Internet Security [1]
VU 184.216

Engin Kirda engin@infosys.tuwien.ac.at
Christopher Kruegel chris@auto.tuwien.ac.at

Layer 4 Protocols

Many protocols use IP as the underlying network layer

- Important ones are

  ICMP (Internet Control Message Protocol)

  UDP (User Datagram Protocol)

  TCP (Transmission Control Protocol)

ICMP

ICMP (Internet Control Message Protocol)

- is used to exchange control/error messages about the delivery of IP datagrams
- ICMP messages are encapsulated inside IP datagrams

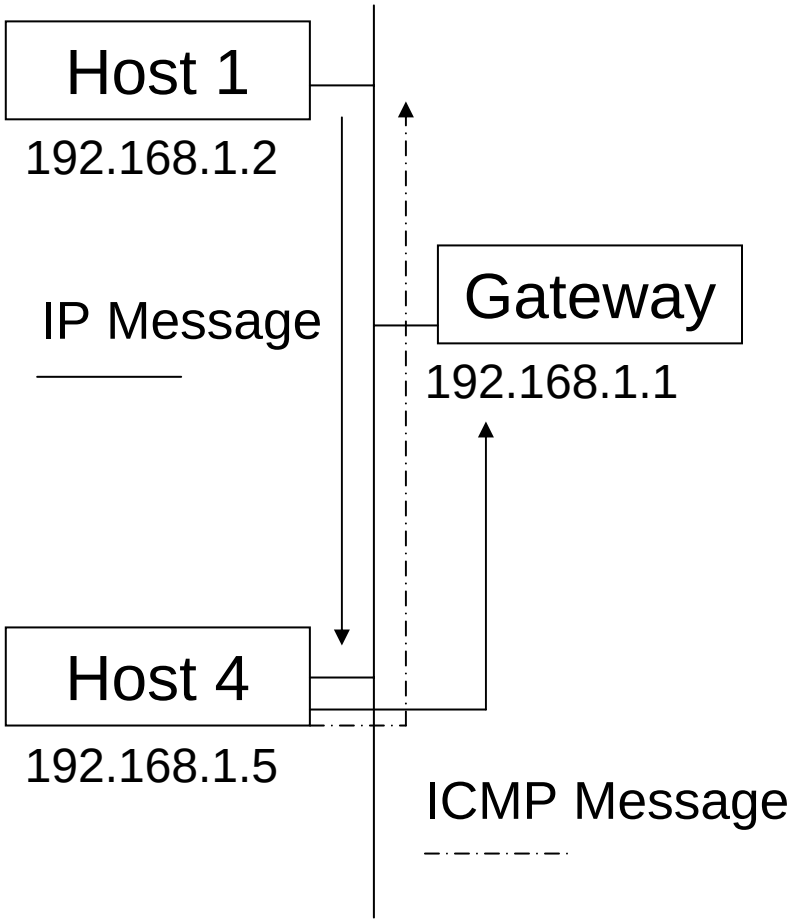
- ICMP messages can be:
  - Requests
  - Responses
  - Error messages
    - includes header and first 8 bytes of offending IP datagram

ICMP Redirect

- Is used for stating that there is a better route to a host/net
- Is sent by a router that routes a packet over the same interface that was used for receiving this packet

type (= 5)	code	checksum
IP address of the router that should be used		
IP header + first 8 bytes of the original datagram		

ICMP Redirect - Example



1) In Host1‘s configuration, it is stated to use Host4 as a gateway. So when Host1 sends a packet outside the subnet, this is forwarded to Host4.

2) Host4 gets the packet, but has to forward the packet to Gateway.

3) Additionally Host4 sends Host1 an ICMP redirect message. „The net xxx can be reached better via Gateway yyy”.

ICMP Redirect

- A host that receives an ICMP redirect message checks:
  - whether the new router is directly connected to the network
  - the redirect must be from the current router for this destination
  - the redirect can‘t tell the host to use itself as the router
  - the route that is being modified has to be an indirect route

- What is not checked
  - is message really sent by the current router?
  - is the target host (the new router) a router?

ICMP Redirect Attacks

- ICMP redirect messages can be used to re-route traffic on specific routes or to a specific host that is not a router at all
- The attack is very simple: just send a spoofed ICMP redirect message that appears to come from the host‘s default gateway
- Can be used to
  - hijack traffic
  - perform a denial of service attack

ICMP Dest. Unreachable

- ICMP message used by gateways to state that the datagram cannot be delivered
- Many subtypes
  - Network unreachable
  - Host unreachable
  - Protocol unreachable
  - Port unreachable
  - Fragmentation needed but don‘t fragment bit set
  - Destination host unknown
  - Destination network unknown etc.

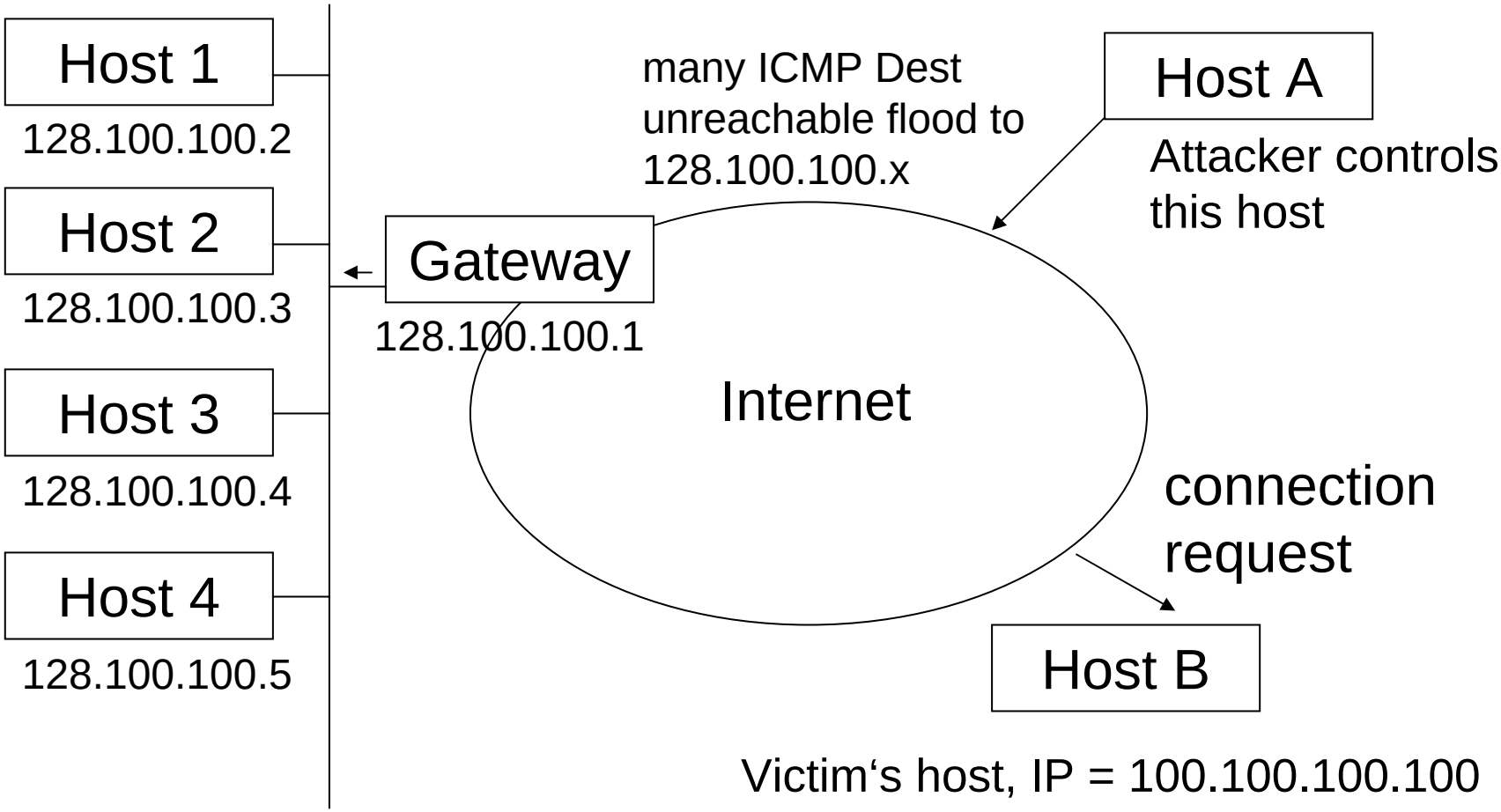
Dest. Unreachable Attack

- Can be used to „cut“ out nodes from the network
- Is a denial of service attack (DOS)

Example

An attacker injects many forged destination unreachable messages stating that 100.100.100.100 is unreachable into a subnet (e.g. 128.100.100.). If 128.100.100.2 net tries to connect to 100.100.100.100, he will immediately get an ICMP Dest. Unreachable from the attacker‘s host. For 128.100.100.2, this means that there is no way to contact 100.100.100.100, and therefore communication fails.

Dest. Unreachable Attack



ICMP Time Exceeded

Used when

- TTL becomes zero (code =0)
- The reassembling of a fragmented datagram times out (code=1)

type (=11)	code (0 or 1)	checksum
unused		
IP header + first 8 bytes of the original datagram		

Traceroute

- Program to determine the path to a specific host/net by evaluating ICMP Time Exceeded messages
- Does this by
  - sending a series of IP datagrams to the destination node
  - each datagram has an increasing TTL field (start=1)
  - gets back ICMP Time Exceeded messages by the intermediate gateways
  - so the full path can be reconstructed by Traceroute
- Traceroute also allows to use loose source routing
- Useful tool for topology mapping

User Datagram Protocol

UDP (User Datagram Protocol)

- relies on IP
- connectionless
- unreliable (checksum optional)
- best-effort
- datagram delivery service

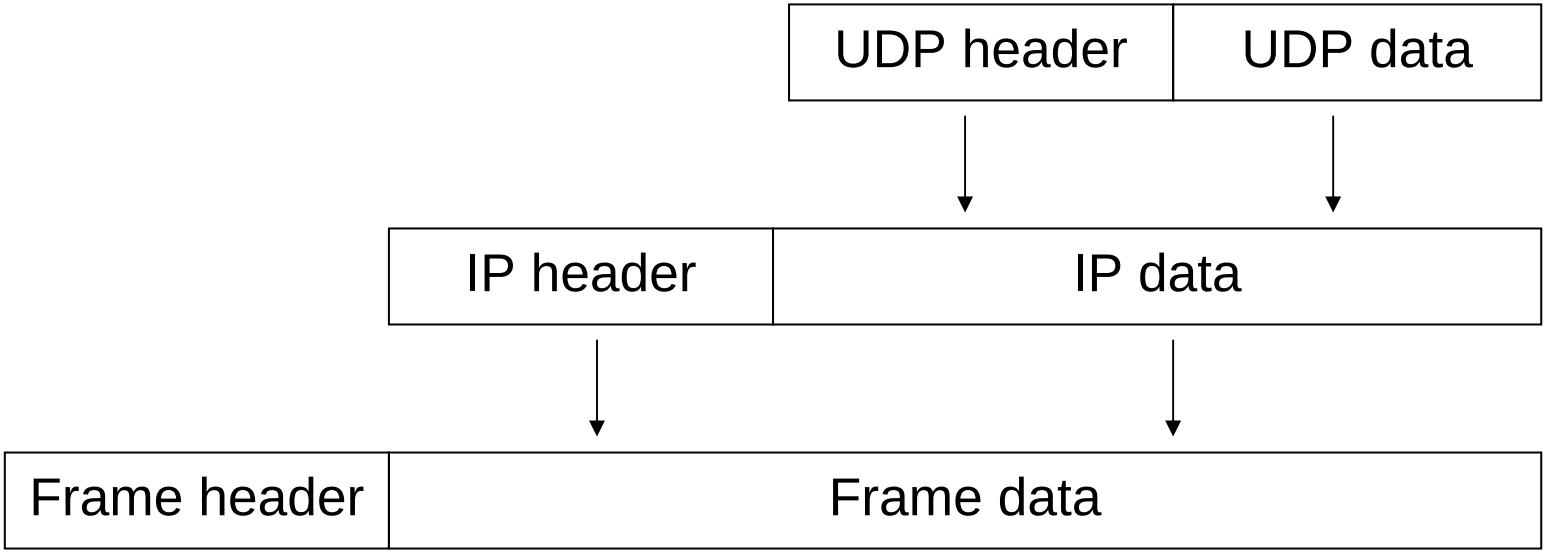
➢ delivery, integrity, non-duplication and ordering are not guaranteed

UDP Message

- Port abstraction
  - allows addressing different destinations for the same IP
- Often used for multimedia
  - and for services based on request/reply schema (DNS, RPC, NFS)
  - more efficient than TCP

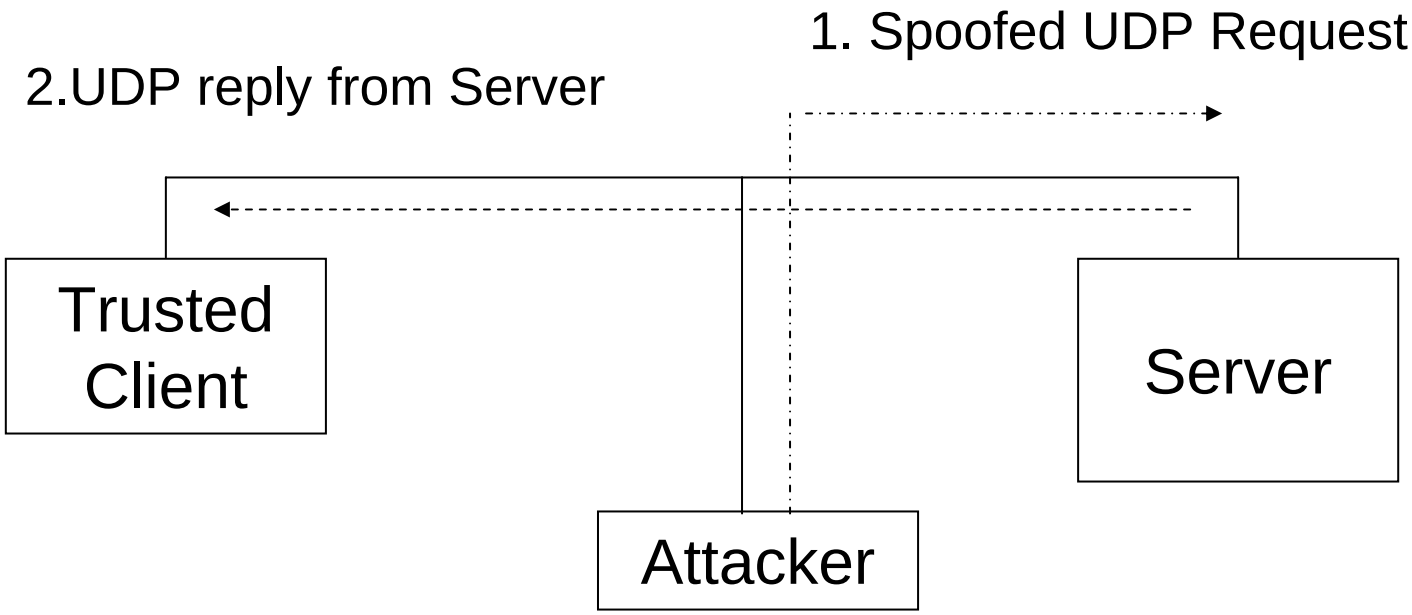
UDP source port (2 bytes)	UDP destination port (2)
UDP message length (2)	Checksum (2)
Data	

UDP Encapsulation



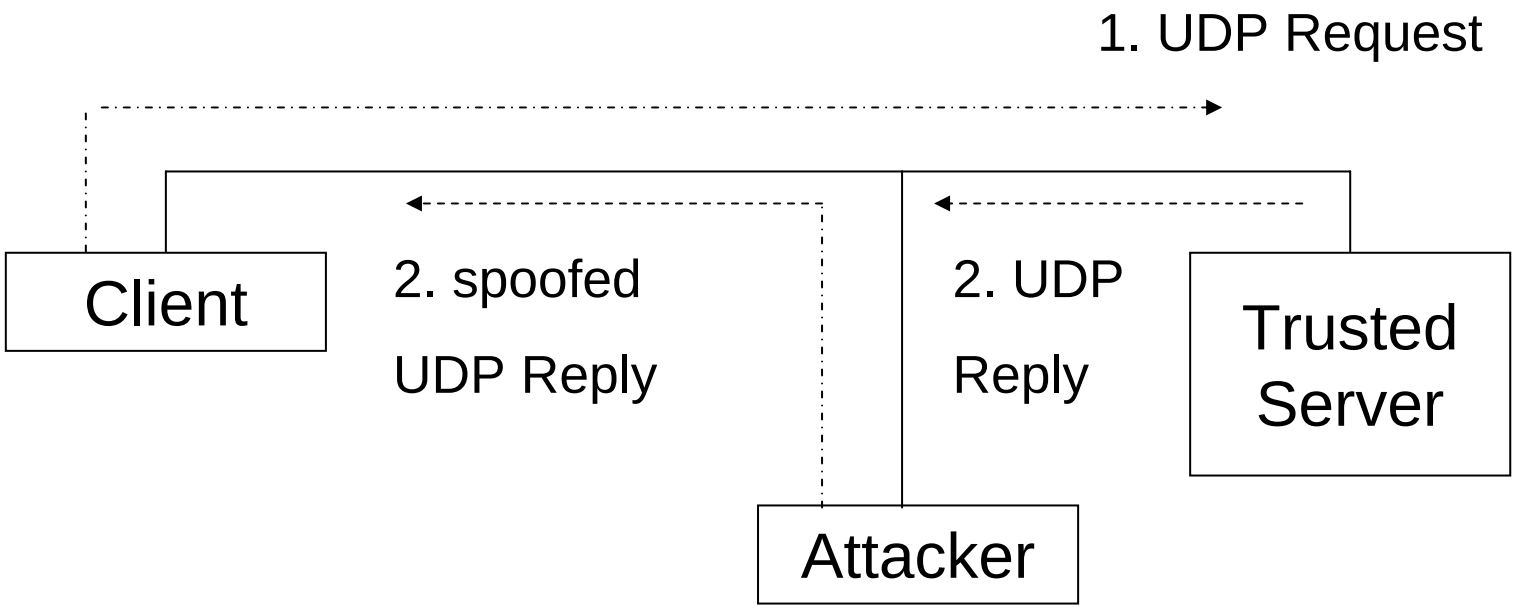
UDP Spoofing

- Basically IP spoofing, as easy to perform



UDP Hijacking

- Variation of the UDP spoofing attack
- Racing against the legitimate server



UDP Storms

- Spoofed UDP datagram is sent to the echo service (port 7)
- Source port is set to the chargen device (port 19)
- Reply of the echo service is interpreted as a request by the chargen service
- Reply of the chargen service is interpreted as a request by the echo service
- ... etc …
- Same attack can be carried out using two echo services

UDP Portscan

- Used to determine which UDP services are available
- Zero-length UDP packet is sent to each port
- If an ICMP error message „port unreachable“ is received, the service is assumed to be unavailable
- Many TCP/IP stack implementations implement a limit on the error message rate, therefore this type of scan can be slow (e.g. Linux limit is 80 messages every 4 seconds)

UDP Portscan

How to do a UDP portscan?

- by hand (with packet filter and RAW-socket)
- use netcat (http://netcat.sourceforge.net/) and tcpdump
- or use e.g. nmap -sU <address> (http://www.insecure.org/nmap/)

TCP

TCP (Transmission Control Protocol)

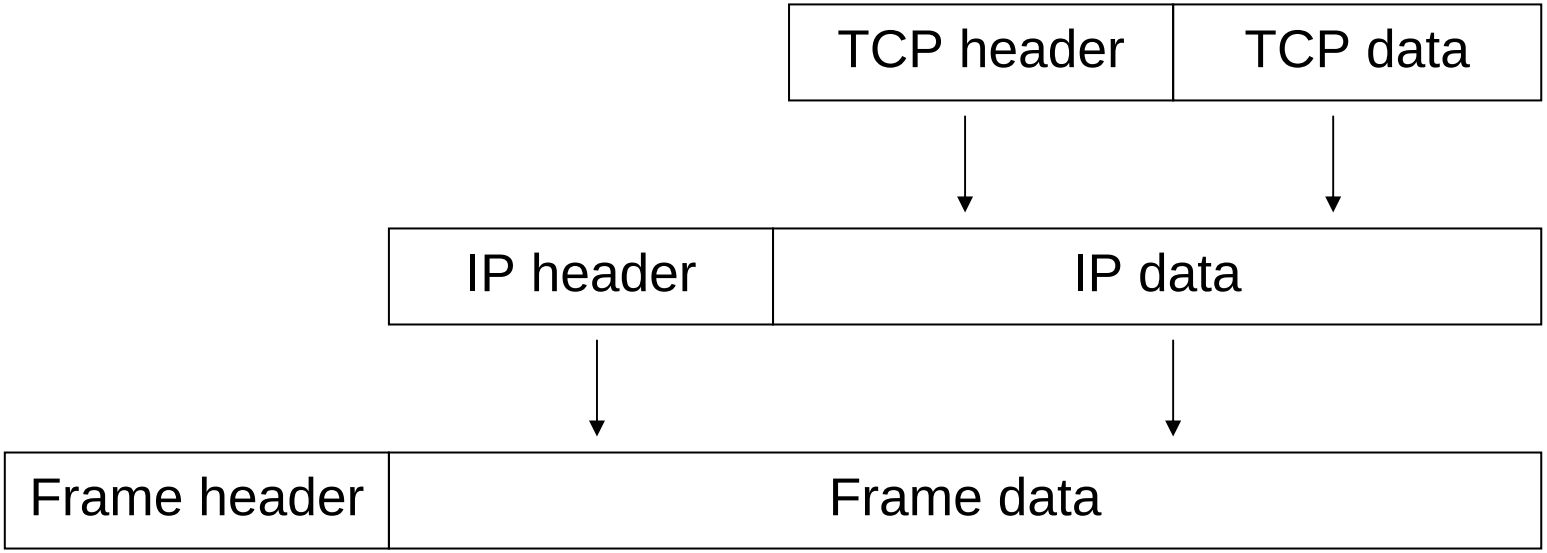
- relies on IP to provide
- connection-oriented
- reliable
- stream delivery service

➢ no loss, no duplication, no transmission errors, correct data ordering

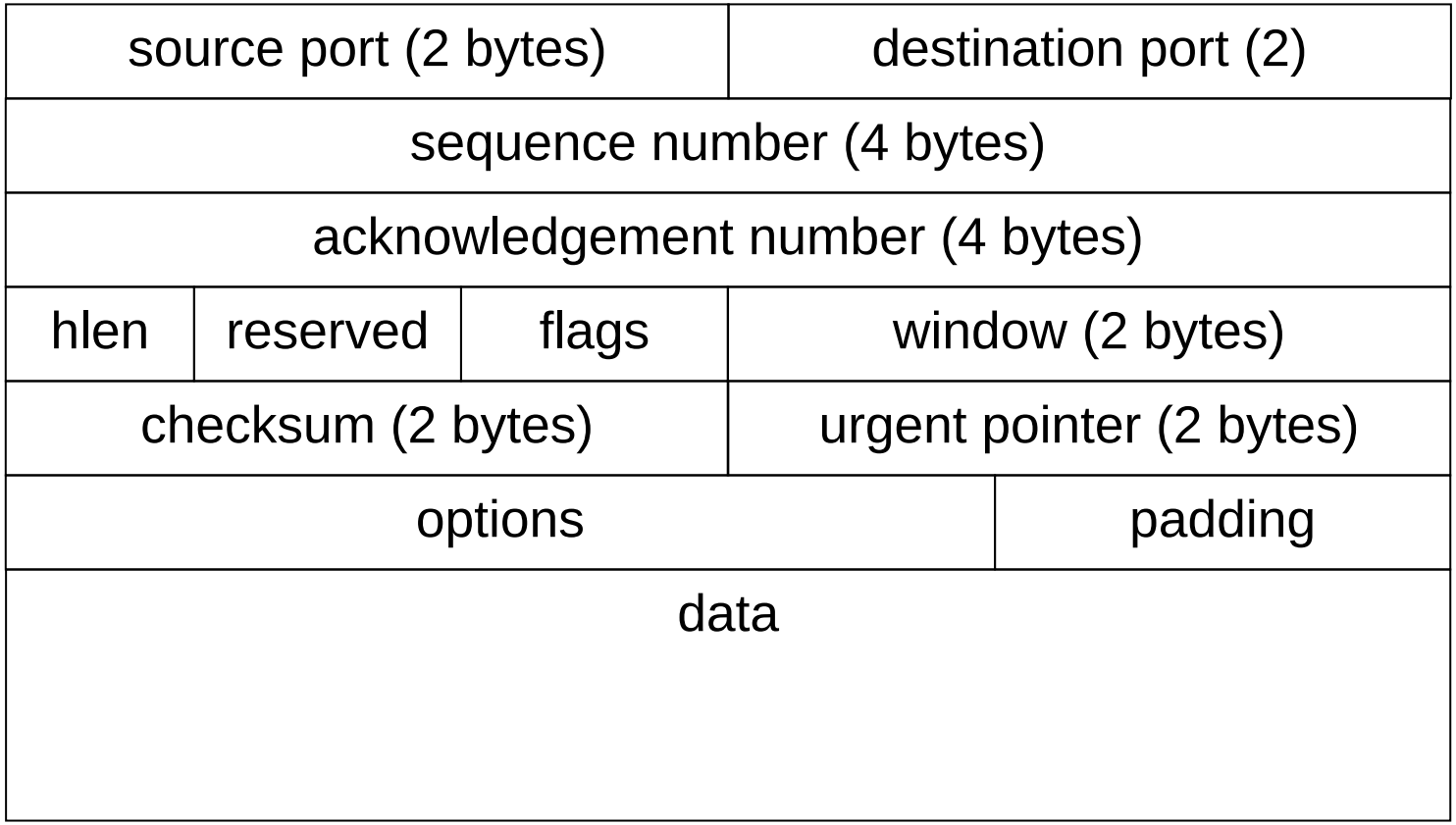
TCP

- Provides (like UDP) the port abstraction
- Allows two nodes to establish a virtual circuit
  - identified with quadruples
  - <src_ip, src_port, dst_ip, dst_port>
  - virtual circuit is composed of two streams (full duplex)
- The pair <IP address, port> is called a *socket*

TCP Encapsulation



TCP Segment


source port (2 bytes)
destination port (2)
sequence number (4 bytes)
acknowledgement number (4 bytes)
hlen
reserved
flags
window (2 bytes)
checksum (2 bytes)
urgent pointer (2 bytes)
options
padding
data

TCP Seq/Ack Numbers

- Sequence number (seq)
  - specifies the position of the segment data in the communication stream
  - seq = 1234 means:

    The payload of this segment contains data starting from 1234

- Acknowledgement number (ack)
  - specifies the position of the *next expected byte* from the communication partner
  - ack = 12345 means:

    I have received the bytes correctly to 12344, I expect the next byte to be 12345

- Both are used to manage error control
  - retransmission, duplicate filtering

TCP Window

- Used to perform flow control
- Segment will be accepted only if the sequence number has a value between
  - last ack number sent and
  - last ack number sent + window size
- The window size changes dynamically to adjust the amount of information that can be sent by the sender
  - set by the receiver to announce how much it can take
  - window size = amount of data the client can handle now

TCP Flags

- Flags are used to manage the establishment and shutdown of a virtual circuit
  - SYN: request for synchronization of seq/ack numbers (used during connection setup)
  - ACK: the acknowledgement number is valid (all segments in a virtual circuit have this flag set, except the first)
  - FIN: request to shutdown a virtual circuit
  - RST: request to immediately reset the virtual circuit
  - URG: states that the urgent pointer is valid
  - PSH request a „push“ operation on the stream (pass the data to the application (interactive) as soon as possible)

TCP Options

Most important

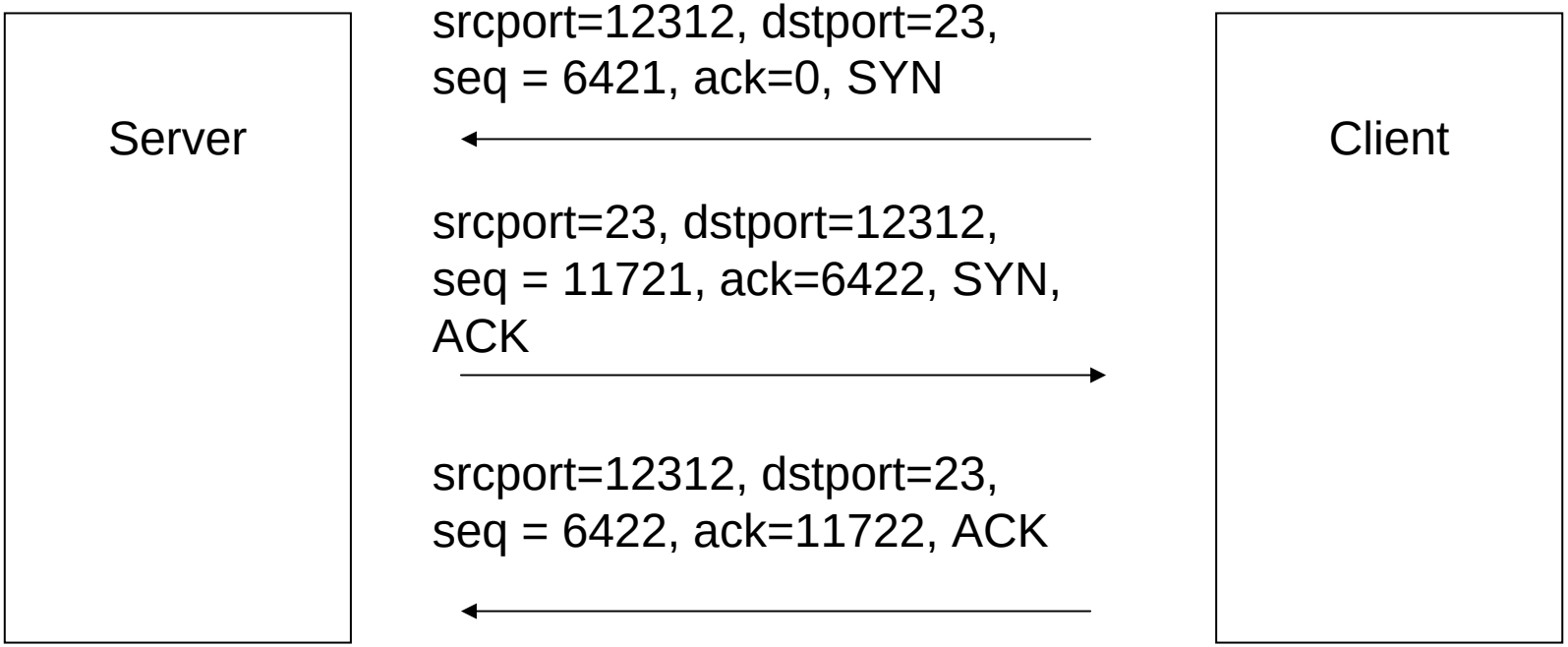
- Maximum Segment Size (MSS)
  - the size of the largest packet a partner is able to receive
  - set during setup phase
- Window scale factor
  - allows to specify a larger window of TCP segments to accept
- Timestamp
  - for TCP-Echo requests + responses (similar to ICMP)

TCP Virtual Circuit :: Setup

- A server listens to a specific port
- Client sends a connection request to the server, with SYN flag set and a random initial sequence number c
- The server answers with a segment marked with both the SYN and ACK flags and containing
  - an initial random sequence number s
  - c+1 as the acknowledge number
- The client sends a segment with the ACK flag set and with sequence number c+1 and ack number s+1

Three Way Handshake

- Three way because 3 TCP segments are necessary to set up a virtual circuit



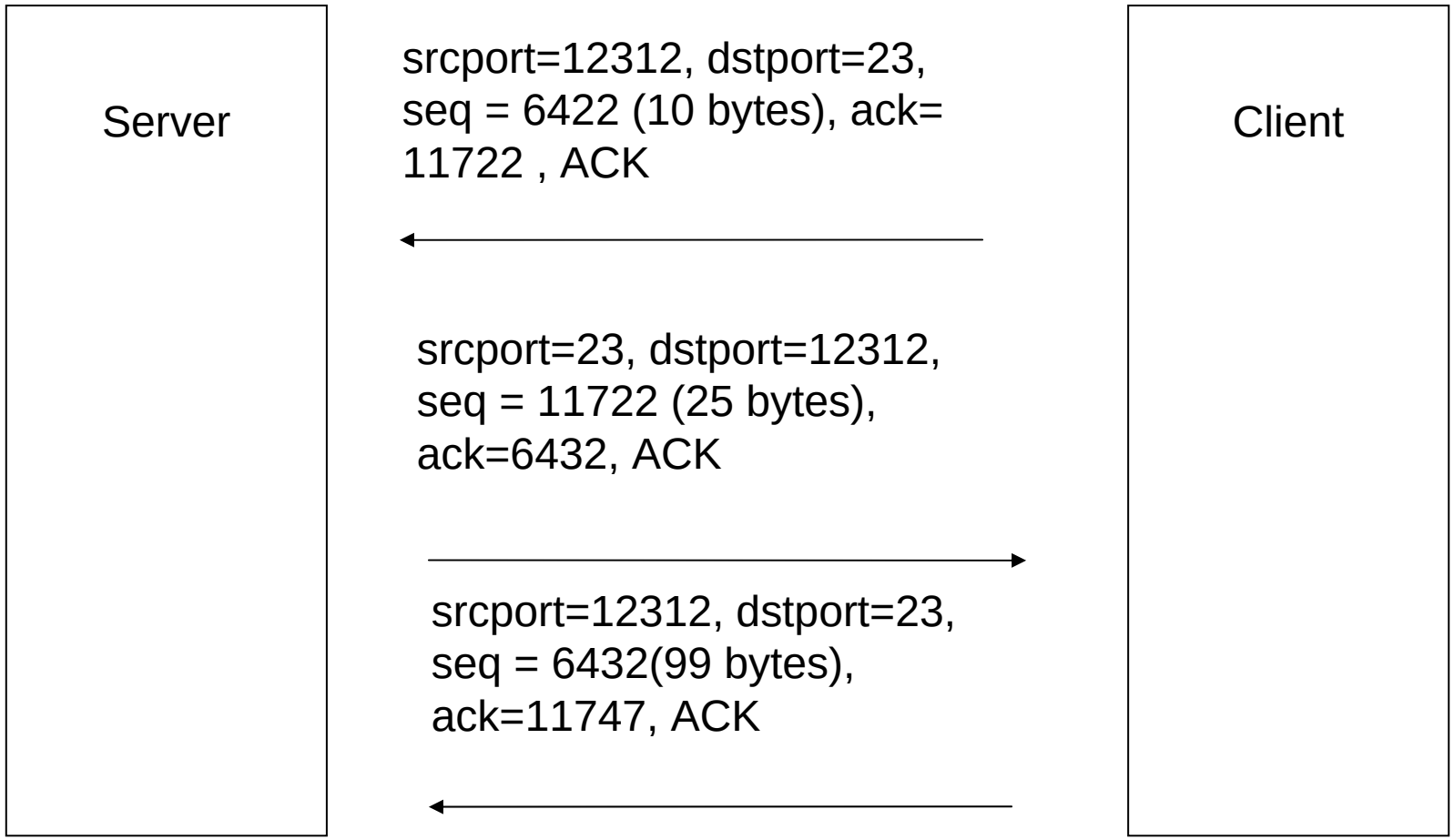
Initial Sequence Number

- The TCP standard (RFC 793) specifies that the sequence number should be incremented every 4 µs
- BSD UNIX systems initially used a number that is incremented by 64000 every half second and by 64000 each time a connection is established
- This number is important, because it adds protection against simple attacks.

TCP Data Exchange

- Each TCP segment contains
  - sequence nr = position of data in stream (often, last ack number)
  - ack nr = sequence number of last correctly received segment increased by the payload size of this segment
- A partner accepts a segment of the other partner only if the numbers are inside the transmission window
- An empty segment may be used to acknowledge the received data
- Packets with no payload and SYN or FIN flag consume one sequence number

TCP Data Exchange



Acknowledgement

- Not sent directly after data has been received
- delayed ACK: if some data has been received, the receiver waits up to 200 ms in hope that some more data will arrive, which can be acknowledged at once.
- delayed ACK is only used if no data has to be transported back to the sender

If no ACK is received at the sender (timeout), retransmission takes place

Virtual Circuit :: Shutdown

- One of the partners, e.g. A, can terminate its stream
  - by sending a segment with the FIN flag set
- B answers with a segment with the ACK flag set
- From this point on A will not send any data to B: it will just acknowledge data sent by B
  - with empty segments
- When B shuts its stream down, the virtual circuit is considered closed

Sample TCP Connection

From	To	S	A	F	Seq-Nr	Ack-Nr	Payload
192.168.0.1	192.168.0.2	1	0	0	4711	0	0
192.168.0.2	192.168.0.1	1	1	0	38001	4712	0
192.168.0.1	192.168.0.2	0	1	0	4712	38002	0
192.168.0.2	192.168.0.1	0	1	0	38002	4712	‚Login:\n‘ 7
192.168.0.1	192.168.0.2	0	1	0	4712	38009	‚s‘ 1
192.168.0.1	192.168.0.2	0	1	0	4713	38009	‚e‘ 1
192.168.0.1	192.168.0.2	0	1	0	4714	38009	‚c‘ 1
192.168.0.1	192.168.0.2	0	1	0	4715	38009	‚\n‘ 1
192.168.0.2	192.168.0.1	0	1	0	38009	4716	0
192.168.0.1	192.168.0.2	0	0	1	4716	38009	0
192.168.0.2	192.168.0.1	0	1	0	38009	4717	0

TCP Security

- Scanning
- OS Fingerprinting
- TCP Spoofing
- TCP Hijacking
- Denial of Service
  - SYN flooding
  - Process Table Attack
  - Land Attack

TCP Scanning

- Used to check whether a port is open on a host
  - `/etc/services` lists standard port/service mappings
- Should be done without letting monitored host know that it is scanned (OS/tools can log connections)
- Used to get some extra information about the host

In the simplest form a TCP connection is opened to a ports

- if this succeeds a service is assumed to be available

TCP SYN Scanning

- Also known as „half open“ scanning
- The attacker sends a SYN packet (packet with SYN flag)
  - If the server answers with a SYN/ACK packet, then the port is open (or with a RST packet: the port is closed)
- The attacker sends a RST packet instead of an ACK

➢ Therefore the connection is never opened and the event is not logged by the operating system / monitor application

TCP FIN Scanning

- The attacker sends a FIN-marked packet
- In most TCP/IP implementations (not Windows)
  - if the port is closed, a RST packet is sent back
  - if the port is open, the FIN packet is ignored
- Variations of this type of scanning technique
  - XMAS Scan: FIN + PSH + URG set
  - NULL Scan: no flags set

OS Fingerprinting

- OS Fingerprinting
  - allows to determine the operating system of a host by examining the reaction to carefully crafted packets
  - use of reserved flags in the TCP header
  - use of weird combination of flags in the TCP header
  - check the selection of TCP initial sequence numbers
  - analysis of response to particular ICMP messages
  - server response at a special port (Login)

NMAP

- Is a tool for performing portscans and for OS fingerprinting
- http://www.insecure.org/nmap/
- supports
  - IP scans
  - UDP portscans
  - TCP portscans (SYN, FIN scanning)
  - OS fingerprinting

TCP Spoofing

Attack aimed at impersonating another host

mostly during the TCP connection establishment phase

- Node A trusts node B (e.g. login with no password)
- Node C wants to impersonate B with respect to A in opening a TCP connection
- C kills B (flooding, redirecting, crashing)
- C sends A a TCP segment in a spoofed IP packet with B‘s address as the source IP and an initial sequence number T

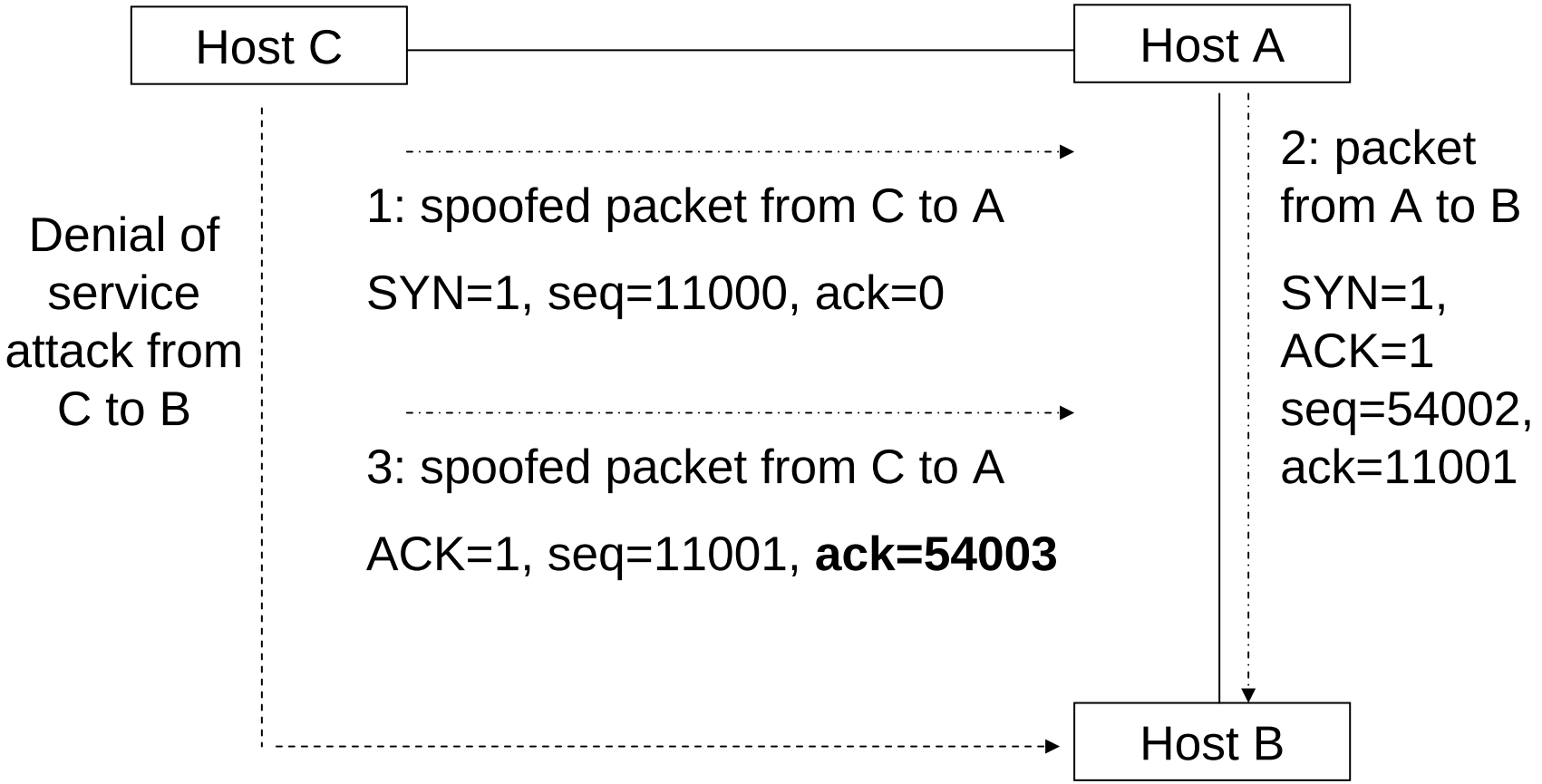
TCP Spoofing

- A replies with a TCP SYN/ACK segment to B with S as the sequence number and T+1 as acknowledge number
- C does not receive the segment from A to B, but in order to finish the handshake it has to send an ACK segment with S+1 as the acknowledge number to A

for this two possibilities exist:

- C eavesdrops the SYN/ACK segment and calculates the number
- C guesses the correct sequence number

TCP Spoofing



TCP Hijacking

- Technique to take control of an existing TCP connection
- The attacker uses spoofed TCP segments to
  - insert data into the streams
  - reset existing connections & reopen them
- But the correct sequence/acknowledgement numbers must be used (guessed or eavesdropped by the attacker)

DOS TCP Attacks

SYN flooding attack (known as Neptune attack)

- very common denial-of-service attack
- attacker starts handshake with SYN marked segment
- victim replies with SYN-ACK segment
- attacker‘s host stays silent
- a host can only keep a limited number of TCP connections in half-open state. After that limit, connections are not accepted.

- Current solution
  - drop half open connections in FIFO manner
  - SYN cookies

DOS TCP Attacks

Process Table Attack

- Daemons are programs that listen on a particular port for connection requests
- When a new connection is established the daemon
  - forks a new process that will handle the connection
  - waits for the next connection

Many daemons run with root privileges (no restrictions)

A huge number of connections fill up the process table and

no new processes can be created

DOS TCP Attacks

Land Attack

- A TCP segment with the SYN flag set is sent to an open port
- The source address and port are the same as the destination address/port
- The host starts an „internal“ ACK storm, which is very CPU intensive
  - tries to open connections to a port that is already in use
  - sequence numbers (a new one is generated when the SYN segment arrives) don‘t match which results in an ACK-storm

Conclusions

- Internetworking
  - local delivery
    - data link layer
    - point-to-point frame exchange (Ethernet, ARP)
  - end-to-end connection
    - network layer□
    - packet switching (IP)
    - routing (routing protocols)
  - additional services for end-to-end connections
    - transport layer
    - error and notification service (ICMP)
    - ports (UDP, TCP)
    - reliable transmission (TCP)