data unit that can be communicated in a single network layer transaction.
  - Path MTU : MTU size on the network between two Internet Protocol hosts.
- PDU : Protocol data unit. Information that is transmitted as a single unit among peer entities of a computer network.
- SDU : Service Data Unit.
- TLV : Type-Length-Value. Encoding scheme used for optional information elements in a certain protocol. The first byte is a type, the second byte is the size of the value field and the rest is the value field.
- DIFS : DCF Interframe Space. If a station detects the medium has been continuously idle for a duration of DIFS, and the last frame transmission was successful (no corruption or error), it is then allowed to transmit a frame. If the last transmission contained an error, the station must wait for a duration of EIFS for any frame (re)transmission. DIFS respects the following relation : DIFS = SIFS + (2 * Slot time).
- EIFS : Extended Interframe Space. Similar to DIFS but is only activated if the last frame contains an error. The EIFS duration is defined as follows : Transmission time of Ack frame at lowest physical rate (or bit rate) + SIFS + DIFS.
- SIFS : Short Interframe Space. Amount of time in microseconds required for a wireless interface to process a received frame and to respond with a response frame. More precisely, it is the time interval between the last bit of the transmitted frame and the first bit of the corresponding response frame.
- RIFS : Reduced Interframe Space.

2 | Layers

The reference model used is a simplified version of the OSI reference model (which contains 7 layers). This simplified model is divided into 5 layers :

- Physical
- Datalink (LAN)
- Network (WAN)
- Transport (TCP/UDP)
- Application (Web/mail)

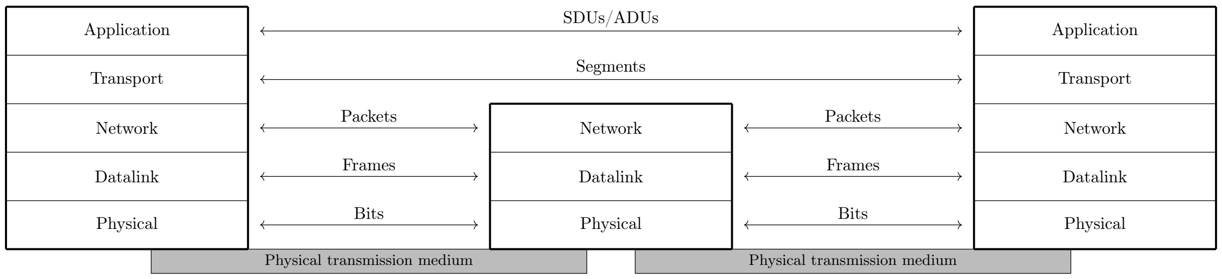


Figure 2.1: The five layers reference model

2.1 Physical Layer

Here, we will see it only as a black bow which changes the value of a bit (because of electromagnetic interference) and which delivers more or fewer bits than requested (because of an imprecise clock frequency).

2.2 Datalink Layer

The Datalink layer allows two hosts that are directly connected through the physical layer to exchange information using frames (finite sequence of bits with a particular syntax or structure) on MAC addresses.

When the Datalink layer entity needs to transmit a frame, it issues as many Data.request primitives to the underlying physical layer as there are bits in the frame. The physical layer will then convert the sequence of bits that will be sent over the physical medium. The receiver's physical layer will decode the received signal, recover the bits and issue the corresponding Data.indication primitives to its Datalink layer entity. If there are no transmission errors, this entity will receive the frame sent earlier.

A frame can be separated into 3 parts :

1. Header : contains a flags to indicate its type (ACK, DATA) a sequence number and the length of the payload if there is one.
2. Payload : contains the information that needs to be transmitted.
3. Error-detecting code : allow the receiver to detect transmission errors. It is either a Hamming code, a Checksum, a Cyclic Redundancy Check (CRC), or a hash function.

When transmitting a frame between switches, the packet stays the same. But when reaching a router, it will replace the addresses contained in the frame by its addresses.

2.3 Network Layer

The network layer allows to exchange information between hosts that are not attached to the same physical medium using packets.

A packet is a finite sequence of bytes that is transported by the Datalink layer inside one or more frames. A packet usually contains information about its origin and its destination, and usually passes through devices on its way from its origin to its destination.

2.4 Transport Layer

Different communication flows can take place between the same hosts, they might have different needs and need to be distinguished. The transport layer ensures an identification of a communication flow between two given hosts using segments.

A segment is a finite sequence of bytes that are transported inside one or more packets. They are issued by the transport layer as Data.request to the underlying network layer entity.

They are different types of transport layers, the most widely used are :

- TCP (Transmission Control Protocol) : provides a reliable connection-oriented bytestream transport service.
- UDP (User Datagram Protocol) : provides an unreliable connection-less transport service.

2.5 Application

The application layer includes all the mechanisms and data structure that are necessary for the applications.

Two application layers entities exchange data using generic SDU (Service Data Unit) or ADU (Application Data Unit) terms.

3 | Protocols

3.1 Datalink Layer

3.1.1 Spanning Tree Protocol (STP)

Allows for reducing the network to a Spanning Tree.

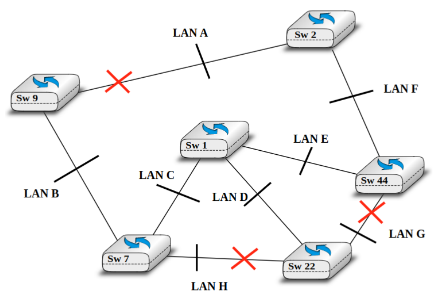


Figure 3.1: Spanning Tree example

Switches running the Spanning Tree Protocol exchange BPDUs (Bridge Protocol Data Unit) that are always sent as frames. Each switch has a unique 64 bit identifier. The switches exchange BPDUs to build the spanning tree.
Intuitively, the spanning tree is built by first selecting the switch with the smallest identifier as the root of the tree. The branches of the spanning tree are then composed of the shortest paths that allow all of the switches that compose the network to be reached.

The BPDU := ⟨ R, c, T, p ⟩ contains :

- The identifier of the root switch (R)
- The cost of the shortest path between the switch that sent the BPDU and the root (c)
- The identifier of the switch that sent the BPDU (T)
- The number of the switch port over which the BPDU was sent (P)

The construction of the spanning tree depends on an ordering relationship among the BPDUs. This ordering relationship could be implemented by the following Python function :

```python
# returns True if bpdu b1 is better than bpdu b2
def better(b1, b2):
    return ((b1.R < b2.R) or
            ((b1.R == b2.R) and (b1.c < b2.c)) or
            ((b1.R == b2.R) and (b1.c == b2.c) and (b1.T < b2.T)) or
            ((b1.R == b2.R) and (b1.c == b2.c) and (b1.T == b2.T) and (b1.p < b2.p)))
```

A switch port can be in three different states : Root, Designated, and Blocked. All the ports of the root switch are in the Designated state.

The state of the ports on the other switches are determined as follow :

1. Each switch listens to BPDUs on its ports.
2. For every received BPDUs, the priority vector $V[q] = \langle R, c + cost(q), T, p, q \rangle$ (where $cost(q)$ is the cost associated to the port q over which the BPDU was received) is computed and the best for each port is stored.
3. The root switch is found by looking at the smallest identifier in the priority vector table :
   - The root switch at $\langle R, 0, R, p \rangle$ (where R is the switch identifier and p will be set to the port number over which the BPDU is sent).
   - The other switches at $\langle R, c', S, p \rangle$ (where R is the root identifier, c' the cost of the best root priority vector, S the identifier of the switch and p will be replaced by the number of the port over which the BPDU will be sent) know their Root port (the one with the best priority vector).
4. A port is Designated if the switch's BPDU is better than the port's priority vector, otherwise it is Blocked.

3.2 Network Layer

3.2.1 Internet Protocol Version 6 (IPv6)

Soon, IPv4 won't be able to accommodate for all addresses anymore, which is why another solution is needed. IPv6 addresses are encoded on 128 bits ($3.4 * 10^{38}$ different IPv6 addresses), their format header can easily be parsed by hardware devices, they and can be configured automatically, and they provide more security.

Textual representation of IPv6 addresses

The preferred format for writing IPv6 addresses is x:x:x:x:x:x:x:x, where the x's are hexadecimal digits representing the eight 16-bit parts of the address. Example : abcd:ef01:2345:6789:abcd:ef01:2345:6789.

:: is used to indicate one or more groups of 16 bits blocks containing only bits set to 0. Example : ff01:0:0:0:0:0:0:101 is represented as ff01::101.

An IPv6 prefix can be represented as *address*/*length*, where *length* is the length of the prefix in bits. Example : 2001:0db8:0000:cd30:0000:0000:0000:0000/60 == 2001:0db8:0:cd30::/60.

IPv6 addressing architecture

The scalability of a network layer depends on its addressing architecture. IPv6 supports unicast, multicast and anycast architectures.

Unicast

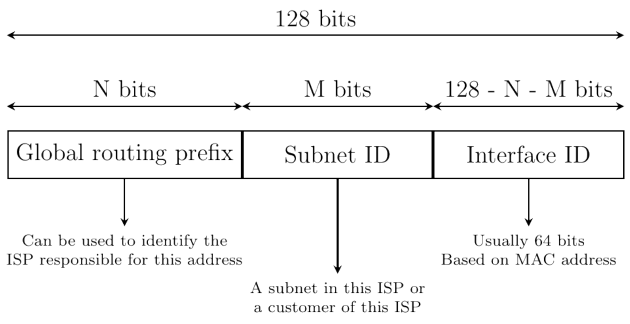

128 bits
N bits
M bits
128 - N - M bits
Global routing prefix
Subnet ID
Interface ID
Can be used to identify the ISP responsible for this address
A subnet in this ISP or a customer of this ISP
Usually 64 bits Based on MAC address

Figure 3.2: Structure of IPv6 unicast addresses

An IPv6 unicast address is composed of three parts :

1. A global routing prefix that is assigned to the Internet Service Provider that owns this block of addresses.
2. A subnet identifier that identifies a customer of the ISP.
3. An interface identifier that identifies a particular interface on a host.

To preserve the scalability of the routing system, it is important to minimize the number of routes that are stored on each router. Hierarchical address allocation allows for minimizing the amount of routes known to the router. It therefore only knows the route for certain address blocks (2^{128} grouped in 2^{64} subnets). Two types of address allocation:

- Provider-Independent (PI): For companies connected to at least two ISPs. Address blocks are given out independently of the ISP.
- Provider-Aggregatable (PA): Depends on the ISP.

The drawback with PA addresses is that when your provider changes, all the addresses you use in a PA address block also change. The typical size of the IPv6 address blocks are :

- /32 for an Internet Service Provider
- /48 for a single company
- /56 for small user sites
- /64 for a single user (e.g. a home user connected via ADSL)
- /128 in the rare case when it is known that no more than one host will be attached

The usage of the IPv6 prefix is the following : the longest prefix match assures the route that has the best match with the address is the one that is used (::/0 matches with all and is therefore the default route).

The Unique Local Unicast (ULA) address is an fc00::/7 and is similar to an IPv4 address. The address isn't necessarily unique and a router doesn't forward a ULA. As opposed to local unicast links, it doesn't necessarily need to have the same link.

Link Local Unicast : the address starts with fe80::/64, followed by 64 interface bits. Used when two hosts on the same link (or LAN) want to exchange a packet. The router can't forward a packet with a local unicast link. (Used when regular IPv6 isn't an option, that is, an isolated LAN.)

Multicast

Allows for sending a packet to all people in a same LAN group.

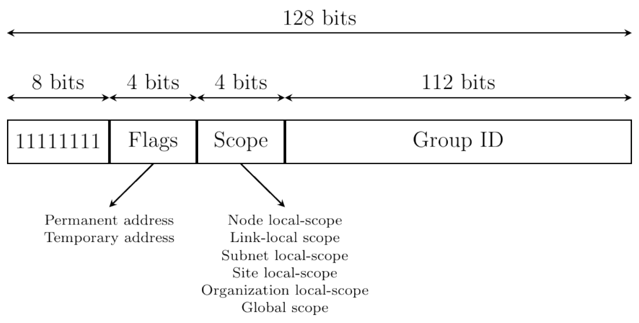
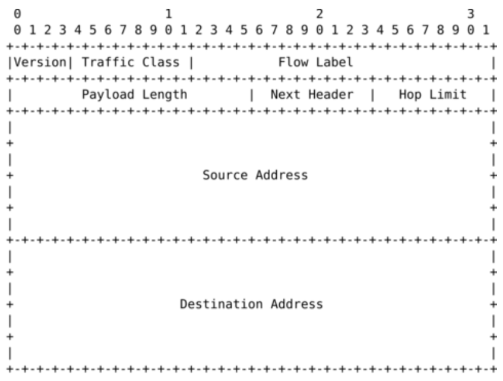


Figure 3.3: IPv6 multicast address structure

IPv6 packet format



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version| Traffic Class |           Flow Label                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Payload Length        |  Next Header  |   Hop Limit   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                         Source Address                        +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                      Destination Address                      +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3.4: IPv6 packet format

The Header field (40bits) is composed of Payload length (16bits, maximum $\approx 64kB$), Hop limit[1] (8bits) and Next header (8bits) + Version (4bits) + SRC (128bits) + DST (128bits) + Traffic (8bits ECE/ECT).

The Next Header field identifies the encapsulated data (if UDP, TCP, etc.)

3.2.2 Internet Control Message Protocol version 6 (ICMPv6)

Internet Control Message Protocol version 6 is used in IPv6 packets to transmit messages to the sender (used mainly for debugging purposes). There are two types of messages:

- Error messages : Destination unreachable, Packet too big, Time exceeded, Parameter problem.
- Information messages : when a host receives an Echo request, it has to respond with an Echo reply.

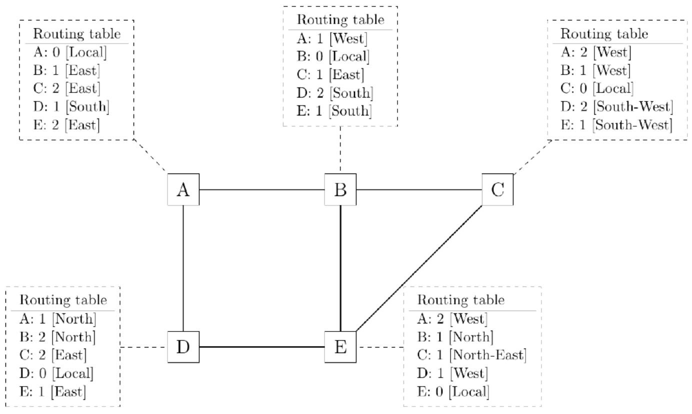
In order to do so, we will a 4 fields to the IPv6 packets : Type, Code, Checksum, Message Body.

[1]When retransmitting a packet, the hop limit is decremented. When it reaches 0, the packet is dropped and a error is send to the source.

3.2.3 Intradomain routing

Routing Information Protocol (RIP)

The Routing Information Protocol (RIP) is the simplest routing protocol that was standardized for the TCP/IP protocol suite. RIP routers periodically exchange RIP messages sent inside a UDP segment.



Routing table
A: 0 [Local]
B: 1 [East]
C: 2 [East]
D: 1 [South]
E: 2 [East]

Routing table
A: 1 [West]
B: 0 [Local]
C: 1 [East]
D: 2 [South]
E: 1 [South]

Routing table
A: 2 [West]
B: 1 [West]
C: 0 [Local]
D: 2 [South-West]
E: 1 [South-West]

Routing table
A: 1 [North]
B: 2 [North]
C: 2 [East]
D: 0 [Local]
E: 1 [East]

Routing table
A: 2 [West]
B: 1 [North]
C: 1 [North-East]
D: 1 [West]
E: 0 [Local]

Figure 3.5: Routing tables in RIP

When a router boots, its routing table is empty and it cannot forward any packet. To speedup the discovery of the network, it can send a request message to the RIP IPv6 multicast address. All RIP routers listen to this multicast address and any router attached to the subnet will reply by sending its own routing table as a sequence of RIP messages. These messages contain the distance vectors that summarize the router's routing table.

Open Shortest Path First (OSPF)

With link-state routing, it's very costly for big networks to store all of the network in memory. To resolve this issue, we introduce hierarchical routing that divides the network into regions. In OSPF, a region is called an area. OSPF uses the following terminology :

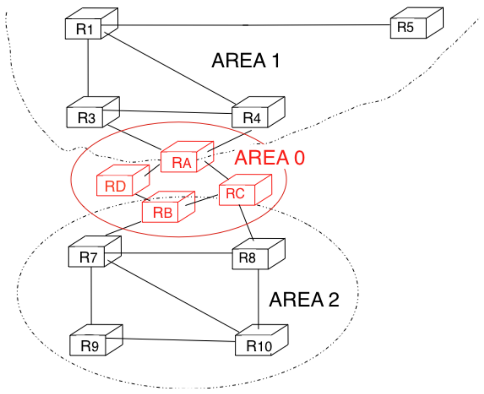


Figure 3.6: OSPF areas

- Area : set of routers and links grouped together. Routers know the topology of their area and how to reach the backbone area, they exchange link-state packets with everyone in the area.
- Backbone Area : area that groups the border routers and those that aren't in an area (AREA 0).
- Area border router : router attached to several areas (RA, RB, RC).
- Internal router : router whose directly connected networks belong to the area.

Inter-area

Inter-area routing is done by switching distance vector protocols to prevent routers from having additional costs to reach routers from a different domain.

LAN

When routers boot in a LAN, they elect a Designated Router (DR) so as to not have to exchange HELLO packets between all the routers. The routers can only exchange HELLOs with the DR; it represents the LAN.

3.2.4 Interdomain routing

Border Gateway Protocol (BGP)

In the Border Gateway Protocol, when en BGP router advertises a route towards a prefix, it announces the IP prefix and the interdomain path used to reach this prefix. As in BGP, every domain is identified by a unique Autonomous System (AS) number, the interdomain path contains the AS numbers of the transit domains that are used to reach the associated prefix.

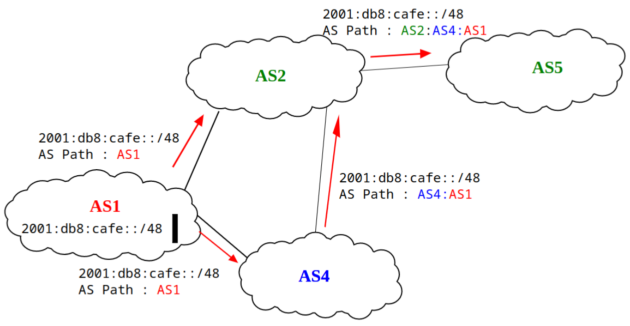


Figure 3.7: Exchange of BGP routes

Example with the above BGP route : Thanks to the AS-Path, $AS5$ knows that if it sends a packet towards *2001:db8:cafe::/48*, the packet first passes through $AS2$, then through $AS4$ before reaching its destination inside $AS1$.

There are two types of relations between domains :

- customer→provider : the customer pays for his domain to be distributed on the Internet (in return, the provider shares all all routes it knows.)
- Shared cost : this happens when domains have similar sizes. Domain only advertises its internal routes/prefixes and the routes it learned from its customers (not other shared-cost routes)

3.3 Transport Layer

3.3.1 User Datagram Protocol (UDP)

The User Datagram Protocol is used when delay must be minimized or losses can be recovered by the application itself (or real-time application like interactive videos). It is an unreliable connectionless transport service, running on top of an unreliable connectionless network service, which uses ports to allow communicating with multiple applications (errors are detected using the checksum).

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Length             |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3.8: UDP header format

3.3.2 Transmission Control Protocol (TCP)

The Transmission control protocol is a bidirectional bytestream of a connection-oriented transport service running on top of an unreliable connectionless network service.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Acknowledgment Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Data |       |C|E|U|A|P|R|S|F|                               |
| Offset| Res.  |W|C|R|C|S|S|Y|I|            Window             |
|       |       |R|E|G|K|H|T|N|N|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |         Urgent Pointer        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3.9: TCP header format

Connection establishment

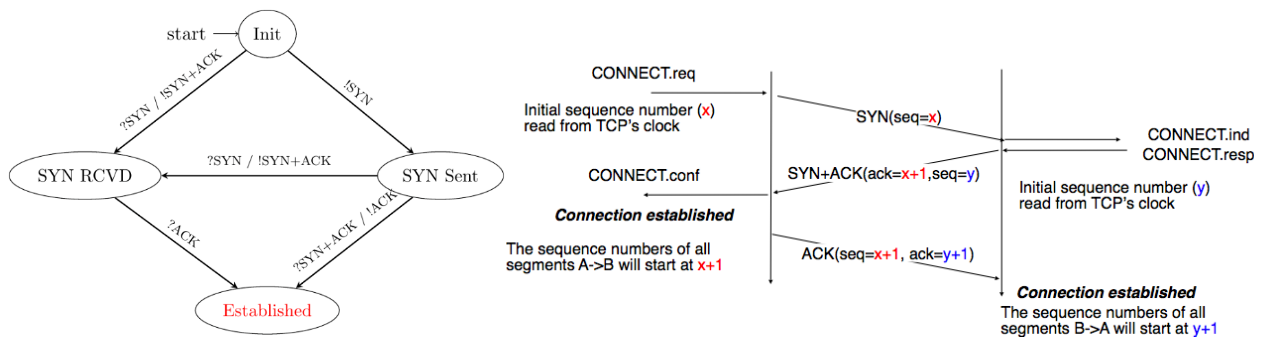


Figure 3.10: Three-way handshake

The Three-way handshake uses a sequence number, an acknowledgement number, and SYN/ACK/RST flags. The different messages types in TCP are :

- Synchronize (SYN) : Used to initiate and establish a connection. It also helps you to synchronize sequence numbers between devices
- Acknowledgement (ACK) : Helps to confirm to the other side that it has received the SYN
- Synchronize+Acknowledgement (SYN+ACK) : SYN message from local device and ACK of the earlier packet
- Reset (RST) : Used to signal a refused connection
- FIN : Used to signal a connection release (end of transmission)

Congestion control

Network congestion describes the reduced quality of service that occurs when a network node is carrying more data that it can handle.

TCP controls congestion by acting on the window size since a connection can't send data faster than $\frac{window}{rtt}$ where *window* is the maximum between the host's sending window and the window advertised by the receiver, and *rtt* is the round-trip-time (delay between the transmission of a segment and the reception of an ack).

In general, the maximum throughput that can be achieved by a TCP connection depends on its maximum window size and the rtt if there are no losses. If there are losses, it depends on the MSS, the rtt and the loss ratio. This gives the following relation :

$$Throughput < min(\frac{window}{rtt}, \frac{k * MSS}{rtt * \sqrt{p}})$$

Where :

- k = a constant
- MSS = Maximum Segment Size (most hosts use the same $MSS = 1460B$)
- p = loss ratio

As we can see, TCP connections with smaller *rtt* and/or a larger *MSS* can achieve a higher throughput than other TCP connections (unfairness).

The TCP congestion control scheme can be implemented using the following algorithm (also called *AIMD congestion control*) :

```
# Initialization
cwnd = MSS  # congestion window in bytes
ssthresh= swin # in bytes

# Ack arrival
if tcp.ack > snd.una:  # new ack, no congestion
    if dupacks == 0:  # not currently recovering from loss
        if cwnd < ssthresh:
            # slow-start : quickly increase cwnd
            # double cwnd every rtt
            cwnd = cwnd + MSS
        else:
            # congestion avoidance : slowly increase cwnd
            # increase cwnd by one mss every rtt
            cwnd = cwnd + MSS * (MSS / cwnd)
    else:  # recovering from loss
        cwnd = ssthresh  # deflate cwnd RFC5681
        dupacks = 0
else:  # duplicate or old ack
    if tcp.ack == snd.una:  # duplicate acknowledgment
        dupacks += 1
        if dupacks == 1 or dupacks == 2:
            send_next_unacked_segment  # RFC3042
        if dupacks == 3:
            retransmitsegment(snd.una)
            ssthresh = max(cwnd/2, 2*MSS)
            cwnd = ssthresh
        if dupacks > 3:  # RFC5681
            cwnd = cwnd + MSS  # inflate cwnd
    else:
        # ack for old segment, ignored
        pass

Expiration of the retransmission timer:
    send(snd.una)  # retransmit first lost segment
    sshtresh = max(cwnd/2, 2*MSS)
    cwnd = MSS
```

3.4 Application Layer

3.4.1 DNS

The Domain Name System (DNS) protocol runs above the datagram and bytestream service. The header of DNS messages is divided into 5 parts. The first tree are mandatory (the other two are optional) and are :

- ID : each request is given an ID to verify that it is the correct DNS request (example : the request foo will be given the ID_{foo} and the request bar will be given the ID_{bar} and will not conflict when the response is send)
- Flag Recursive : Used to know if the DNS server will resolve the entire IP at once (the resolver recuses through the DNS hierarchy to retrieve its answer)
- Flag Authority : Used to known wherever the response comes from a server cache or not

The command *dig* can give retrieve the IP of a server (3 types of record : A (IPv4), $AAAA$ (IPv6), and MX (mail server). Example : If you want to retrieve the IP of a mail server :

```
dig -t MX student.uclouvain.be +short
# student-uclouvain-be.mail.protected.outlook.com
COM=$(dig +short -t NS com | head -n 1)
OUT=$(dig +short @$COM -t NS outlook.com | head -n 1)
PRO=$(dig +short @$OUT -t NS protected.outlook.com | head -n 1)
MAI=$(dig +short @$PRO -t NS mail.protected.outlook.com | head -n 1)
dig +short @$PRO student-uclouvain-be.mail.protected.outlook.com | head -n 1
```

3.4.2 Mails

An email system is composed of a :

- A message format
- Protocols exchange between host and server
- A client software to create/read mails
- A software to allow the server to efficiently exchange mails.

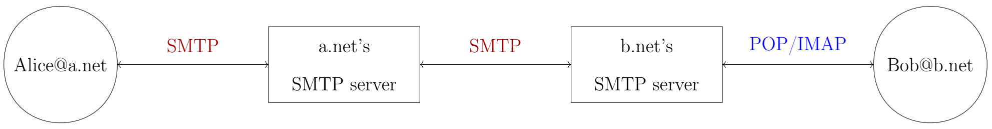


Figure 3.11: Simplified mail architecture

Here, Alice sends a mail using $SMTP$ to her mail server. Bob will retrieve its mail either using POP (removes it from the server) or using $IMAP$ (create a copy). These 3 protocols uses TCP.

Simple Mail Transfer Protocol (SMTP)

SMTP allows client to send emails and does not require authentication to be used. Each SMTP message send must contain: the (mail) address $from$, the (mail) address to and a date.

Post Office Protocol (POP) and Internet Message Access Protocol (IMAP)

POP and IMAP allows client to retrieve their emails from a server. Both POP and IMAP require authentication to be used.

1. Establish a connection with the mail server
2. Authenticate the user
3. Downloads the mails from the server

Multipurpose Internet Mail Extensions (MIME)

MIME was designed to allow email to carry non-ASCII characters and binary files without breaking the email servers that were deployed at that time. In order to do so, new header fields were added to support MIME (MIME-version, Content-type, and Content-Transfer-Encoding).

3.4.3 HyperText Transfer Protocol (HTTP)

To replace the File Transfer Protocol (FTP), the *World Wide Web* was created using hupertext to link files together. The three components of the World Wild Web are :

1. Uniform Resource Identifier (URI) that uniquely identifies a resource on the WWW. It is divided into three parts : '*protocol://user@server:port/resource*'
   (a) A protocol for the application layer (http://, mailto:, ftp://)
   (b) Authority which is the IP address or DNS server where the file is located
   (c) A path to the file in UNIX format
2. HTML : A standard document format (divided into two parts : header and body) containing hypertext link
3. HTTP : A standard protocol with efficient access to document on the server, made of request and responses. It contains :
   (a) A method (GET, HEAD, POST), a URI and a version of the HTTP
   (b) A header (specifies optional parameters)
   (c) An optional MIME document

Comparisons

HTTP evolved during its different versions :

	HTTP 1.0	HTTP 1.1
TCP Connection	One per data	Persistent support (keep alive)
Caching	If-Modified-Since	Based on an entity tag
Host Header	No	Yes

Table 3.1: TCP in HTTP 1.0 VS HTTP 1.1

	HTTP 1.1	HTTP 2.0
Encoding	ASCII	Binary
Optimization method	Pipelining	Multiplexing
Compression support	No	Yes
Push support	No	Yes

Table 3.2: TCP in HTTP 1.1 VS HTTP 2.0

3.4.4 Secure Shell (SSH)

The Secure Shell runs directly above the TCP protocol and allows for remote logins. Each user possess a public and a private key. When a message is send via SSH, it is send using the public key of the receiver that will be able to decrypt it using its private key.

Contrary to TLS, SSH does not require any certificate. As the algorithm takes into account the fact that the client's cache contains the public key of the server. This means that, in the case of a MITM (Man-In-The-Middle) attack, the protocol will see that the keys do not match and warn the user (which will decide to continue or not).

For a connection, there is simply an exchange where the client indicate its SSH version and where the server answers with its. Then, the client will communicate the encryption algorithm it knows and the server will answer with the one it will use.

3.4.5 Transport Layer Security (TLS)

The TLS family of protocols works basically the same way as SSH but instead of using private keys, it will use certificates. This is due to the fact that, when navigation over a large number or pages, the client cannot store every public key it encounters. Instead of exchanging keys, the client/server will exchange digital certificates.

4 | Tips

4.1 Window size

4.1.1 Maximum window size with go-back-n and selective repeat

A reliable protocol that uses n bits to encode its sequence number can send up to 2^n successive frames. however, to ensure a reliable delivery of the frames, *go-back-n* and *selective repeat* cannot use a sending window of 2^n frames.

For *go-back-n*, if a sender sends 2^n frames they are received in-sequence by the destination but all the returned ack are lost. This will cause the sender to retransmit all frames, which will all be accepted by the receiver and delivered a second time to the user. This can be avoided if the maximum size of the sending window is $2^n - 1$ frames.

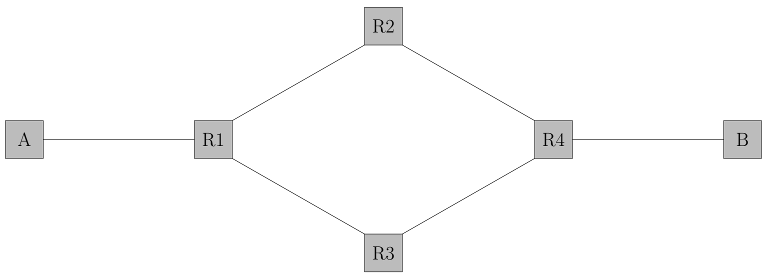
For *selective repeat*, a similar problem occurs with the difference that the receiver accepts out-of-sequence frames. This means that a sending window of $2^n - 1$ is not sufficient to ensure a reliable delivery. To avoid this problem, a selective repeat sender cannot use a window that is larger than $\frac{2^n}{2}$ frames.

4.2 HTTP

4.2.1 Role of Header's fields and what if they are deleted ?

- Host : Specifies the URI address and the server port receiving the request. Even if the port is optional, deleting this port would result in errors (to be precise : 400 Bad Request)
- Referrer : Indicate the site where the visitor came from. Deleting this could help in reducing tracing but will not avoid it as the visitor could still be traced by his login, his navigator and his cookies.
- If-Modified-Since : This allows for a conditional request : we only download a resource if it has changed since the last time we visited it. Deleting this field would result in performance losses as the resources would be reloaded even if there is no need to.

4.3 IPv6 Routing



On the above network, each router has only one route by default (A to $R1$, $R1$ to $R2$, $R2$ to $R4$, $R3$ to $R4$, and $R4$ to $R2$).

We can see that there is a cycle between $R2$ and $R4$, and B cannot send any packet. If we were to keep only one default route, it would be impossible for the traceroute to work as $R4$ can only send to $R2$, $R3$ or B while it has to be capable of sending to B when receiving from $R2$ and to $R3$ when from B. To fix this, we need $R3$ to route to $R1$ and $R1$ to have two routes : $R1$ to A when receiving from $R3$, and $R1$ to $R2$ when receiving from A.

4.4 Link State

In a Link State protocol, when :

- A link between two routers is broken, both routers will see that they no longer receive anything from this link. They will update their rooting table and send a new LSP without this link to indicate to the other routers that it no longer exist.
- A router is out of order, all its links will be broken. The other routers will only remove the LSP they received from the broken router when it expires.