Communication Networks

S. Ryffel

14th October 2006

Contents

1 Direct Link Networks

1.1 OSI-Model

7	Application Layer	
6	Presentation Layer	
5	Session Layer	
4	Transport Layer	process-to-process channel, transparent transfer of data, reliability, multiplexing
3	Network Layer	routing, segmentation,
2	Data Link Layer	connectivity among locally attached network nodes, error control, MAC
1	Physical Layer	transport of bits, electrical and physical specifications for devices

1.2 TCP/IP

4	Application	DNS, TFTP, TLS/SSL, FTP, HTTP, POP3, SIP, SMTP, SSH, TELNET
		Routing protocols like BGP and RIP, which for a variety of reasons run over TCP and UDP respectively, may also be considered part of the application or network layer.
3	Transport	TCP, UDP, DCCP . . .
		Routing protocols like OSPF, which run over IP, may also be considered part of the transport or network layer. ICMP and IGMP run over IP may be considered part of the network layer.
2	Network	IP (IPv4, IPv6)
		ARP and RARP operate underneath IP but above the link layer so they belong somewhere in between.
1	Link	Ethernet, Wi-Fi, Token ring . . .

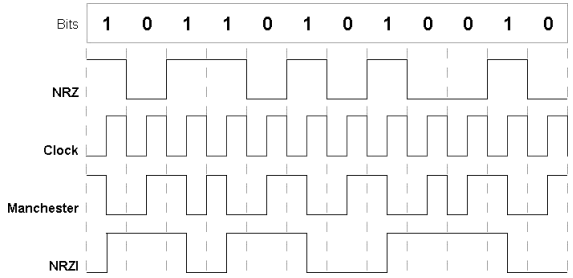


Figure 1: Encodings

1.3 Encoding

See figure 1.

NRZ non-return-to-zero

NRZI non-return-to-zero-inverted, 1: keep level, 0: change level

Manchester

4B/5B mapping groups of 4 bits onto groups of 5 bits in order to provide as many transitions as possible, transmitted using NRZI

1.4 Framing

Byte-oriented Protocols

- BISYNC-protocol [Peterson and Davie, 2004, p. 80]
- PPP [Peterson and Davie, 2004, p. 81]

Bit-oriented Protocols

- HDLC [Peterson and Davie, 2004, p. 83]
  - Frame delimiter: `01111110`
  - *Receiver* decision rule

```
011111 ...
 0:  Bit stuffed, remove it
 1 ...
  0:  Frame delimiter
  1:  Error, wait for the real delimiter
```

Clock-based Protocols

- SONET [Peterson and Davie, 2004, p. 84]

1.5 Error Detection

Parity Block

Every bit stream is divided into lines containing 7 data bits and a parity bit. 7 lines form a block where the ith bit of the 7th line is the parity of all ith bits of the other lines.

Internet Checksum

[Peterson and Davie, 2004, p. 90]

Treat message as stream of 16bit numbers and add them. The resulting 16bit number is the checksum.

Cyclic Redundancy Check (CRC)

[Peterson and Davie, 2004, p. 92]

1.6 Reliable Transmission

- Stop-and-Wait: [Peterson and Davie, 2004, p. 97]
- Sliding-Window: [Peterson and Davie, 2004, p. 100]

2 Shared Medium Access

Given a channel with an average of G generated frames per time slot, then the probability of the arrival of exactly k frames in a time slot is given by the Poisson distribution

$$P[k|D] = \frac{G^k e^{-G}}{k!}$$

2.1 Multiple Access

Aloha

Aloha sends, when it has something to say. (see fig 2)

Normalized Throughput: given N Stations sending with probability p then $G = Np$

- normal Aloha:

$$S = G \cdot \left(1 - \frac{2G}{N}\right)^{N-1} \overset{N\to\infty}{=} G \cdot e^{-2G}$$

- slotted Aloha:

$$S = G \cdot \left(1 - \frac{G}{N}\right)^{N-1} \overset{N\to\infty}{=} G \cdot e^{-G}$$

Carrier Sense Multiple Access

Carrier Sense Listen (before) sending.
Does not eliminate collisions eg because of the "vulnerable period" or "shadowing" generically unstable

Vulnerable Period Maximal propagation time of a signal in a network. When a station starts to send, an other can start too in this time and cause a collision in spite of carrier sense.

Shadowing When in a wireless network two stations are out of reach of each other but send to one in the middle and therefore cause a collision.

1-persistent Station sends as soon as the channel becomes idle.

non-persistent When the network is busy, the station waits a random amount of time. best strategy

p-persistent When the channel becomes free, the station sends with probability p. (only when slotted)

Packet Size Versus Bus Length: for all collisions to be detected

$$T_{tr} > 2T_{pr}$$
$$\frac{S}{C} > 2\frac{L}{v}$$

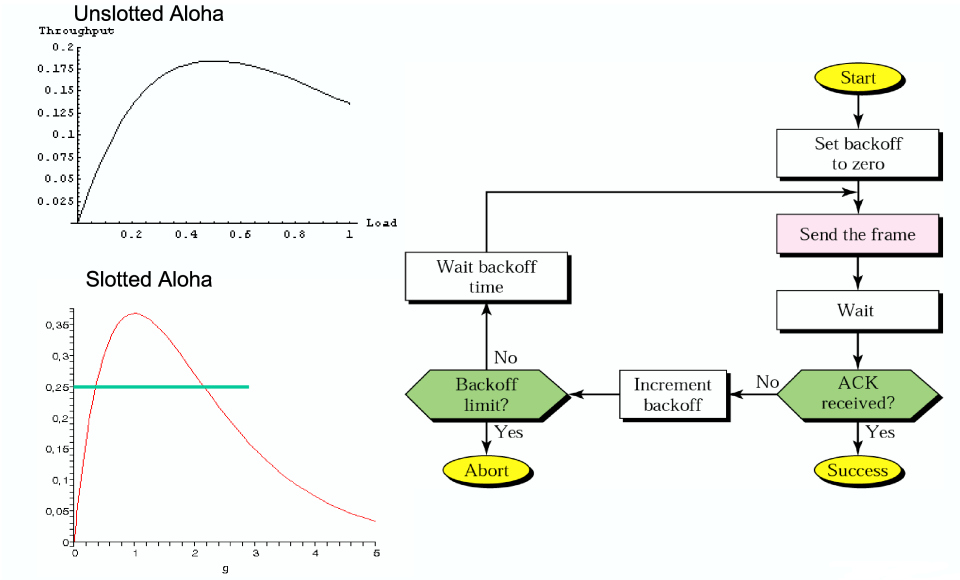


Figure 2: Aloha Protocol

T_{tr} Transmission Time of a frame.
T_{pr} Propagation Time of a message.
And S is the frame size [bit], C the link capacity [bit/s], L the bus length [m] and v the signal propagation speed [m/s].

CSMA with Collision Detection (CSMA/CD)

Stations can detect collisions while sending (duplex) and abort transmission and so reduce the cost. (see fig 3) No need for p-persistence since the cost of collisions are low.

Jam Signal so other stations detect collision as well.

Backoff Wait $2^N\times$ max-propagation-time after collision, where N is number of transmission attempts.

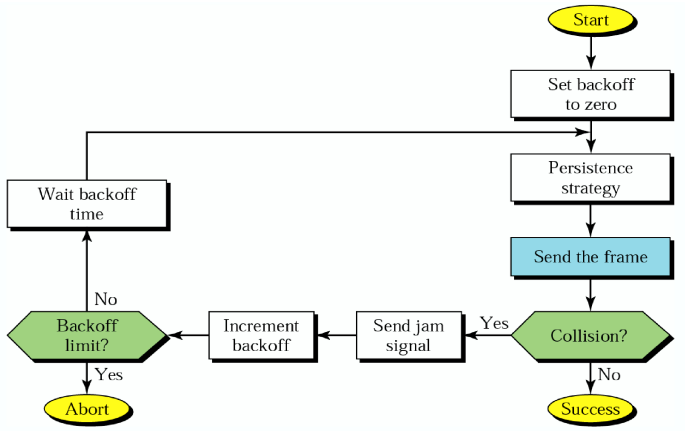


Figure 3: CSMA/CD

CSMA/CA (MACA)

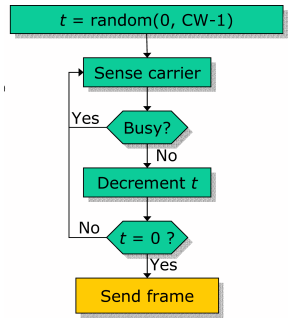


Figure 4: CSMA/CA Wait Procedure

Collision detection does not work well on wireless networks

- *Hidden Node Problem, "shadowing":* A and B can each communicate with C, but are hidden from each other
  A *hidden node* is out of reach of others.
- *Exposed Node Problem:* Occurs when a node is unnecessarily prevented from sending packets to other nodes due to a neighboring transmitter, even though no interference would actually occur.
  Example: Consider an example of 4 nodes labeled R1, S1, S2, and R2, where the two receivers are out of range from one another, yet the two transmitters in the middle are in range of each other and one of the receivers. Here, if a transmission between S1 and R1 is taking place, node S2 is prevented from transmitting to R2 as it concludes that it will interfere with the transmission by its neighbor S1.
- Would require full duplex radio interfaces. Signal strength difference big because of heavy damping.

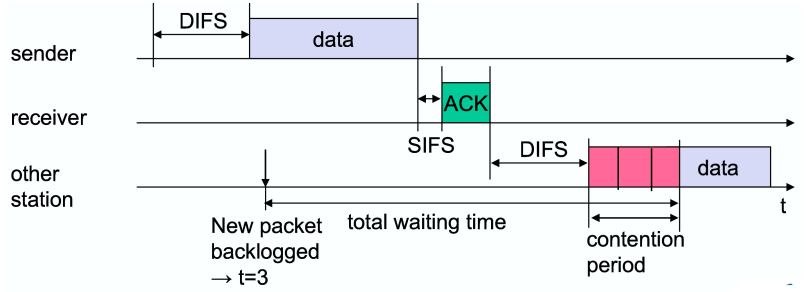


Figure 5: Distributed Coordination Function

Distributed Coordination Function from 802.11 (see fig 5)

DIFS DCF Interframe Space ($50\mu s$), time a station has to wait so send data.

SIFS Short IFS ($10\mu s$), gives ACKs a higher priority

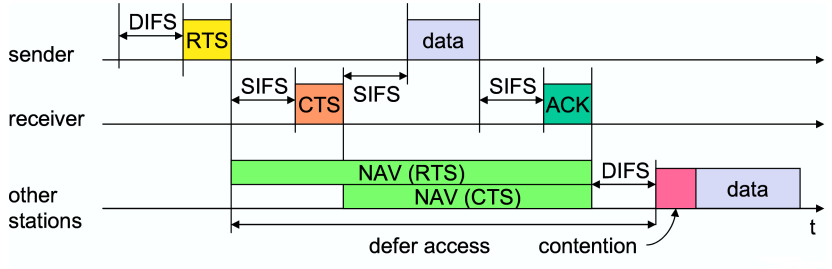


Figure 6: CSMA/CD with RTS/CTS

CSMA/CA with RTS/CTS (MACA) See figure 6.

A node wishing to send data initiates the process by sending a Request to Send frame (RTS). The destination node replies with a Clear To Send frame (CTS). Any other node receiving the CTS frame should refrain from sending data for a given time. This solves the hidden and the exposed node problem.

RTS Request To Send

CTS Clear To Send, Acknowledgement by receiver, sender sends after SIFS

NAV Net Allocation Vector to keep track of the reservations

Carrier Sense with non-persistent transmissions

- Non-backlogged (non-waiting) nodes send after sensing the medium.
- Packets arriving when medium is busy get backlogged and are not transmitted immediately when medium gets idle.
  Contention Window Maximal time backlogged nodes wait before sending packets (see figure 4).
  - Is doubled after each collision.
  - Measured in slots ($20\mu s$).
  - Only decrement timer, if medium is idle ie wait for n free slots.
- Backlogged nodes transmit after sensing the medium idle for a random time.

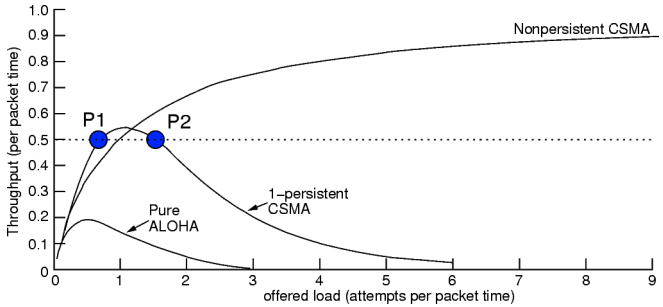


Figure 7: Performance Comparison

Multiple Access with Collision Avoidance for Wireless (MACAW)

MACAW works like MACA but receiving stations send an ACK when the frame arrived successfully. Other stations have to wait with their frames until they hear the ACK.

2.2 Controlled Access

Token Passing

A token, which gives the right to send, circulates among the nodes. This is used in Token Ring LAN.

Polling: the Point Coordination Function

- Polling mode for IEEE 802.11 wireless LAN is implemented by few only.
- Could provide bit-rate guarantees.
- APs send “beacon” frames at regular intervals
  Between beacon frames, PCF defines:
  - Contention free period (CFP) when polling is used.
    AP polls stations
  - Contention period (CP) when DCF is used.

2.3 Local Area Networks (LANs)

IEEE Project 802

See figure 8.

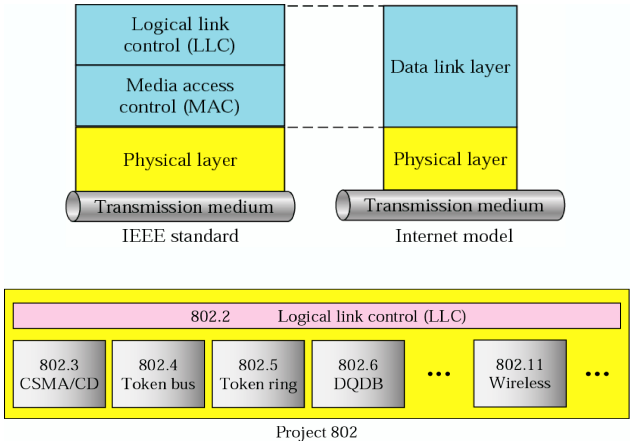


Figure 8: IEEE Project 802

Ethernet (IEEE 802.3)

[Peterson and Davie, 2004, p. 110]

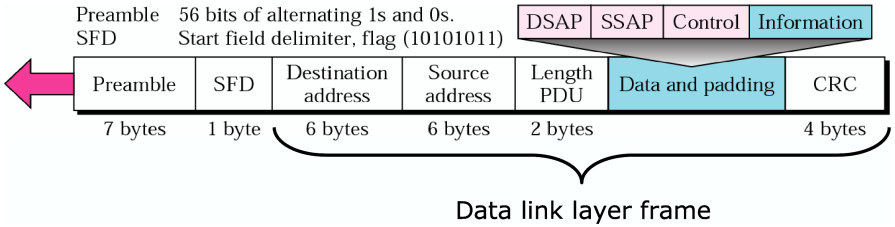


Figure 9: Ethernet (IEEE 802.3) MAC Frame Format

Ethernet (IEEE 802.3) MAC Frame Format see figure 9.

- Overhead: 18bit at link layer, 26bit on the wire
- Addresses: 48bit, multicast (8th bit is 1), unicast, broadcast (ff:ff:ff:ff:ff:ff)
- CRC 32

Sending Algorithm

- CSMA/CA, 2.5km
- minimal packet size is 512bit in order to detect collisions

- random exponential backoff: $[0 \dots 2^n - 1] \cdot 51.2\mu s$

Ethernet Type	Media Type	Speed	Distance	Coding
10Base-T	UTP cat 3 (or better)	10 Mb/s	100 m	Manchester
10Base-FL	Multimode fiber	10 Mb/s	2000 m	Manchester
100Base-T	Cat 5 UTP or STP	100 Mb/s	100 m	4B/5B + MLT-3
100Base-FX	Multimode fiber	100 Mb/s	2000 m	4B/5B + NRZ-I

Figure 10: Ethernet Types

Ethernet Types See figure 10.

Wireless LAN (IEEE 802.11)

[Peterson and Davie, 2004, p. 130]

Service Modes

1. *Infrastructure Mode, Basic Service Set (BSS)*: with AP
2. *Ad Hoc Mode*: without AP
3. *Extended Service Set*: several BSS interconnected by a Distribution System (eg wired LAN), fast link-level handover

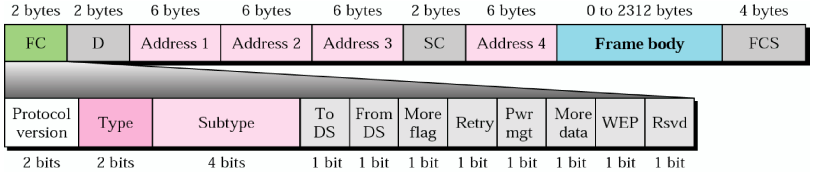


Figure 11: Wireless LAN (IEEE 802.11) Frame Format

Wireless LAN (IEEE 802.11) Frame Format See figure 11.

Wireless LAN (IEEE 802.11) MAC address format See figure 12.

IEEE 802.11 Wireless LAN Standards See figure 13.

scenario	to DS	from DS	address 1 (phys. dest.)	address 2 (phys. source)	address 3 (log. Addr.)	address 4 (log. Addr.)
ad-hoc network	0	0	DA (phys. dest.)	SA (phys. source)	BSSID	-
infrastructure network, from AP	0	1	DA (log & phys. dest.)	BSSID (phys. source.)	SA (log. source)	-
infrastructure network, to AP	1	0	BSSID (phys. dest.)	SA (log & phys. source)	DA (log. dest)	-
infrastructure network, within DS	1	1	RA (phys. dest.)	TA (phys. source)	DA (log. dest)	SA (log. source)

DS: Distribution System
AP: Access Point
DA: Destination Address
SA: Source Address
BSSID: Basic Service Set Identifier
RA: Receiver Address
TA: Transmitter Address

Figure 12: Wireless LAN (IEEE 802.11) MAC address Format

Standard	Spectrum	Physical rate	Data rate	Compatible
802.11	2.4 GHz	2 Mb/s	1.2 Mb/s	-
802.11a	5.0 GHz	54 Mb/s	32 Mb/s	-
802.11b	2.4 GHz	11 Mb/s	6-7 Mb/s	802.11
802.11g	2.4 GHz	54 Mb/s	32 Mb/s	802.11b 802.11

Figure 13: IEEE 802.11 Wireless LAN Standards

Wireless LAN (IEEE 802.11) Security

- WEP (Wired Equivalent Protocol), considered too weak
- IEEE 802.11i with stronger encryption, robuster authentication
- WPA (Wi-Fi Protected Access) by Wi-Fi Alliance

Beacon synchronization See figure 14.

- Beacon frames are sent using DIFS+random backoff interval
- Beacon frames contain timestamp, BSSID, management information

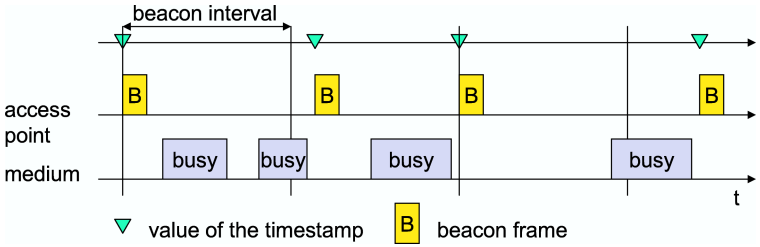


Figure 14: Beacon synchronization interval $100ms$

3 Switching and Internetworking

3.1 Extending LANs

Layer	*Switch*
3. Networking	Router
2. Data Link	Bridge
1. Physical	Repeater or hub

Repeater

Connects two network segments.
- MAC protocol must be identical
- a hub is a multi-port repeater

Bridges

Function
- Connects two or more LANs
  - with different data-link layer protocols
  - bandwidth increases
  - probability of collisions decreases
- Forwards complete, correct frames
  - only to segments where destination is
    forwarding table: table mapping MAC-addresses to ports
    when port unknown, broadcast frame.
  - buffer frames for busy ports

Learning Bridge
- Save address and port for each arriving frame
- *Loop Problem*: When multiple Bridges connect station A to B, then each forwards the frame which results in a multiplication of the frame.

Spanning Tree
- *Purpose*: Bridges dynamically create a tree of the typology.
- Each Bridge has an unique 48b ID
- Configuration Messages
  - Format: Root ID - Cost - ID of transmitting bridge
  - Sent to special multicast address that assigned to “all bridges” of a LAN.
- *Process:* by Radia Perlman
  1. The node with the smallest ID is the *root bridge*.
     - It advertises itself as the root bridge by flooding configuration messages.
     - It stops advertising itself if it receives a configuration message of a bridge with a lower id.
  2. Each Bridge marks the port with the least cost path to the root as a *root port*.
  3. On each LAN-segment mark a *designated port* of the *designated bridge*.
     - Bridge with least cost path to root bridge (or the smaller id).
     - Mark corresponding port as the designated port.
  4. Forward frames only on marked ports.
- *Failure Management*:
  - Changes in topology: Discard messages older than 20s and recompute.
    Bridges wait some time before recomputing and reconfiguring status of ports (30s).
    * Loss of connectivity, not so severe
    * Temporary loops, packets multiply, severe
  - Root node sends Configuration Messages every 2s.
    When node does not receive messages anymore, it advertises itself as the root bridge.

Virtual LANS (VLAN)

[Peterson and Davie, 2004, p. 190]

3.2 Switched Networks

Components
- Switching Nodes: buffer and forward data packets
- End Nodes: provide data, network is transparent to them
- Links: physical connections

Three “Hows” of Switched Networks
- *How . . .*
  - . . . to switch data?
    Asynchronous (Packets), Synchronous (Time slots)
  - . . . to select route?
    network computed (centralized, decentralized), host computed
  - . . . to prevent links from overload?
    congestion control: hop by hop or end to end

- *When . . .*
  . . . to decide route?
  connection-oriented: before first packet is sent
  connection-less: at any time

Switching

Switching

- *Circuit Switching*
  synchronous switching, eg telephone
- *Packet Switching* (see table 1)
  asynchronous switching
  - *Connection Oriented* (Virtual Circuit) eg TCP
  - *Datagram* eg UDP

Datagram	*Virtual Circuit*
Immediate transmission	Connection set-up delay (≥RTT)
No state information in switches about connections	Switches maintain state information about connections
Transmission order not preserved	Transmission order is preserved
Address lookup for forwarding of packets	Identifier lookup for forwarding packets
not vulnerable to failing switches	vulnerable to failing switch in VC
Resources not reserved, non-deterministic delay	Resources may be reserved, no congestion

Table 1: Datagram Versus Virtual Circuit

Datagram - Connectionless Switching

Every Switch looks at the destination-address and its routing-table and decides where to route the packet to.

Virtual Circuit - Connection-oriented Switching

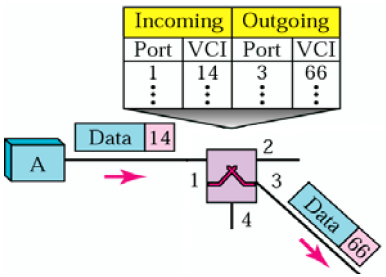
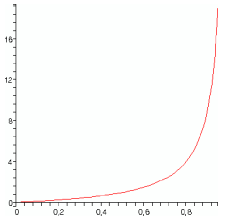


Figure 15: Virtual Circuit Switching

The packet contains a Virtual Circuit Identifier (VCI) which tells the switch where to route the packet to. Every Switch has a VC table containing: input interface, input VCI, output interface and output VCI.

Properties of Packet Switching



$$\sim \frac{r}{1-r}$$

Figure 16: Waiting Time over Load

- statistical multiplexing, allowing changing bitrate
- congestion: figure 16

Packet Switch (Router) Functions

Forwarding plane

- receive and transmit data-link frames
  layer 1 and 2 protocols
- verify packet header data before using them
- update header data
  swap VCI, update TTL, re-compute checksum
- determine output port and forward packet
- buffer packets when outgoing link is busy

Control plane

- Compute routing to be used in forwarding
- Network management
- Uses network for control traffic

Packet Switch Buffer Designs

Buffer the Output Controllers (OCs)

- 100% throughput when buffer are infinite
- Performance-demanding and expensive: switch fabric may have to forward N packets to each of N outputs per time.

Buffer the Input Controllers (ICs)

- suffers from head of the line blocking with FIFO:
  packet that is waiting for a busy port blocks whole line asymptotically 59% of outgoing link capacity
- 100% throughput with non-FIFO possible: Tradeoff between complexity of scheduling and performance

Congestion in Packet Switching Networks

Different strategies for discarding packets

- Drop tail: last arrival is lost
- Random drop
- Random early detection: packets are discarded, before buffer is entirely full
- active queue management

Congestion Control

- Long term congestion:
  Network dimensioning: that network has enough capacity
  Traffic engineering: change routing, lease extra capacity
- Short term congestion, Congestion Control
  Sources reduce rate and then probe for increasing it again.

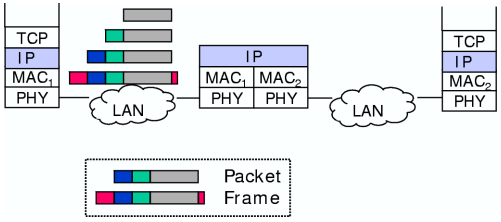
3.3 Wide-Area Networking

Internetworking

- Interconnection of networks
- Different network technologies: LAN, P2P, WiFi
- Different address formats

Network Layer

see figure 17.



Packet on layer 3
Frame on layers 1,2

Figure 17: Network Layer

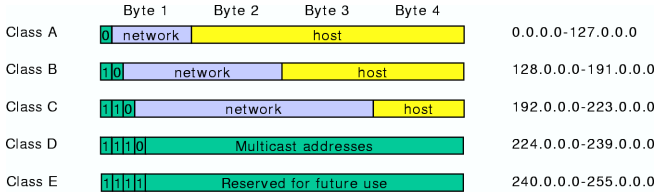


Figure 18: Original IP Address Classes

IP Protocol Stack

IP Addressing

- 32Bit in V4: 4.3 billion addresses
- Two purposes:
  1. Identify a host
  2. Give the location of the host
- Address has two parts:
  Network prefix: Identifies the network, routes outside the hosts network only look at this network address.
  Host suffix: Identifies the node in that network.
- Special addresses
  Unicast address: Identifies a host
  Network address: First address of a subnetwork, all host suffix bits are zero
  Multicast address: Last address of a subnetwork, all host suffix bits are one
  Packet is bradcasted in the subnet.
  Broadcast address `255.255.255.255`: Broadcast to the local subnet
- *IP address classes*: see figure 18.
  Address space depletion, in spite of only 5% of the address space beeing used!

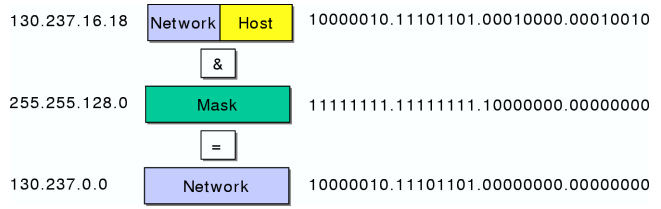


Figure 19: Address Mask

Classless Inter-Domain Routing (CIDR)

- IP network address represented as a prefix: `192.16.30.0/20`
  Where `/20` denotes the amount of bits of the network prefix.

IP forwarding of packets

- When the network suffix matches with one of the ports, send it there.

- Routers have routing (forwarding) tables:
  Forwarding table links the network prefix with the next hop.
  When the network is not in the table, send the packet to the default gateway.

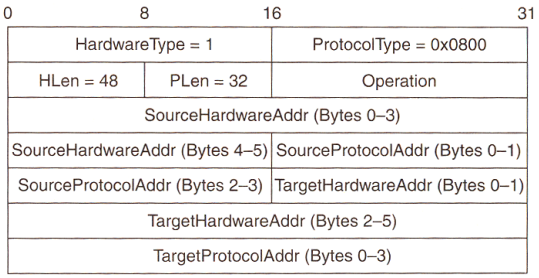


Figure 20: ARP-Frame

Address Resolution Protocol (ARP)

- ARP helps to find the MAC of a host with a given IP.
  Mappings of communication-partners are stored and removed after 15min.
- ARP is used between one IP hop.
- Protocol: (over data link layer)
  1. Sender broadcasts an ARP-Request on the network.
  2. The host replies with his MAC. See figure 20.
     HardwareType type of physical network, eg Ethernet
     ProtocolType eg IP
     HLen / PLen hardware/protocol address length
     Operation request (1), reply (2)
  3. When the sender does not get a reply, it sends the package to the gateway

IP Header See figure 21.
VER Version = 4
HLEN Header length in words (32bit, 4Byte), 20-60bytes
DS Service type, QoS
Total length Length of header + data in bytes, max 65536bytes
Identification, flags, offset for fragmentation
Flags: Reserved, must be zero; Don't Fragment (DF); More Fragments (MF)
Time to live max numbers of routers to pass
Protocol 6:TCP, 17:UDP
Header checksum 16bit checksum of all fields in header

Fragmentation [Peterson and Davie, 2004, p. 239]

- Any router my fragment a package when it is longer than the sub-network can carry. The fragments are IP packets themselves and sent independently.
- Fragments get reassembled by the receiver.
- Hosts choose Maximum Transmission Unit (MTU) of their network.
  $\texttt{data} + \texttt{ip-header} \leq \texttt{MTU}$

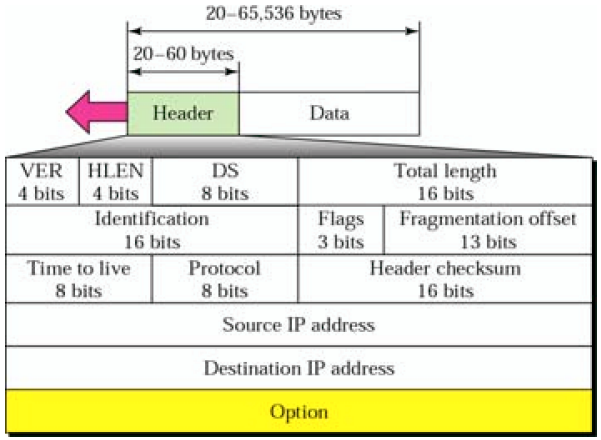


Figure 21: IP-Header

- IP Header:
  - Flags: OR 001, set fragmented flag.
  - Fragmentation Offset: Set offset in bytes of frame in package.

Avoid Fragmentation with Path MTU Discovery
Finds the smallest MTU on the path of a packet.

- *Procedure:* Send the largest packet you can, and if it won't fit through some link get back a notification saying what size will fit. The notifications arrive as ICMP (Internet Control Message Protocol) packets known as "fragmentation needed" ICMPs (ICMP type 3, subtype 4). The notifications are requested by setting the "do not fragment" (DF) bit in packets that are sent out.
- *Problem:* Some network and system administrators block all ICMPs.

MTUs of different networks See table 2.

Medium	*MTU in bytes*
Token Ring (4Mbps)(802.5)	4464
Token Ring (16Mbps)	17914
FDDI	4352
Ethernet	1500
PPPoE (z. B. DSL)	$\geq$ 1492
SLIP/PPP (low delay)	296
X.25	576
ISDN	1500
ATM	4500

Table 2: MTUs of different networks

Internet Control Message Protocol (ICMP)
- Indicated by protocol “ICMP” in IP header
- Error messages:
  - Destination unreachable: No route, fragmentation needed, port does not exist
  - Time exceeded: TTL. Packet reassembly
  - Parameter problem: bad header
  - Redirect: use other router
- Query messages:
  - Echo request/reply
  - Time stamp, address mask, router solicitation

Dynamic Host Configuration Protocol (DHCP) [Peterson and Davie, 2004, p. 261]
- Automatic configuration of hosts by DHCP server
- *Protocol* (over UDP, client port: 68, server port: 67)
  1. DHCP Discover: Local host sends a broadcast packet with a request for its last IP address.
  2. DHCP Offer: The DHCP server sends a broadcast packet with an available IP address.
  3. DHCP Request: Local host sends a broadcast packet to request the provided IP address.
  4. DHCP ACK: The DHCP server acknowledges the request with a broadcast packet.
  5. The local host sends a broadcast ARP packet to check whether its IP address is already in use.
- *Parameters*
  - IP address / IP mask
  - Default router address
  - Addresses of DNS-Server
  - Link MTU, TTL ...

4 Routing

Routing vs. Forwarding

Routing A distributed process which builds the routing information in the routers.

Forwarding The process in the Router or Switch which forwards packets using the information given in the local routing table.

4.1 Distance Vector Routing (RIP)

Algorithm

1. Each router sends a list of all known destinations with a metric (distance vector) to its neighbors.
2. Each router updates its internal tables according to the information received.
3. The distance vector is sent periodically and when the routing table was adapted.
4. Distance values are set do infinite after a timeout.
5. K of n rule: A change is only accepted if it persists during at least k of n periods. In order to avoid routing oscillations.

Approaches to Solve the Count-to-infinity Problem

Problem Broken links are not recognized because some stations think they can reach other stations not knowing that their route leads over the broken link.

Hold-Down Interval
- If your route D goes down, wait for some time dt before using another path (hold-down time)
- During dt, advertise cost to D as infinite.
- Issues:
  - dt has to be long enough, for all router to learn about broken link.
  - Convergence is delayed, which still is better than no convergence.

Report the Entire Path
- Solves the problem at significant costs.
- Used in BGP4 and partially in RIP2.

Split Horizon
- If route do D is learned from N, it is not advertised back to N.
- Does not work in some cases, especially if the network that tries to reach D over N contains a loop.

Poisson Reverse
- Destinations known to be unreachable are explicitly reported as such to all routers (instead of not mentioning them).
- Used in conjunction with Split Horizon: If route do D is learned from N, it is advertised with infinite costs back to N.

Routing Information Protocol (RIP)

- RIP is a DV-Protocol.
- Periodicity of routing updates: 30s, route timeout: 180s
- 16 is infinite, therefore the maximum network diameter is 15 hops
- UDP port 520
  Works because RIP is talking to neighbors only.

Header Format

Operation req or resp

Address family net x eg IP, Apple-Talk, ...

Operation	Version (1 / 2)	Null
Address family net 1		Null
IP address net 1		
Subnet mask net 1		
Next hop net 1		
Distance to net 1		
Address family net 2		Null
IP address net 2		
Subnet mask net 2		
Next hop net 2		
Distance to net 2		

Figure 22: RIP Header Format

4.2 Flooding

[Peterson and Davie, 2004, p. 280]

Goal To distribute a packet in the whole network.

- Each node should receive the packet at least once.
- Not too many duplicates.
- The destination addresses are not known.
- Packet-content: source, sequence number, ttl, body

Algorithm

```
On message from neighbor:  look up record in database (db)
IF TTL==0:
   discard message
ELSE IF record not present:
   add it, copy message to outgoing links (“replicate”)
ELSE IF seq# in db is lower  then seq# of message:
   replace record, replicate
ELSE IF seq# in db is higher then seq# of message:
   transmit value in db to incoming interface
ELSE IF seq# are equal:
   do nothing
```

4.3 Link State Routing

Algorithm

- Each router broadcasts a list of his neighbors connected through direct links with associated metrics.
- With time, each router will have the full topology of the network.
- Each router computed the best route using Dijkstra’s Shortest Path First (SPF)

Dijkstra’s Shortest Path First (SPF)

1. Start at the source u with the construction of the tree.
2. Select the leaf v with the lowest cost and add its neighbors as leaves from v. Mark their respective costs of the path from u over v.
3. Iterate 2.

Packet Content

- ID of the router which generated the packet.
- List of direct neighbors with corresponding metric.
- Sequence number
- Time to live

Open Shortest Path First (OSPF) V2

LS age			Options	Type=1
Link state ID				
Advertising router				
LS sequence number				
LS checksum			Length	
0	Flags	0	Number of links	
Link ID				
Link data				
Link type		Num_TOS	Metric	
Optional TOS information				
More Links				

Figure 23: OSPF Header Format

- Detection of broken links with periodic hello messages
- Minimum frequency of link state advertisements is 1/30min
- sent as IP packet
- Header Format [Peterson and Davie, 2004, p. 287]
  LS age Time to life
  Type 1: cost of link; 2: advertise networks
  LS sequence number to detect duplicate LSA
  Metric costs

4.4 Comparison

Bandwidth consumed

Depends on topology and circumstances.

Computational Complexity

LS

- n destinations with avg. of k links: total of $n * k$ links
- Dijkstra: $O(n * k \log n)$

DV
- Scan of n row and k column matrix on every vector $\rightarrow O(n * k)$ per scan.
- Multiple iterations possible

Robustness

Both algorithms are sensitive to malfunctioning routers

Functionality

LS wins (eg different metrics).

Speed of convergence

LS converges faster:
- Absence of multiple iterations
- No count-to-infinity problem
- Routing information passed without costly computation

DV vs: LS

Distance Vector	*Link State*
• Router sends DV information from routing table to all neighbours • Simple to implement • Simple to configure • Bad convergence • Bad scaling properties (network diameter < 16 hops)	• Router broadcasts list of direct neighbors • Faster convergence • Generates less traffic • Fast reaction on topology changes • Bad scaling properties

4.5 Autonomous Systems / Routing Domains

See figure 24.

Motivation for hierarchical routing
- Scalability: DV and LS scale badly
- Administration: Routing Policies, Metrics, Trust

5 The Global Internet

5.1 Hierarchical Routing

See figure 25.

Routing Domain An aggregate of networks or subnetworks that use a common internal routing protocol and communicate to other routing domains via an Inter-domain routing protocol.

Autonomous System Synonym for Routing Domain.

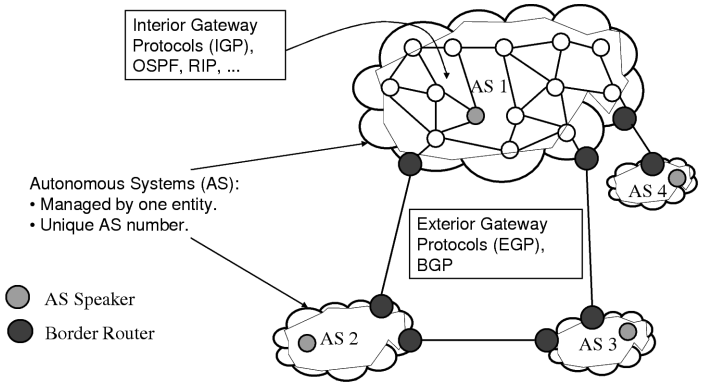


Figure 24: Autonomous Systems / Routing Domains

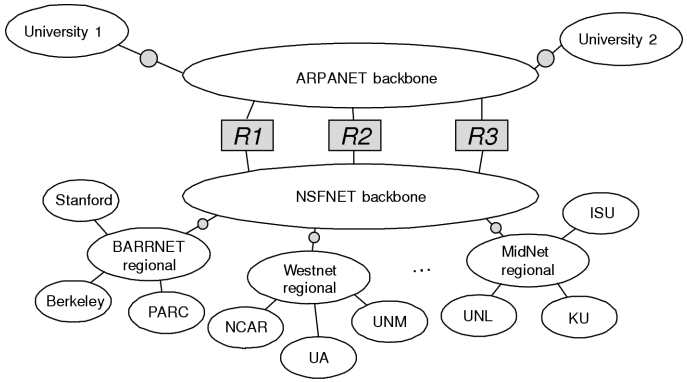


Figure 25: Beyond the core routing system

Gateway-to-Gateway Protocol (GGP)
- Early inter-domain routing protocol
  (or Exterior Gateway Protocol (EGP))
- Distance Vector protocol
- Addresses based on address classes A,G,C
- Metric: hops

The Internet: The Big Picture

See figure 26.

Autonomous Systems (AS)

Stub AS Single connection to other AS → local traffic only

Multihomed AS Multiple connections, no transit traffic

Transit AS Multiple connections, accepts transit traffic.

Unique AS number assigned that allows for exchanging routes as AS paths.

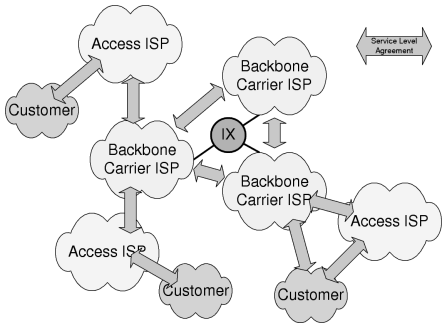


Figure 26: The Internet: The Big Picture

Border Gateway Protocol (BGP)

[Peterson and Davie, 2004, p. 308]

- Arbitrarily interconnected set of ASes
- Communicate reachability and route information among *AS speakers*.
- Information collected by *AS speakers* used to configure forwarding table of *border routers*.
  [Peterson and Davie, 2004, p. 310]
- BPG is a Path vector protocol: Exchange of routes as AS path vectors.
  → Path analysis
- No metric exchanged
- Extensive support for *routing policies* and manual configuration.

BGP Architecture See figure 27.

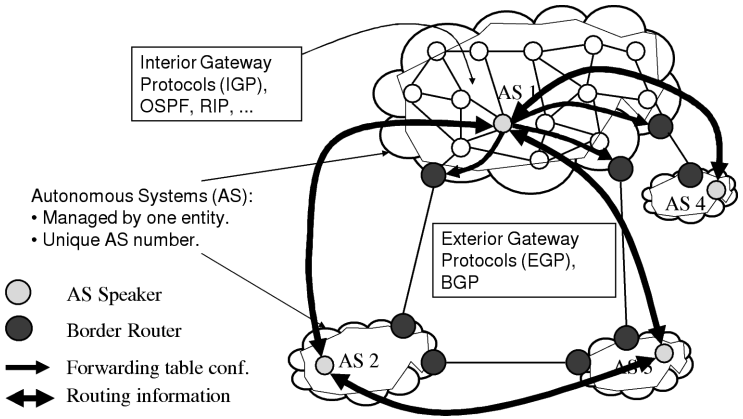


Figure 27: BGP Architecture

BGP Messages

open • To initialize communication, announce AS

update (full or incremental)

- Withdrawn routes: broken routes
- Path attributes: next hop, AS path
- For each path: list of destination networks (IP address with prefix)

notification response to an incorrect message

keepalive to maintain the connection (every 60 seconds by default)

- TCP port 179
- Support for authentication

5.2 Model of a Router

See figure 28.

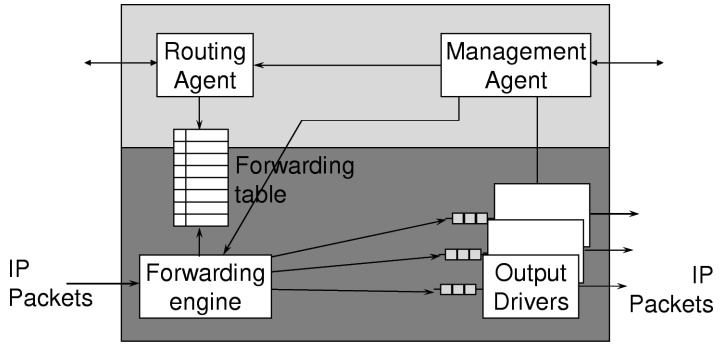


Figure 28: Model of a Router

Classless Inter-Domain Routing See figure 29 and figure 19.

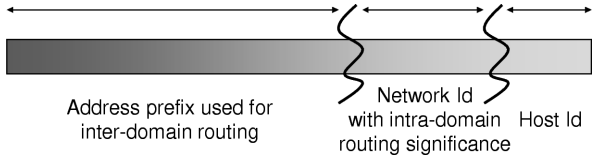
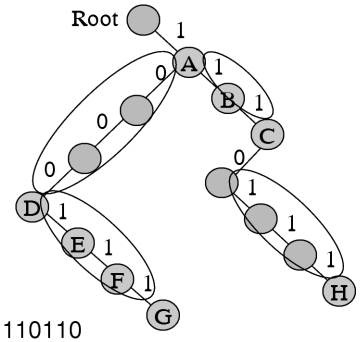


Figure 29: CIDR Address Structure

Forwarding Lookups

- Size of backbone router forwarding table is $> 180'000$ entries.
- Algorithmic solutions: Patricia tries, see figure 30
- Hardware solutions: Content Addressable Memories (CAM)
- Caching
- *Solution found in 1997:*

Forwarding table	
1*	A
11*	B
111*	C
1000*	D
10001*	E
100011*	F
1000111*	G
1110111*	H



Example: look up 100011110110
100101100011

Performance: $O(W)$ memory access, too slow, large memory requirements, caching expensive

Figure 30: Patricia Trie

- Novel application of a binary search strategy with hash tables
- 2-3 memory accesses
- Pre-computation of tables

6 Quality of Service

6.1 Quality of Service

Definition

Expectations on the network service
- Order of delivery
- Replication of messages
- Delay from transition of first to last bit
- Reliability: error, drops, unavailable
- Security: integrity of messages, privacy

Today and Tomorrow

Today: Best Effort
- Reliability handled end-to-end (TCP)
- Quality varies with the load of the network

Tomorrow: Congestion control for all traffic
- Batch transfers: reactive congestion
- Stream transfers: admission control grants bandwidth or blocks

Quality Aspects

Delay Propagation, processing, buffering

Information (packet) loss Full buffers, address errors, route changes, bit errors

Human Requirements

- Delay <150ms for conversations
- Video and audio <80ms apart
- Information loss depends on application
- Consistency and predictability

6.2 Service Architecture

- Set of traffic classes with quality guarantees
- Scheduling for separation of classes in network nodes
- Signaling
- Service-level agreement
- Network management: monitoring, allocation, planing

Traffic Control

General
- Quality degradation ad IP layer primarily due to overload
  - Limit the traffic:
    preventive or reactive congestion control
  - Prioritize traffic:
    congestion first hits the lop priority traffic, differentiated pricing

Preventive/Admission congestion control
- New connections need to be admitted
  - End system sends request
  - Network controller decides and verifies that the traffic meets the specified parameters
- Guarantees bounds for throughput, loss and delay
- Traffic description:
  - peak rate bound
  - token rate r and burst size b gives sending rate under a period τ: $r + b/\tau$

Reactive congestion control
- *Host starts sending*
  - Starts at low bitrate
  - Increases rate if no packets are lost
- Senders with connections through a *congested link* will reduce their rates.
  No guarantee of throughput and quality
- Unsuited for streaming sessions because of varying throughput

Scheduling

Weighted fair queuing
- A weight a_i is associated with each flow
- All arriving packets are time stamped
- Packets are served in order of increasing timestamps
- A Flow i is served at a rate proportional to $\frac{a_i}{\sum_j a_j}$

Service priorities
- High-priority load must be limited. Else it congests itself, low-priority might be useless
- Delay priority: High priority served first
- Discard priority: Low priority discarded first

Signaling

- Inform network and end-user:
  Traffic descriptor and quality expectations
- Requirements
  uni- and multicast connections, function with stateless routing and node failures, scalability
- Resource Reservation Protocol (RSVP)
  For setup of both uni- and multicast connections. Details on page 16.
- Stream Protocol, version 2 (ST-II)
  aka IPv5, obsolete

6.3 Standardized Service Architectures

Integrated Services (IntServ)

Traffic Classes

Guaranteed Service Deterministic multiplexing
- Parameter-based admission control
- Service guarantee: delay bound, no congestion
  delay bound: $\frac{(b-M)(p-R)}{R(p-r)} + \frac{M+C_{tot}}{R} + D_{tot}$
  R: Rate; M: maximum packet size; r, b, p: token-bucket parameters; C, D specified
- Per-flow scheduling in all routers
  (eg WFQ see section Scheduling on page 16)

Controlled-load Service Flow granted according to peak rate
- Statistical multiplexing
- Measurement-based admission control, overbooking
- Quality as an unloaded network
- Service equivalent to best effort under low load

Best Effort no guarantees

Resource reservation protocol (RSVP)
1. Sender issues PATH message with flowspec to destination
   Tspec: token-bucket descriptor of traffic flow
   Phop: address of least RSVP-router
   Sender identification
   Adspec: latency, capacity, pathMTU
2. Each router checks and saves reservation
3. Receiver issues RESV messages to reserve resources

Differentiated Services (DiffServ)

General Packets entering a network are classified
- Assigned to different *Behavior Aggregates*
- Each behavior aggregate is specified by a single DS code point (DS field of IP Header)
- Each node forwards a packet according to the per-hop behavior associated with the code point

Service level agreement
- Between user ant network provider
- Regulates a user's access to a service class
- Regulates the amount of traffic that may be sent within the different behavior aggregates
  Excess-traffic will be marked differently (out-of-profile)

Per-hop behaviors (PHB) A forwarding behavior
- PHB in a node is determined by code point in the packet header
- Examples:
  - Expedited Forwarding: forward with minimal delay and loss,
  - Assured Forwarding: drop first packets marked as out (out-of-profile), typically excess-traffic or different classes
  - Weighted RED: every forwarding-queue is assigned a weight;
    eg: premium is 1 (20%), best effort 4 (80%)

Premium (expedited) service Virtual leased line (peak-rate), only one destination

IntServ vs. DiffServ

IntServ
- RSVP complex, connection state in router
  → scales badly
- Service class definitions could be simpler

DiffServ
- Simple but too static (access control by SLA)
- Premium service useful
- Actual performance of services may be unclear

7 Transport Protocols

7.1 Transport Protocols

End-to-end Packet Transmission

Port	Name	Protocol	Description
20	ftp-data	TCP	File Transfer Protocol (data channel)
21	ftp	TCP	File Transfer Protocol (control channel)
23	telnet	TCP	Terminal
25	smtp	TCP	Mail transfer
53	dns	UDP/TCP	Domain name server
67	bootps	UDP/TCP	Bootstrap server
68	bootpc	UDP/TCP	Bootstrap client
69	tftp	UDP	Trivial File Transfer Protocol
80	www	UDP/TCP	WorldWideWeb HTTP
109	pop2	UDP/TCP	Post Office Protocol ver 2
110	pop3	UDP/TCP	Post Office Protocol ver 3
119	nntp	TCP	USENET News Transfer Protocol
123	ntp	UDP/TCP	Network Time Protocol

Figure 31: Some well known port numbers

Application addressing
- Server uses a “well known port number”, see figure 31
- Client port numbers are assigned dynamically, “Ephemeral” ports

TCP - Connection-oriented Transmission

Transmission Control Protocol (TCP)
- Connection-oriented:
  - Three-way handshake [Peterson and Davie, 2004, p. 383]
    1. `SYN; seq. no. x` (x random)
    2. `ACK x+1; SYN; seq. no. y` (y random)
    3. `ACK y+1; seq. no. x+1`
  - Connection termination
    1. `FIN; ACK y; seq. no. x`
    2. `FIN; ACK x+1; seq. no. y`
    3. `ACK y+1; seq. no. x+1`

    Both partners have to terminate the connection individually.
  - Unstructured byte stream
- Sliding window protocol
- Sender window determined by the minimum of receiver window size and congestion window size

TCP sender transmits segments
- “Smart sender, dumb receiver rule”
- Maximum Segment Size (MSS) set at connection setup
- Each segment has a sequence number

TCP receiver sorts segments in sequence number order
- Ignore duplicates
- Acknowledge received segments
  - Acknowledgements are accumulative, number identifies next expected byte
  - Expected within a given time (based on RTT)
  - ACKs are sent with delay (approx 200ms)
- Check checksum
- Send advertised window size in ACK

TCP Header Format see figure 32. [Peterson and Davie, 2004, p. 381]

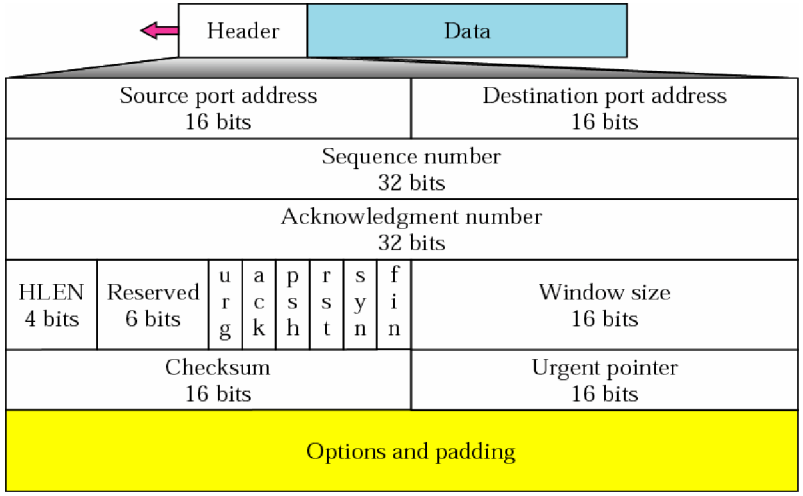


Figure 32: TCP Header Format

Identifier every TCP-connection is uniquely identified by
`<SrcPort, SrcIPAddr, DstPort, DstIPAddr>`
Sequence Number seq. number of the first byte in this segment
ACK Number seq. number of the next expected byte
Hlen size of the header in words (32bits)
Checksum same as in UDP (see section 7.1 on page 18)

TCP Protocol Functions

- *Reliability*
  - Retransmission of lost segments on
    - Retransmission Time Out (RTO), ACK timed out
    - Three ACKs for same byte number
  - Retransmission methods: stop-and-go and go-back-N ARQ
- *Flow Control "Sliding Window"*
  Do not overflow receiver [Peterson and Davie, 2004, p. 387]
  - Receiver announces an window size
    $AdvertisedWindow = MaxRcvBuffer - ((NextByteExpected - 1) - LastByteRead)$
  - Sender has to respect $AdvertisedWindow$
    $LastByteSent - LastByteAcked \leq AdvertisedWindow$
  - Self-clocking algorithm:
    Avoid the "Silly Window Syndrome" [Peterson and Davie, 2004, p. 393]

```
Send full segment when:
   available data ≥ MSS and
   advertised window ≥ MSS
Else
   wait for more data if there is unacknowledged
      data in flight
   otherwise send segment
```

- *Congestion Control* to not overload the network, see section 7.2.

Problems

- *Sequence number*: Two packets may not have the same sequence number.
  The sender must not be able to send 2^{32} bits inside the Maximum Segment Lifetime (MSL) = 120s.
- *Advertised Window*: The sender should be able to send at the full rate.
  $AdvertisedWindow \geq RTT * Bandwidth$

UDP - Datagram Transmission

User Datagram Protocol (UDP) Unreliable Datagram Service

- Application does flow and error control itself
- Connection-oriented too costly: overhead, setup
- Retransmission useless

UDP Header Format see figure 33.

Pseudo-Header three fields from the IP header: protocol number (17), Source and destination IP address, UDP length fields

Checksum over UDP-header, payload and pseudo-header

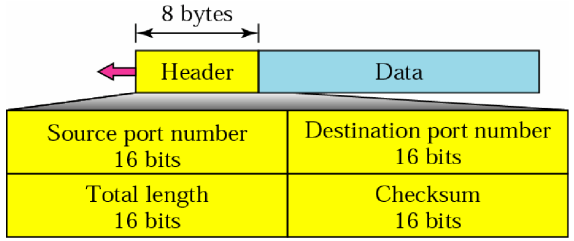


Figure 33: UDP Header Format

7.2 Congestion Control in TCP

[Peterson and Davie, 2004, p. 467]

Round-trip Time (RTT) Estimation

$$avRTT = a * avRTT + (1 - a) * sampleRTT$$
$$devRTT = devRTT + a * \left(\left|avRTT - sampleRTT\right| - devRTT\right)$$
$$\rightarrow RTO = b * avRTT + c * devRTT; (b = 1, c = 4)$$

Self-clocking

Congestion Control Loops

Inner loop based on self-clocking

- ACKs cannot arrive faster than the bottleneck capacity
- Obtains global stability fast (within four RTT)

Outer loop for setting the window size

- Ideally $w = RTT * C_{fair} + Q_{ref}$

Window Control

Additive Increase / Multiplicative Decrease (AIMD) (see figure 34).

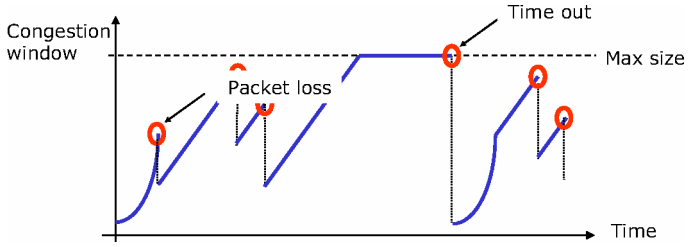


Figure 34: Congestion Window Control

Decrease
- Duplicate ACKs (Fast-Retransmit): reduce by half (FastRecovery)
  (data is getting through but with losses)
- RTO timeout: reduce to one segment
  (nothing is getting through)

Increase
- Slow Start, when network state is unknown
  1. One segment initial window size
  2. For each ACK increase the window size by one (exponential)
- When network state is known
  * Linear increase by one segment per RTT
  * After RTO, slow start until half of the previous window size (Connection-Threshold), then linear increase

Fast Retransmit [Peterson and Davie, 2004, p. 474]
Instead of waiting for the Timeout, retransmit a Packet if tree Duplicate-ACKs arrived for it. This does not replace Timeouts.

Fast Recovery [Peterson and Davie, 2004, p. 476]
After a Fast Retransmit, reduce Window Size by a half and continue with AIMD, instead of a slow start.

8 The Domain Name System (DNS)

[Peterson and Davie, 2004, p. 636]

8.1 Terminology

Name Identify objects, help locate objects, define membership, specify a role
Name Space Defines set and possibly structure of possible names
Directory Service Defines and implements name to value bindings

8.2 Goals

- Map user friendly names to router friendly addresses
- Hide address changes
- Allow different types of entries supporting different applications:
  user names, IP addresses, mailboxes, . . .
- Hierarchical distribution of the naming authority

8.3 The DNS Name Space

See figure 35.
- Arbitrarily nested tree:
  Each organisation is responsible for its sub-tree
- Naming Structure is a logical view:
  It may not correspond to the actual network topology or organisation

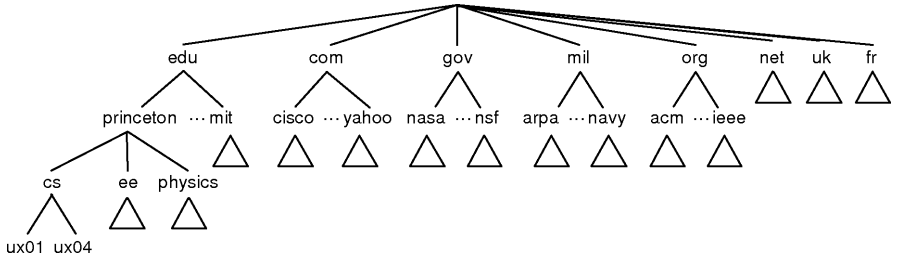


Figure 35: The DNS Name Space

- The nesting can be arbitrarily deep

Internet Domain Names

Top Level domain names
com Commercial organisations
edu Educational organisations
gov Governmental organisations
mil Military organisations
net Large network providers
org Other, mostly non-commercial organisations
arpa Used for reverse lookups
int International organisations
xx 2-Letter country code, according to ISO 3166
biz, name, museum, aero, eu, info, . . . new TLD

DNS Records
A Maps a hostname to a 32-bit IPv4 address
AAAA Maps a hostname to a 128-bit IPv6 address
CNAME Canonical name record makes one domain name an alias of another
MX Mail exchange record maps a domain name to a list of mail exchange servers for that domain
PTR Pointer record maps an IPv4 address to the canonical name for that host.
NS Name server record maps a domain name to a list of DNS servers authoritative for that domain.
SOA Start of authority record specifies the DNS server providing authoritative information about an Internet domain, the email of the domain administrator, the domain serial number, and several timers relating to refreshing the zone.
SRV, TXT, NAPTR Other types

Zones

Zone sub-tree of the name space which is managed as a unit
- May be sub-divided in sub-zones
- A DNS Server holds the data of one or more zones

8.4 Bind DNS Server

BIND (Berkeley Internet Name Domain, previously: Berkeley Internet Name Daemon) is the most commonly used DNS server on the Internet.

Configuration of Zones

Zone file for section6.net See table 3 for explanations.

```
$TTL 1d
section6.net. IN SOA syndie.section6.net. root.syndie.section6.net. (
                     1           ; Serial
                     10800       ; Refresh (for slaves)
                     3600        ; Retry on failed refresh
                     604800      ; Expire (no longer auth)
                     86400 )     ; Minimum TTL of 1 day
              IN NS    syndie.section6.net.
section6.net IN MX 10 syndie.section6.net.
@             IN A     10.0.0.1
syndie        IN A     10.0.0.1
vpn           IN A     10.0.0.2
ns            IN CNAME syndie
mail          IN CNAME syndie
```

Explanations	
`@ IN SOA <primary NS> <mail> (<timings)`	Start of authority, timings in seconds
`IN NS <nameserver>`	Configure nameserver
`<name> IN A <ip>`	A-Record
`<name> IN MX <order> <ip>`	Mailserver of order `<order>`
`<name> IN CNAME <dest-name>`	Canonical name of `<dest-name>`
`<ip> IN PTR <name>`	PTR-Record
`IN HINFO <cpu> <os>`	Specifies cpu and os
`TTL`	Time in seconds, the record may be cached

Table 3: Explanations to the zone file

Administration of the Name Space

Primary Name Server is responsible for one or more zones, loaded from a database
Secondary Name Server increase availability, loaded from the primary (zone transfer)

Root Servers bind the top level of the DNS together; each name server must know the addresses of the root servers:

HOSTNAME	NET ADDRESSES	SERVER PROGRAM
A.ROOT-SERVERS.NET	198.41.0.4	BIND (UNIX)
B.ROOT-SERVERS.NET	128.9.0.107	BIND (UNIX)
C.ROOT-SERVERS.NET	192.33.4.12	BIND (UNIX)
D.ROOT-SERVERS.NET	128.8.10.90	BIND (UNIX)
E.ROOT-SERVERS.NET	192.203.230.10	BIND (UNIX)
F.ROOT-SERVERS.NET	192.5.5.241	BIND (UNIX)
G.ROOT-SERVERS.NET	192.112.36.4	BIND (UNIX)
H.ROOT-SERVERS.NET	128.63.2.53	BIND (UNIX)
I.ROOT-SERVERS.NET	192.36.148.17	BIND (UNIX)
J.ROOT-SERVERS.NET	198.41.0.10	BIND (UNIX)
K.ROOT-SERVERS.NET	193.0.14.129	BIND (UNIX)
L.ROOT-SERVERS.NET	198.32.64.12	BIND (UNIX)
M.ROOT-SERVERS.NET	202.12.27.33	BIND (UNIX)

8.5 Domain Name Resolution

- The DNS resolver is configured with the address of any DNS server.
- Name resolution logically starts at the root of the name tree.
  Since most queries are more or less “local”, queries are rather processed “bottom up”.
- Queries (UDP, port 53)
  Iterative Mode Directed at one single server: Query → Redirection → Query → . . .
  Recursive Mode Directed at whole DNS, Server forwards query

Locality of Reference

- Many queries are for local entities, locality of reference may be used to increase performance
- DNS Servers maintain a cache of recently used names:
  authoritative Answers from a primary or secondary server
  non-authoritative Answers taken from a cache.

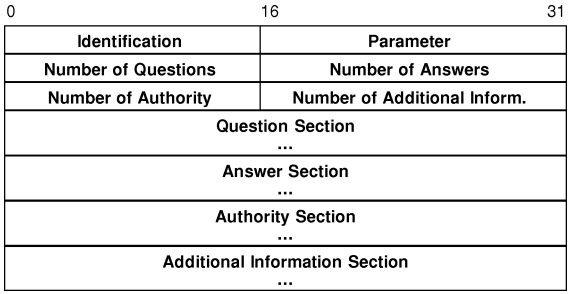


Figure 36: DNS Message

DNS Messages See figure 36.

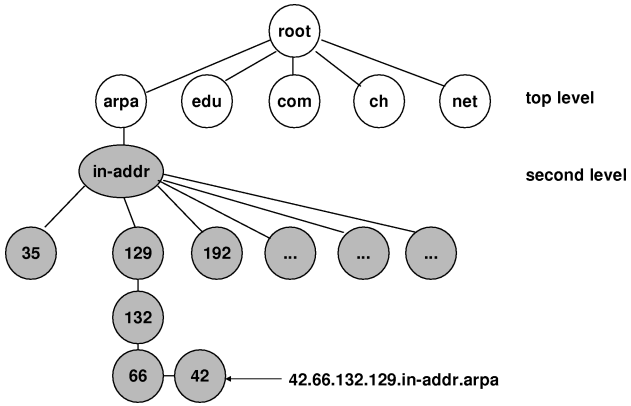


Figure 37: Name space for inverse queries

Inverse Queries "Pointer Queries"

- Inverse query: Given an IP address, provide the corresponding domain name
- Problem: A search for a specific IP address may have to be conducted on all name servers
- Solution: A special second level domain: `in-addr.arpa` contains a hierarchy. These mappings are not complete and contains many empty or old entries.

nslookup

```
sr@elmar ~ # nslookup
> set querytype=a
> www.ethz.ch
Server:         194.230.1.71
Address:        194.230.1.71#53

Non-authoritative answer:
www.ethz.ch     canonical name = www-css.ethz.ch.
Name:   www-css.ethz.ch
Address: 129.132.46.11

> www-css.ethz.ch
Server:         194.230.1.71
Address:        194.230.1.71#53

Non-authoritative answer:
Name:   www-css.ethz.ch
Address: 129.132.46.11

> set querytype=mx
> ethz.ch
Server:         194.230.1.71
Address:        194.230.1.71#53

Non-authoritative answer:
ethz.ch mail exchanger = 10 phil1.ethz.ch.
ethz.ch mail exchanger = 10 phil2.ethz.ch.

Authoritative answers can be found from:
```

9 Network Security

9.1 Security

The "CIA" triad

Security guarantees

Confidentiality Ensure that information is not disclosed to unauthorized subjects.

Integrity Ensure that information is not modified maliciously or accidentally.

Availability Ensure reliability and timely access to information and resources.

Cryptography in General

- *Principle by Kerkov / Kerkhoff*: The attacker knows the whole cryptographic system and all cryptographic algorithms used, with the exception of the keys.
- *Information-theoretically secure*: The attacker can't break the system by using *unlimited* computing resources.

- *Computationally secure*: The attacker can't break the system by using *limited* computing resources.

Communication Channel Model

	not conf.	conf.
not authentic	$\longrightarrow$	$\Longrightarrow$
authentic	$\bullet\!\!\longrightarrow$	$\bullet\!\!\Longrightarrow$

Table 4: Channel types

See table 4.

Not confidential channel An attacker can eavesdrop on all information sent.

Confidential channel No eavesdropping possible on information sent.

Not authentic channel The receiver has no guarantee that the sender is the one he claims to be, and that the content is original.

Authentic channel The receiver can be assured that the sender of the information is the one he claims to be and that the content is original.

9.2 Symmetric Cryptography

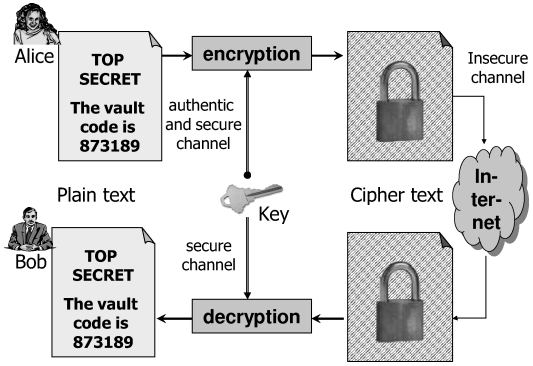


Figure 38: Symmetric Cryptography

See figure 38. Bob can be sure that the message was sent by Alice, if the plain text makes sense (redundancy) only.

One-Time Pad

Perfect security, if the key

1. has same length a s plain text
2. is randomly chosen and
3. kept secret.

Data Encryption Standard (DES)

History

- Developed by IMB in 1977
- DES is a block cipher for 64bit blocks and has 56bit keys

Today

- Security of DES is considered insufficient due to the short key length.
- Successors: Triple-DES or AES

DES En-/Decryption See figures 39 and 40. [Peterson and Davie, 2004, p. 584]

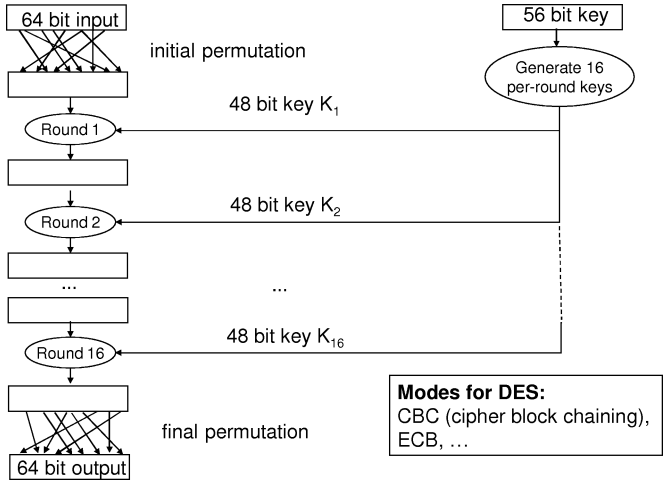


Figure 39: DES En-/Decryption

3DES

See figure 41. 112bit key length.

Advanced Encryption Standard (AES)

- Developed in 2000 by NIST.
- Block cipher that works on 128bit blocks with 128/192/256 bit keys.
- Features:
  - Highly symmetric and parallel structure
  - Robust against all known cryptanalysis attacks
  - Good performance on modern processors

Diffie Hellman Key Exchange

Problem: Establish a common secret over an insecure but authentic channel.

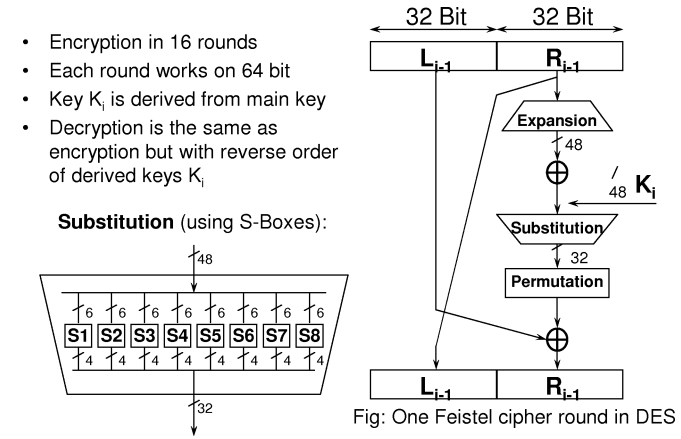


Fig: One Feistel cipher round in DES

Figure 40: DES En-/Decryption Round

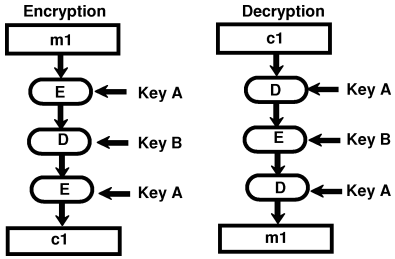


Figure 41: Triple DES En-/Decryption

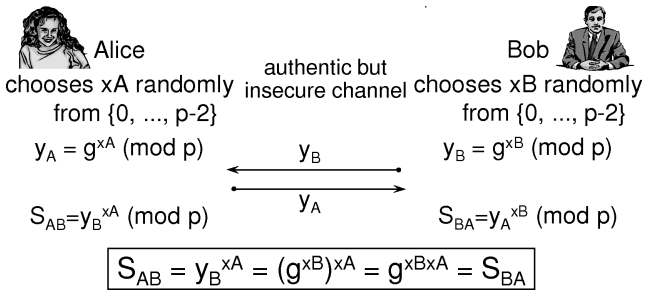


Figure 42: Diffie Hellman

Protocol:

- *Given:* Multiplicative modulo group Z_p^* with generator g; (g, p are public)
  Prime p should be 512-1024 bits
  Generator g should have at least 60 decimals
- *Condition:* Solving the discrete logarithm Z_p^* must be hard!
  The real difficulty for computing the discrete logarithm is not known (NP complete).
- See figure 42.

9.3 Asymmetric Cryptography

Requirements: Alice sends Bob an encrypted message:

- It must be *computationally easy*:
  - for Bob to generate a key pair
  - for sender Alice to generate cipher text
  - for receiver Bob to decrypt cipher test using his private key
- It must be *computationally infeasible*
  - to determine Bob's private key when knowing his public key
  - to recover plain text message when knowing Bob's public key and cipher text
- Either of the two keys can be used for encryption with the other used for decryption

Rivest-Shamir-Adleman (RSA, MIT 1977)

[Peterson and Davie, 2004, p. 588]

Key generation

1. Compute two large primes p and q
2. $m = p \cdot q$ and $f = (p-1) \cdot (q-1)$
3. Choose e such that e and f do not have common dividers, often $e = 3$
4. Compute d such that $e \cdot d = 1 \pmod f$, i.e. $d = e^{-1} \pmod f$
5. *Public key:* (m, e)
6. *Private key:* d

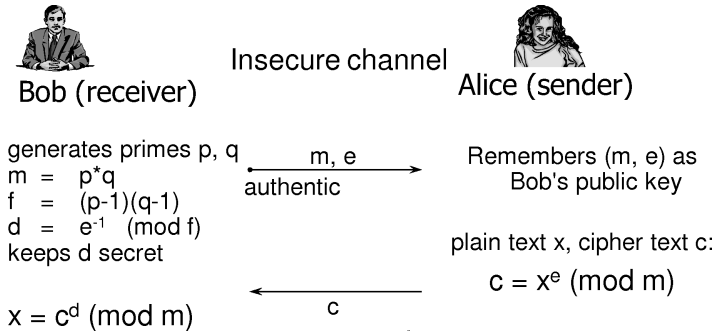


Figure 43: RSA

Protocol See figure 43.

$$c = m^e \mod n \qquad \text{Encoding}$$

$$m = c^d \mod n \qquad \text{Decoding}$$

9.4 Hybrid Encryption: The Digital Envelope

Protocol:

1. Generate a random symmetric session key
2. Encrypt the message symmetrically
3. Encrypt the session key asymmetrically with the receiver's public key
4. Send encrypted message and session key

9.5 Authentication

Message Authentication

[Peterson and Davie, 2004, p. 599]

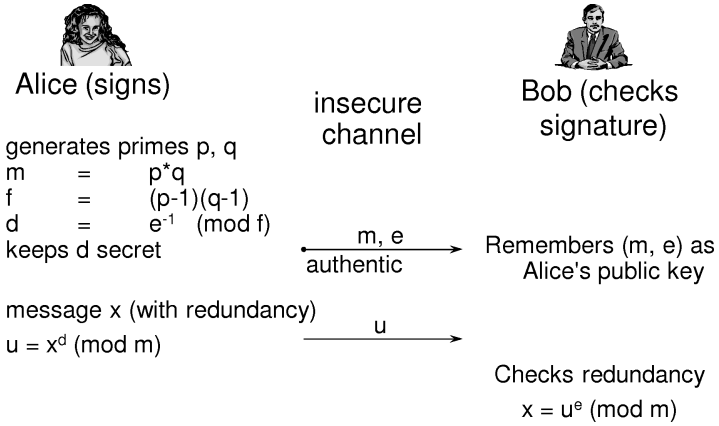


Figure 44: Authentication with Digital Signatures

- With a *symmetric* crypto algorithm:
  Successful decryption implies that the sender knows the key and thus the message is authentic.
  - *Implicit redundancy:* natural or formal language text, executable format, ...
  - *Explicit redundancy:* By checking an explicit authenticator eg a checksum: Message Integrity Code (MIC) or Message Authentication Code (MAC)
- With an *asymmetric* crypto algorithm:
  - Digital signatures, see figure 44.

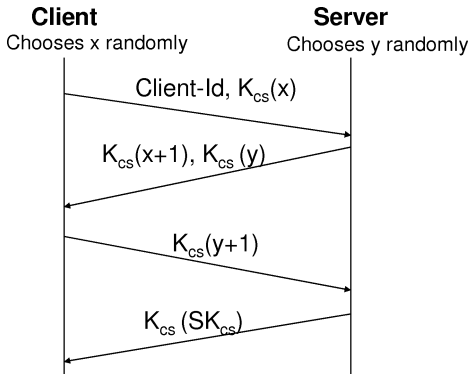
Simple Mutual Authentication

See figure 45.

Certificates: Authentic Public Key Distribution

Distribution Models [Peterson and Davie, 2004, p. 603]

- *X.509*



Legend:

K_{cs} Common secret key
$K_{cs}(x)$ message x symmetrically encrypted with key K_{cs}
SK_{cs} secret symmetric session key
x, y are called "challenges"

Figure 45: Mutual Authentication

  - *Assumption:* The public key of a trusted certification authority (CA) can be distributed in an authentic way.
  - The digital signature of the CA binds the name of the user to his public key.
  - The integrity of a certificate may be checked by anyone over a tree of signed certificates.
- *Web-of-Trust*
  - Verify the authenticity of a key with a network of mutual signatures. It's a decentral alternative to the hierarchical approach.

9.6 Hash Functions

One-way function function f which maps a message m on a message n, such that
- It is easy to compute n from m
- It is difficult to construct a message m' such that $f(m') = n$

Hash function maps a message of arbitrary length on a message of constant length

Message-Digest 5 (MD5)

[Peterson and Davie, 2004, p. 591]
Complicated Hash function that operates on 512 Bit blocks of a message.

9.7 Firewalls: IPTables

Packet Flow

See figure 47.

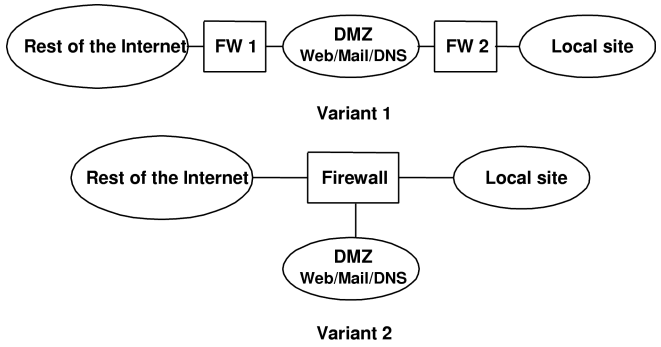


Figure 46: Firewall

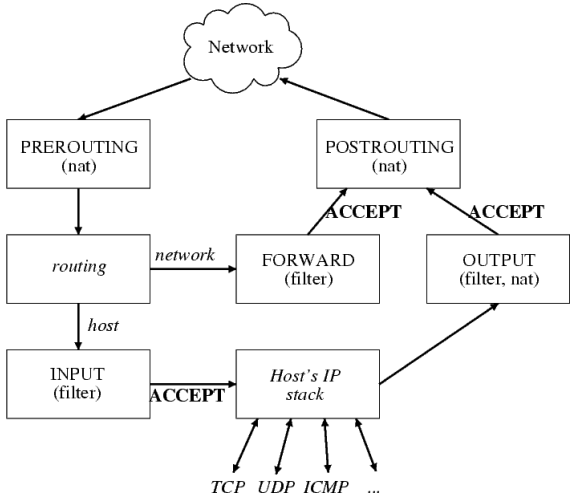


Figure 47: Packet flow through iptables

Rules

If a rule applies, the corresponding directive (target) is executed.

Targets

ACCEPT accept
DROP drop
REJECT drop and inform sender
QUEUE sent to a queue in user space
RETRUN same effect of falling off the end of a chain: for a rule in a built-in chain, the policy of the chain is executed. For a rule in a user-defined chain, the traversal continues at the previous chain, just after the rule which jumped to this chain
LOG log
DNAT rewriting the destination IP address of the packet
`--to-destination ipaddress`
SNAT rewriting the source IP address of the packet
`--to-source <address>[-<address>][:<port>-<port>]`
MASQUERADE Source Network Address Translation
`[--to-ports <port>[-<port>]]`

Configuration: General Match Criteria

`iptables -A INPUT -s 0/0 -i eth0 -d 192.168.1.1 -p TCP -j ACCEPT`

`-t <table>`	If you don't specify a table, then the filter table is assumed. As discussed before, the possible built-in tables include: filter, nat, mangle
`-j <target>`	Jump to the specified target chain when the packet matches the current rule.
`-A <chain>`	Append rule to end of a chain
`-P <chain> [ACCEPT|DROP]`	Default policy for `<chain>`
`-F`	Flush. Deletes all the rules in the selected table
`-p <protocol-type>`	Match protocol. Types include, icmp, tcp, udp, and all
`-s <ip-address>`	Match source IP address `<ip-address> = [!] destination[/prefix]`
`-d <ip-address>`	Match destination IP address
`-i <interface-name>`	Match "input" interface on which the packet enters.
`-o <interface-name>`	Match "output" interface on which the packet exits

Configuration: TCP and UDP Match Criteria

`-p tcp --sport <port>`	TCP source port `<port> = [!] start-port-number[:end-port-number]`
`-p tcp --dport <port>`	TCP destination port
`-p tcp --syn`	Used to identify a new TCP connection request
`-p udp --sport <port>`	UDP source port
`-p udp --dport <port>`	UDP destination port
`--icmp-type <type>`	The most commonly used types are echo-reply and echo-request

Configuration: Common Extended Match Criteria

`-m state --state <state>`	ESTABLISHED, NEW, RELATED (new secondary connection), INVALID

Using User Defined Chains

```
iptables -A FORWARD -s 192.168.4.34 -i eth1 -m state --state NEW -j my-dmz-chain
    iptables -N <chain>   generate new chain
    iptables -X           delete all user-chains
```

10 The New Internet Protocol

10.1 Name and Address Assignment

Organization of Address Assignment

Internet Corporation for Assigned Names and Numbers (ICANN)

- Internationally organized, non-profit
- IP address space allocation: manages pool of available IPs, delegates address blocks to RIRs
- Protocol identifier assignment
- TLD name system management
- Root server system management

Regional Internet Registries (RIR)

- IP assignment, may delegate address assignment to Local Internet Registries (LIR)

10.2 IPv6 Functionality

- Scalability: 128bit addresses
- Security: IPSec for authentication and encryption
- Multicasting: Address format
- Multimedia: Flow label, QoS architecture
- Mobile users: Mobile IP, extension of auto-configuration
- Efficiency: Simple header format

IPv6 Header Format

See figure 48.

Version 6
Priority Traffic class, QoS
Flow Label QoS
Length Length of packet without header in byte
NextHeader Name of the next extension header or the next protocol
eg Fragment, UDP, . . .
Hop Limit TTL

Extension Headers : Routing, Fragmentation, Authentication, Encrypted Security Payload, Hop-by-Hop Options, Destination Options

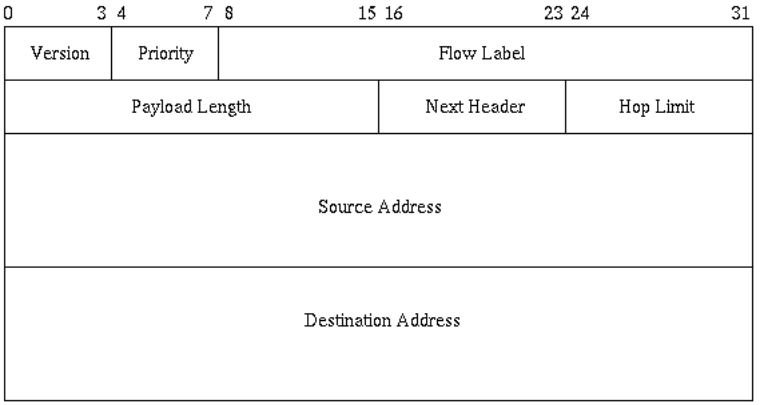


Figure 48: IPv6 Header Format

Adaptions outside of IPv6

- Integrated in IPv6: ICMP, ARP, DHCP, . . .
- TCP, UDP
- Routing protocols
- Security

10.3 Addressing in IPv6

Categories

Unicast Point to point communication
Multicast Group communication
Anycast Send to nearest destination

Notation

- Hexadecimal: FEDC:2A5F:216:AEBC:97:3154:3D12
- Compressed representation: FF08::209A:61
  ::1 (loopback)
- Mixed representation: ::193.136.239.163
- Notation for prefixes: FEDC:BA98:260::/40 (40bit prefix)

Structure of Address Space See table 5.

Global Unicast Format

- Address Format

$$\underbrace{\text{xxxx:xxxx}}_{\text{global routing prefix (n bit)}} : \underbrace{\text{xxxx:xxxx}}_{\text{subnet id (64-n bit)}} : \underbrace{\text{xxxx:xxxx:xxxx:xxxx}}_{\text{interface id (64bit)}}$$

Prefix	*Allocation*
0000 0000	Reserved
0000 0001	Unassigned
0000 001	Reserved for NSAP Allocation
0000 010	Reserved for IPX Allocation
0000 011	Unassigned
0000 1	Unassigned
0001	Unassigned
001	*Global Unicast Address*
010	Unassigned
011	Unassigned
100	Unassigned
101	Unassigned
110	Unassigned
1110	Unassigned
1111 0	Unassigned
1111 10	Unassigned
1111 110	Unassigned
1111 1110 0	Unassigned
1111 1110 10	Link-local Addresses
1111 1110 11	Site-local Addresses
1111 1111	Multicast Addresses

Table 5: Structure of Address Space

Global Part global routing prefix, (001+site identifier)
Used by Inter-domain routers
Local Part subnet id + interface id.
Used by Intra-domain routers

- Addresses currently allocated:

$$\underbrace{\text{001x:xxxx:xxxx}}_{\text{site identifier (3+45 bit)}}:\underbrace{\text{xxxx}}_{\substack{\text{subnetting within}\\ \text{site (16bit)}}}:\underbrace{\text{xxxx:xxxx:xxxx:xxxx}}_{\text{interface id (64bit)}}$$

Address allocation is by /48 global routing prefix.

Special Addresses

IPv4-compatible IPv6 Address Represents an IPv4 address in fixed-size table of IPv6 addresses
::193.136.239.163

IPv4-mapped IPv6 Address Transparent use of transport layer protocols (TCP or UDP) over IPv4 through the IPv6 networking API
::FFFF:193.136.239.163

Link-local Addresses Works in a subnet without being globally unique

$$\underbrace{\text{1111:1110:10}}_{\text{link-local prefix}}\underbrace{\text{00:0000}}_{\text{(24bit)}}:\underbrace{\text{xxxx:xxxx:xxxx:xxxx}}_{\text{interface id (64bit)}}$$

Site-local Addresses Works in a network but is not globally unique

$$\underbrace{\text{1111:1110:11}}_{\text{site-local prefix}}\underbrace{\text{00:0000}}_{\text{(24bit)}}:\underbrace{\text{xxxx}}_{\text{subnet id (16bit)}}:\underbrace{\text{xxxx:xxxx:xxxx:xxxx}}_{\text{interface id (64bit)}}$$

Anycast Addresses Finds one of several equivalent services.

- Routing system delivers packet to nearest destination.
- Can not be distinguished from normal address.
- Can not be used as a source address, server answers with his unicast address.

Subnet Router Anycast Address Used to identify any router accepting packets for a specific subnet.

$$\underbrace{\text{xxxx:xxxx:xxxx:xxxx:xx}}_{\text{Subnet prefix (n bit)}}\underbrace{\text{00:0000:0000:0000}}_{\text{0 (128-n bit)}}$$

IPv6 to IPv4 Addresses

- Address Format

$$\underbrace{\underbrace{\text{2002}}_{\text{prefix (32bit))}}:\underbrace{\text{193.136.239.163}}_{\substack{\text{IPv4 address of}\\ \text{gw node (32bit)}}}}_{\text{global /48 prefix}}:\underbrace{\text{xxxx}}_{\text{subnet id (16bit)}}:\underbrace{\text{xxxx:xxxx:xxxx:xxxx}}_{\text{interface id (64bit)}}$$

10.4 Routing for IPv6

Routing Information Protocol for IPv6 (RIP)

See section 4.1 on page 11 for RIP for IPv4.

Header Format See figure 49.
Commands Request / response

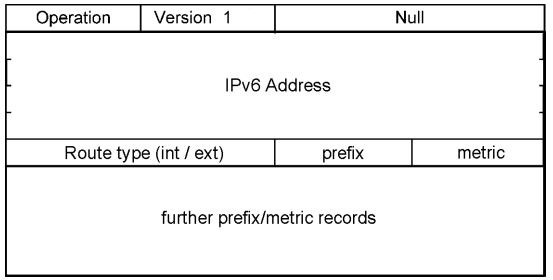


Figure 49: Routing Information Protocol for IPv6 (RIP)

Open Shortest Path First for IPv6 (OSPF)

See section 4.3 on page 12 for OSPF for IPv4.

- OSPF was adapted minimally for IPv6. Link State Records and Areas are identified with an IPv6 address or prefix.
- OSPFv6 runs in parallel with IPv4 OSPF

Inter-domain Routing

- BGP4 cannot be easily adapted
- Adaption of Inter-Domain Routing Protocol (IDRP) conceptually related with BGP

11 Traditional Applications

11.1 Simple Mail Transfer Protocol (SMTP)

[Peterson and Davie, 2004, p. 650]

Mail Gateway Buffers and tries to forward Mails. Listens on port 25.

cs.princeton.edu forwards a message to cisco.com

```
HELO cs.princeton.edu
250 HELLO daemon@mail.cs.princeton.edu [128.12.169.24]

MAIL FROM: <Bob@cs.princeton.edu>
250 OK

RCPT TO: <Alice@cisco.com>
250 OK

RCPT TO: <Tom@cisco.com>
550 No such user here

DATA
354 Start mail input; end with <CRLF>.<CRLF>
Blah blah blah ...
...etc.  etc.  ...
<CRLF>.<CRLF>
250 OK

QUIT
221 Closing connection
```

11.2 MIME

[Peterson and Davie, 2004, p. 646]

```
MIME-Version:  1.0
Content-Type:  multipart/mixed; boundary="---417CA6E2DE4ABCAFBC5"
From:  Alice Smith <alice@cisco.com>
To:  Bob@cs.Princeton.edu
Subject:  promised material
Date:  Mon, 07 Sep 1998 19:45:19 -0400

---417CA6E2DE4ABCAFBC5
Content-Type:  text/plain; charset=us-ascii
Content-Transfer-Encoding:  7bit

Bob, Here's ...
Alice


---417CA6E2DE4ABCAFBC5
Content-Type:  image/jpeg
Content-Transfer-Encoding:  base64
.
.  unreadable encoding of a jpeg figure
.
```

Newline with <CRLF>.

Special Header-Fields

Date sent

Message-ID Global unique identifier, inserted by sending mail-program or smtp gateway.
Format: <random>@domain

Received Received : [from <sending-host>] [by <receiving-host>] [via <physical path>] [for <initial-form]
<hosts> are of the form <real name> (<claimed name> [ip address]

11.3 World Wide Web (HTTP)

[Peterson and Davie, 2004, p. 653]

HTTP-Message Format

```
START_LINE <CRLF>
MESSAGE_HEADER <CRLF>
<CRLF>
MESSAGE_BODY <CRLF>
```

Request eg with telnet <host> <port>

```
GET index.html HTTP/1.1 <CRLF>
HOST: www.cs.princeton.edu <CRLF><CRLF>
```

OPTIONS Returns the HTTP methods that the server supports.
GET Requests a representation of the specified resource.
Options:
- `If-Modified-Since: <date>`
  return if document modified or a 304 response

HEAD Asks for the response identical to the one that would correspond to a GET request, but without the response body.
POST Submits data to be processed.
PUT Uploads a representation of the specified resource.
DELETE Deletes the specified resource.
TRACE Echoes back the received request, so that a client can see what intermediate servers are adding or changing in the request.
CONNECT To connect to `<server>` over a proxy.

```
CONNECT <server>:<port> HTTP/1.0
User-agent: Mozilla/4.0
Proxy-authorization: basic dGVzdDp0ZXN0
```

Response

```
HTTP/1.1 202 Accepted
```

`1xx`	Informational	Request received, continuing process.
`2xx`	Success	200: OK; 201: Created; 202: Accepted; 204: No Content; ...
`3xx`	Redirection	301: Moved Permanently; 302: Moved Temporarily (HTTP/1.0) ... `Location: http://redirect.com`
`4xx`	Client Error	400: Bad Request; 401: Unauthorized; 403: Forbidden; 404: Not Found ...
`5xx`	Server Error	500: Internal Server Error; 505: HTTP Version Not Supported; ...

TCP-Connections See section 7.1 on page 17.
- *HTTP/1.0*
  By default one connection for every element. Persistent connections possible with the `Connection: Keep-Alive` option.
- *HTTP/1.1*
  Persistent connection between client and server.
  - Less overhead (less setups and terminations of TCP-connections)
  - Better congestion control, less slow-starts (see section 7.2 on page 18)
  - Problem: more open connections, when to close it?

11.4 Network Management (SNMP)

[Peterson and Davie, 2004, p. 659]

References

[Peterson and Davie, 2004] Peterson, L. L. and Davie, B. S. (2004). *Computer-netze*. dpunkt.verlag GmbH, Heidelberg, 3. edition.

Index