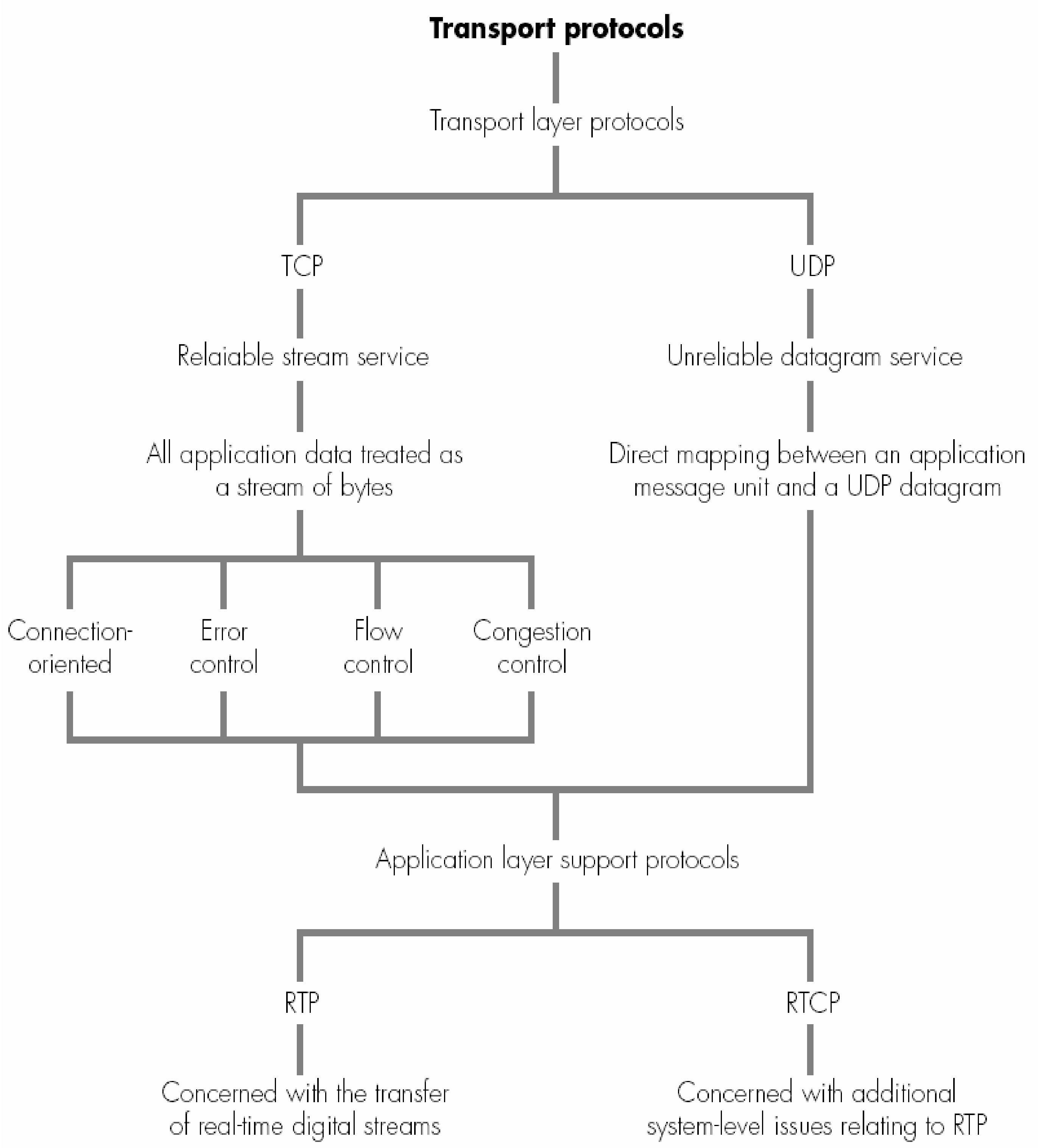
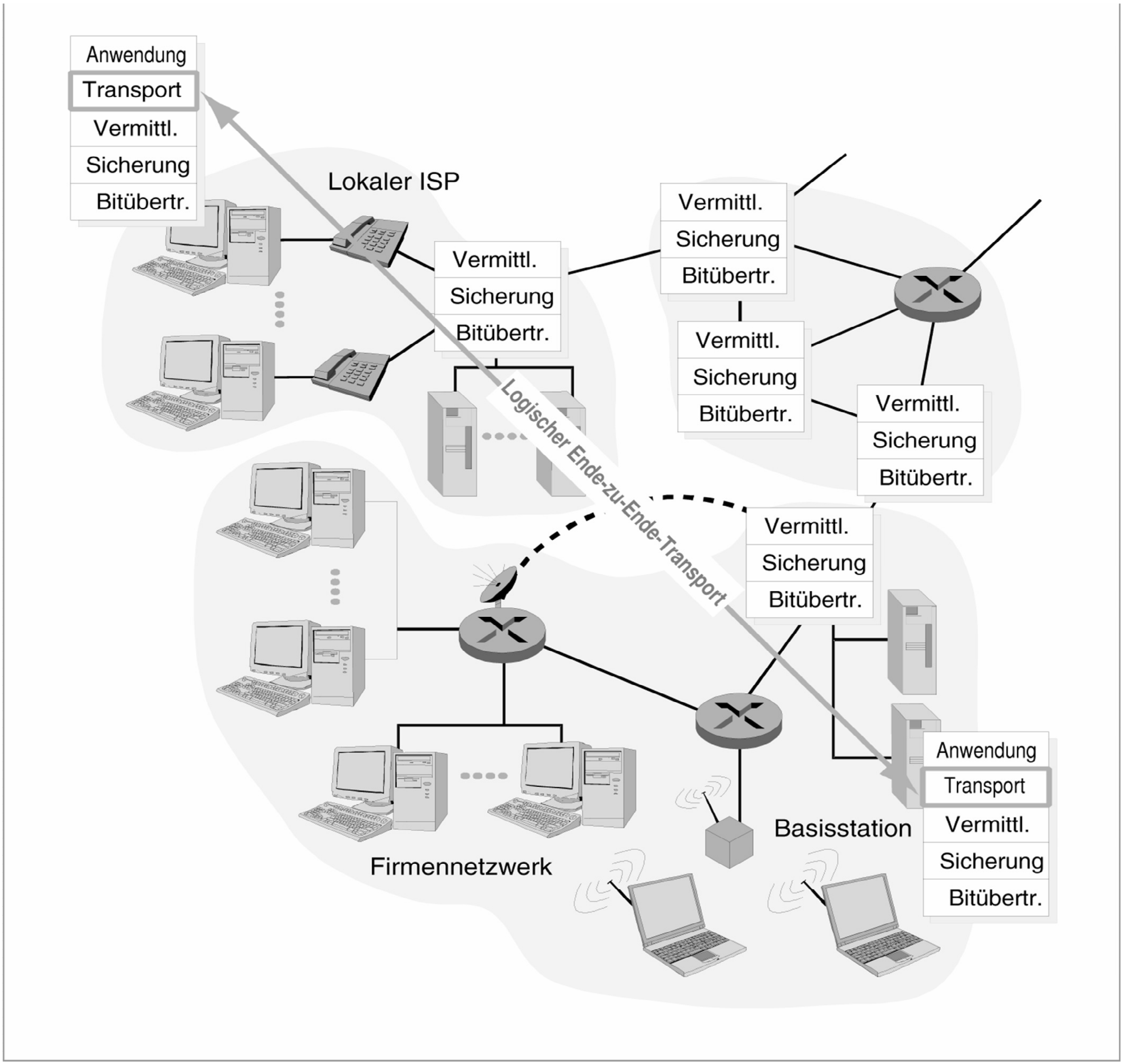
Kap. 2

Transport - Schicht


Transport protocols
Transport layer protocols
TCP
UDP
Relaiable stream service
Unreliable datagram service
All application data treated as a stream of bytes
Direct mapping between an application message unit and a UDP datagram
Connection-oriented
Error control
Flow control
Congestion control
Application layer support protocols
RTP
RTCP
Concerned with the transfer of real-time digital streams
Concerned with additional system-level issues relating to RTP

Transport-Schicht



Transport-Schicht: bietet eine logische Kommunikation zw. Anwendungen

TCP: - Verbindungsorientiert mittels 3-Way-Handshake
- zuverlässiger Datentransport mittels Bestätigung
- Multiplexen & Demultiplexen von Anwendungen mittels Port-Verwaltung

UDP: - Verbindungslos
- unzuverlässige Datenübertragung
- Multiplexen + Demultiplexen von Anwendungen mittels Port-Verwaltung

- Transport protocols for the TCP/IP protocol suite:
  - *Transmission Control Protocol (TCP)*
    - Provides a connection oriented service
    - Converts the best-effort service provided by IP into a reliable service
    - Example application: file transfers
      - The transmitted information should be free of errors
      - Messages are delivered in the sequence that they are submitted
      - Example application: FTP, HTTP

  - *User Datagram Protocol (UDP)*
    - Provides a connectionless (best-effort) service
    - Example application: SNMP
      - Single request-response message exchange

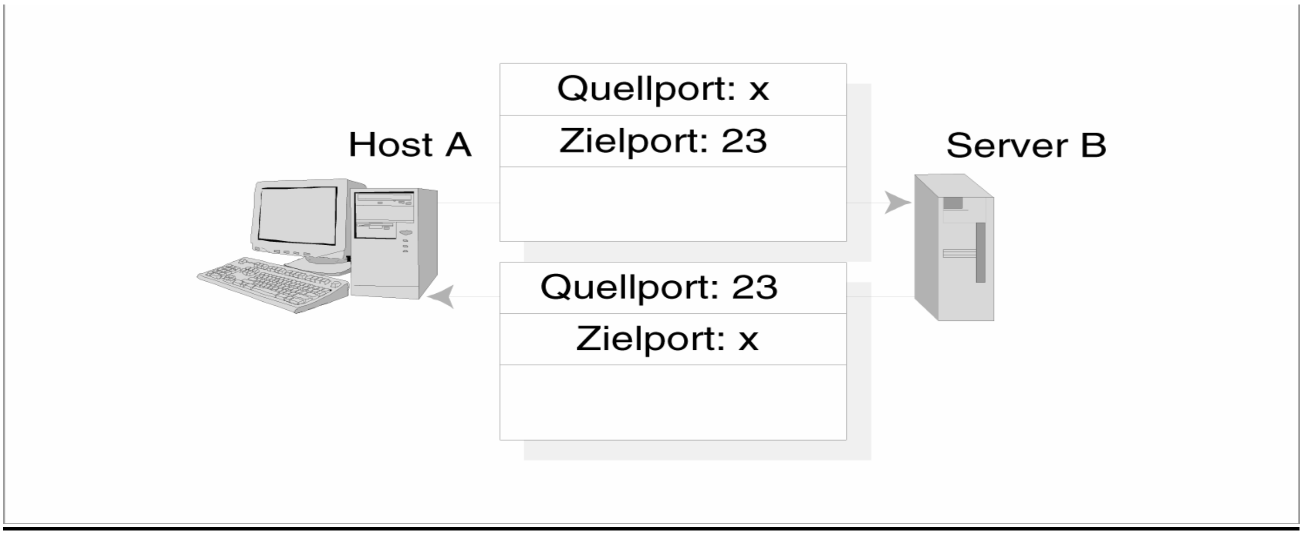
- Identification of the higher layer protocol or application in the PDU header
  - IP uses the *protocol* field to identify the protocol to which the contents of the datagram relate (e.g., TCP, UDP, ICMP)
  - The transport protocol (TCP or UDP) use the *port numbers* in the PDU header to identify the application protocol to which the PDU contents relate

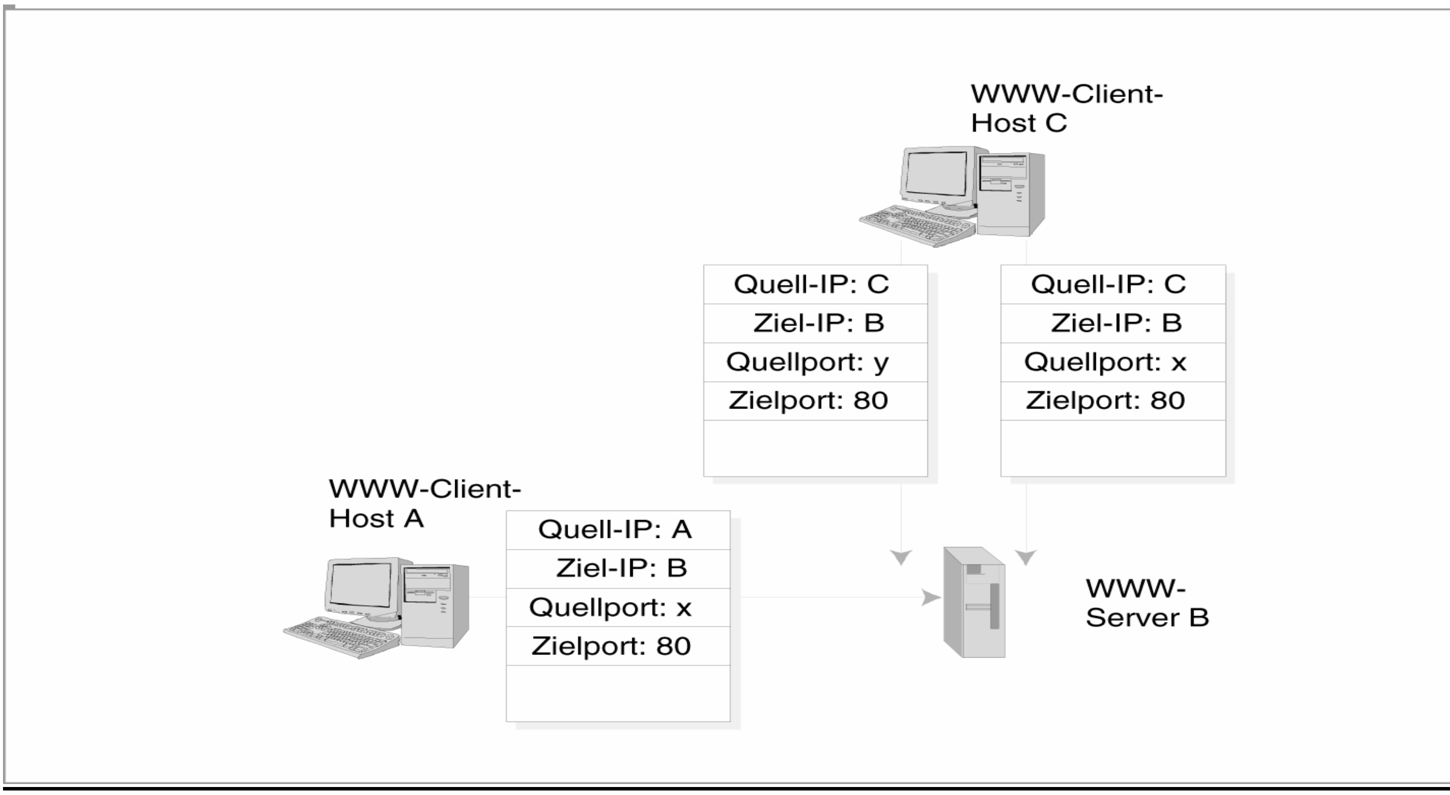
- Port numbers in a client-server communication
  - Client host
    - The port number of the source application protocol has only local significance
    - A new port number is allocated for each new transfer request
    - Client port numbers are called *ephemeral ports* since they are short-lived
    - Allocated in the range 1,024 through 5,000
  - Server
    - Port numbers are fixed: *well-known port numbers*
    - Allocated in the range 1 through 1,023
    - Example: file transfer (application) protocol port number is 21
    - Since a server application receives requests from multiple clients, both the source port number and the source IP address are sent to the application protocol

- Both the application and transport layers are end-to-end

- TCP provides two communicating peer application protocols (e.g., one in the client and one in the server) with a two-way, *reliable stream service*
    - Data is submitted by each local application to the TCP protocol entity as a stream of bytes
    - The stream of bytes flowing in each direction is transferred from one TCP entity to the other in a reliable way, i.e., no transmission errors, no lost or duplicate bytes

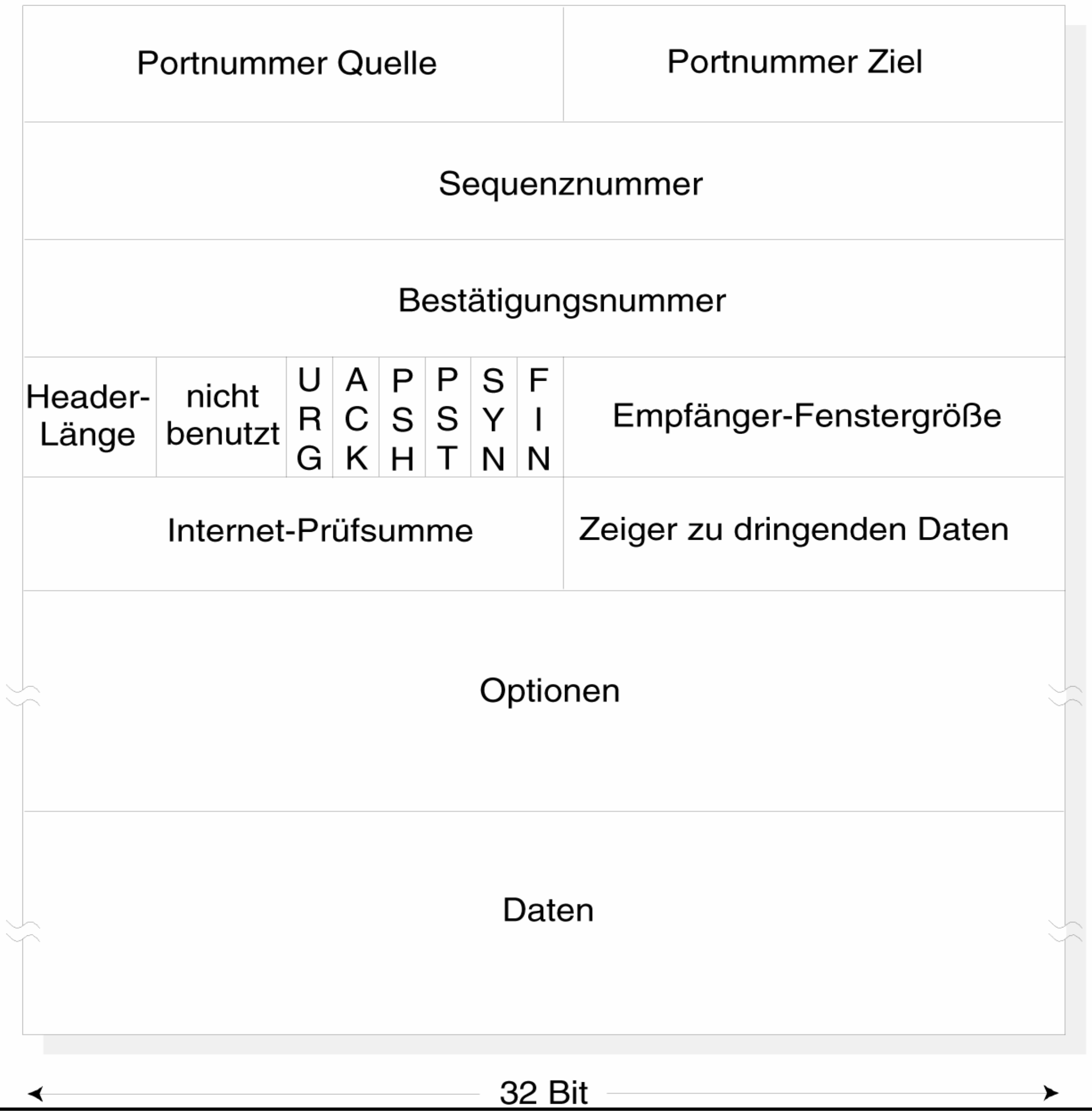
Verwendung von Ports in Client/Server Anwendungen





Benutzer „ A“ und „C“ : -> können die gleichen Portnr. verwenden um mit dem
Server zu kommunizieren

TCP-Meldungs-Format



Felder:

Port Nr. Ziel/Quelle : 64.000 Ports möglich

1 – 1024 reservierte Ports

z. B.: 20, 21, 25, 80 etc., ... „well-known Ports“

Sequ. Nr. + ACK Nr: Bestätigung: z.B.:

1) Verbindungs-Auf/Abbau:
SEQnr. = ACKnr. + 1

2) Datenübertragung
SEQnr. = ACKnr. + LÄNGE in Bytes

Header-Länge: Länge des TCP-Headers; Normalerweise = 20 Bytes

FLAGS:

- ***URG*** = Markiert eine Meldung als „dringend“

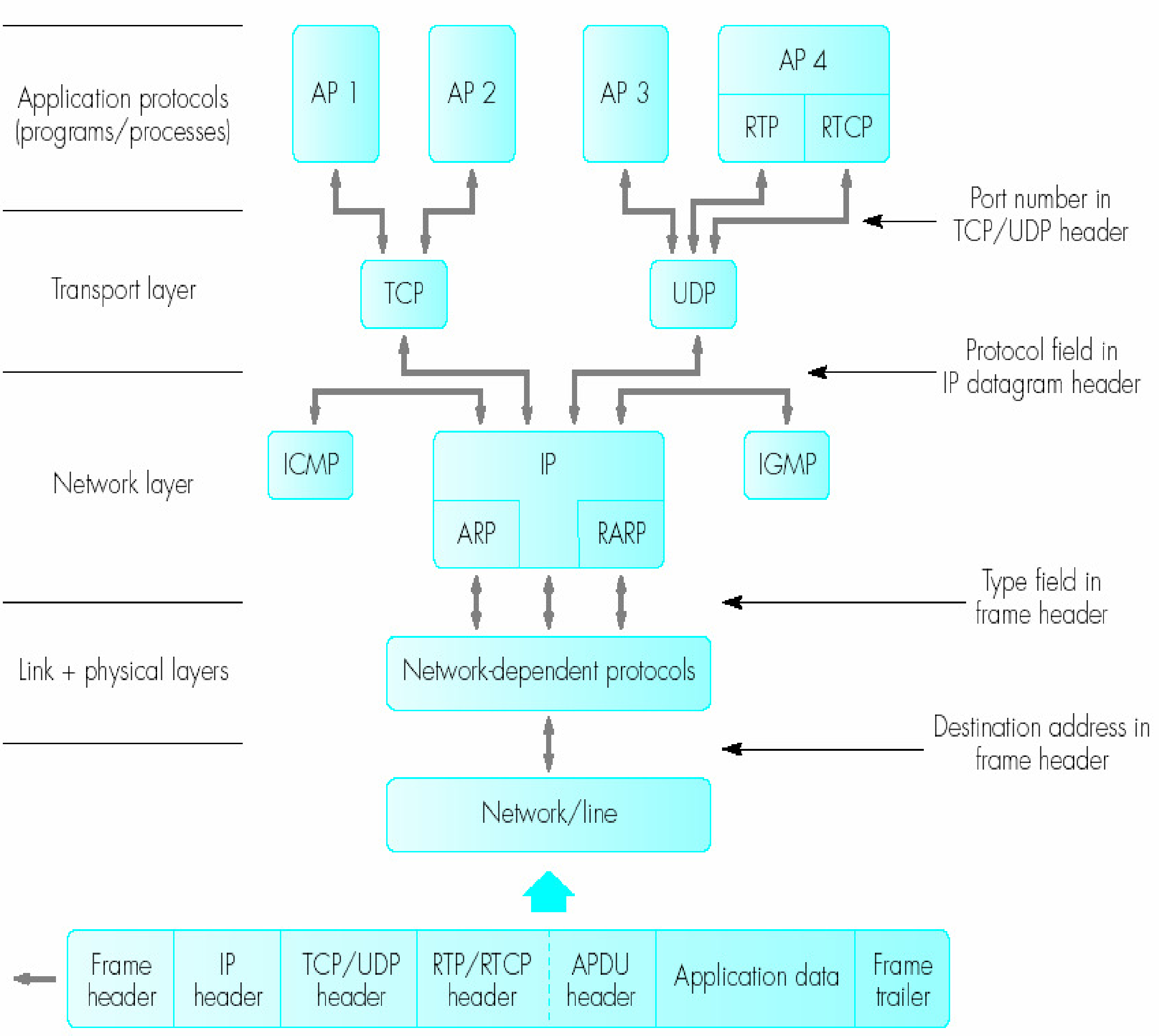
- ***PSH*** = Daten sollen sofort (auf Empfängerseite) weiter geleitet werden

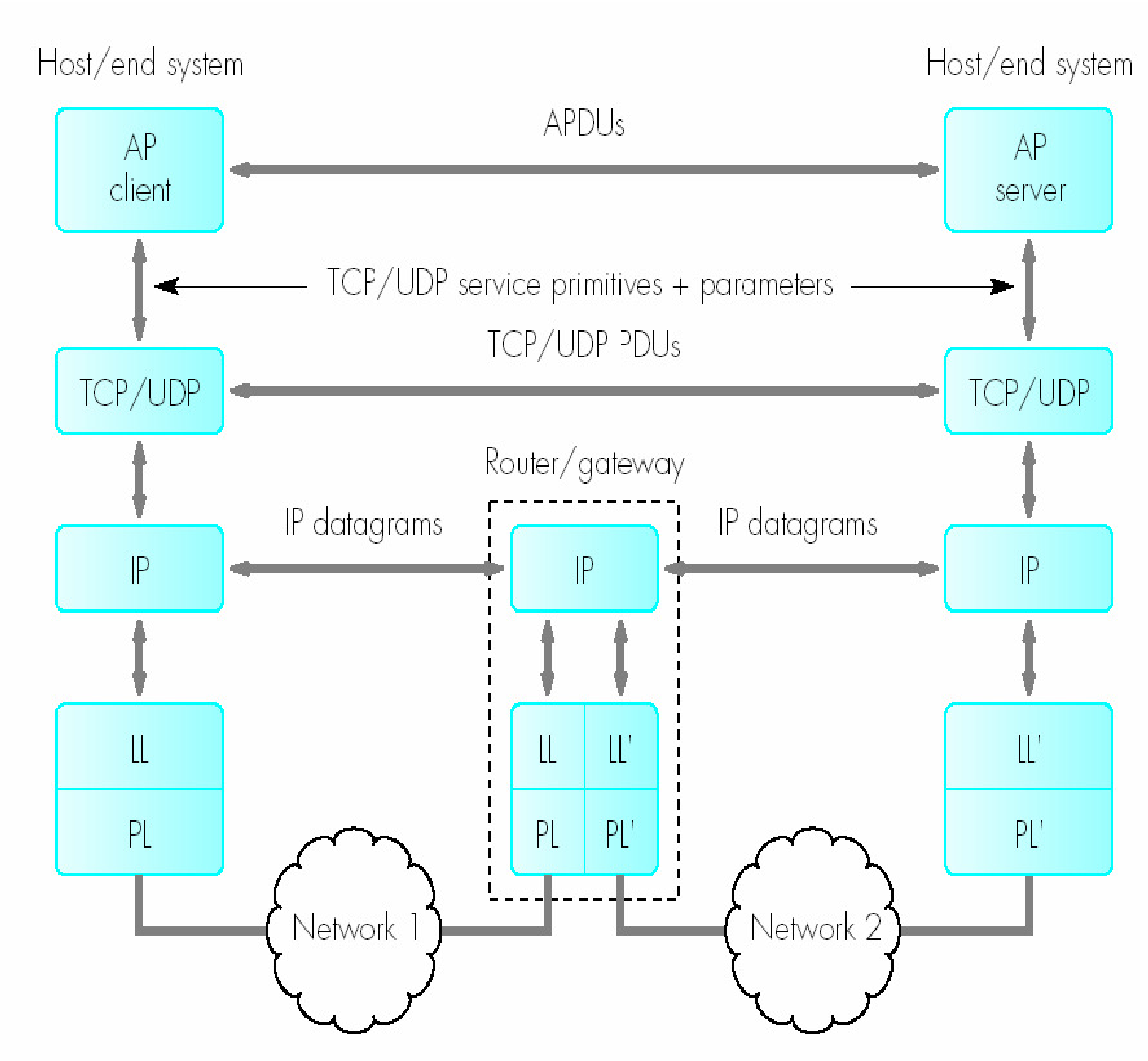
- ***RST, SYN, FIN*** = Verbindungs-Auf/Abbau

- ***ACK*** = Bestätigung

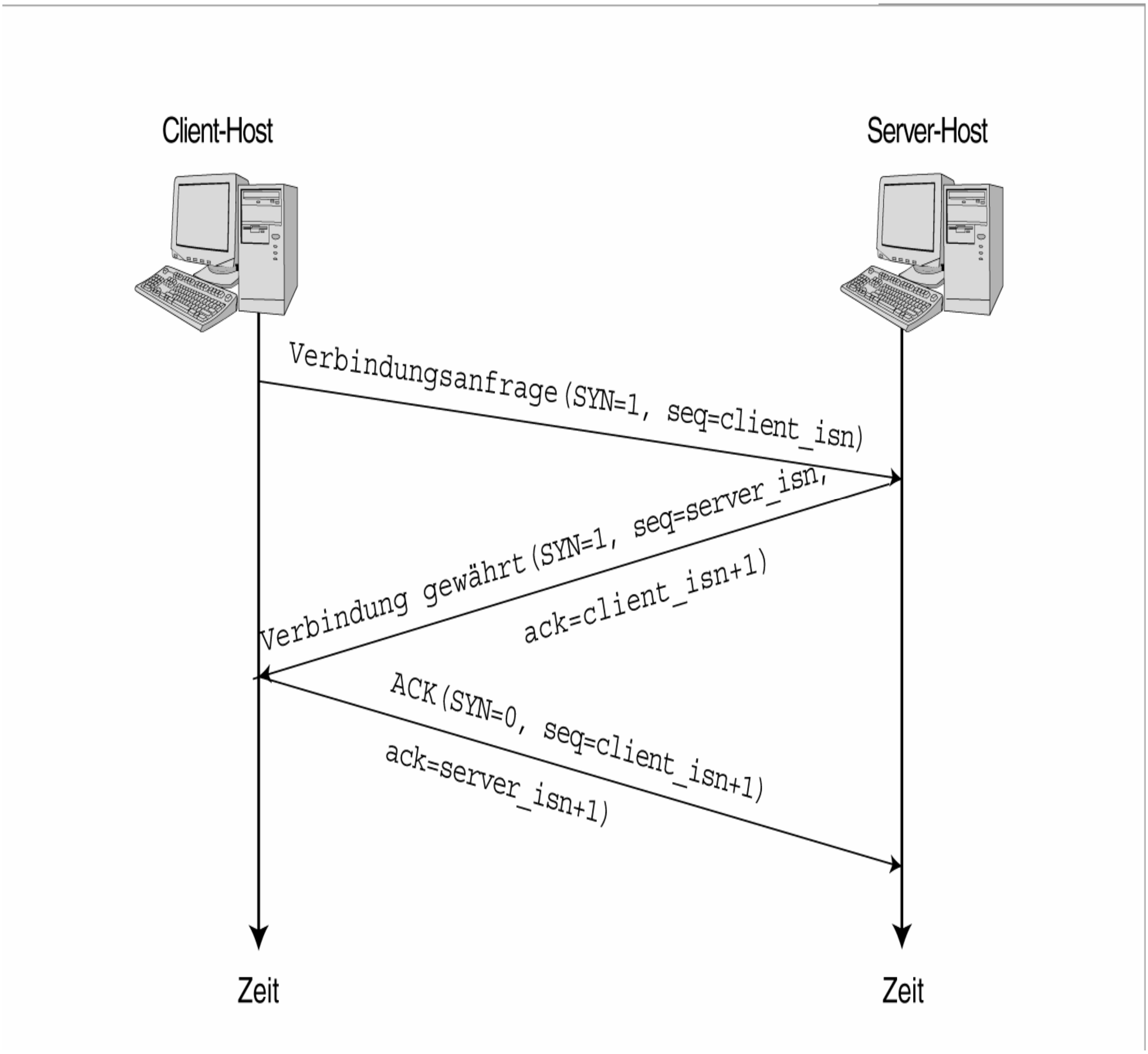
Fenstergröße:

- dient der Flusskontrolle
- # Bytes, die der Empfänger bereit ist, demnächst aufzunehmen


Application protocols (programs/processes)
AP 1
AP 2
AP 3
AP 4
RTP
RTCP
Port number in TCP/UDP header
Transport layer
TCP
UDP
Protocol field in IP datagram header
Network layer
ICMP
IP
ARP
RARP
IGMP
Type field in frame header
Link + physical layers
Network-dependent protocols
Destination address in frame header
Network/line
Frame header
IP header
TCP/UDP header
RTP/RTCP header
APDU header
Application data
Frame trailer

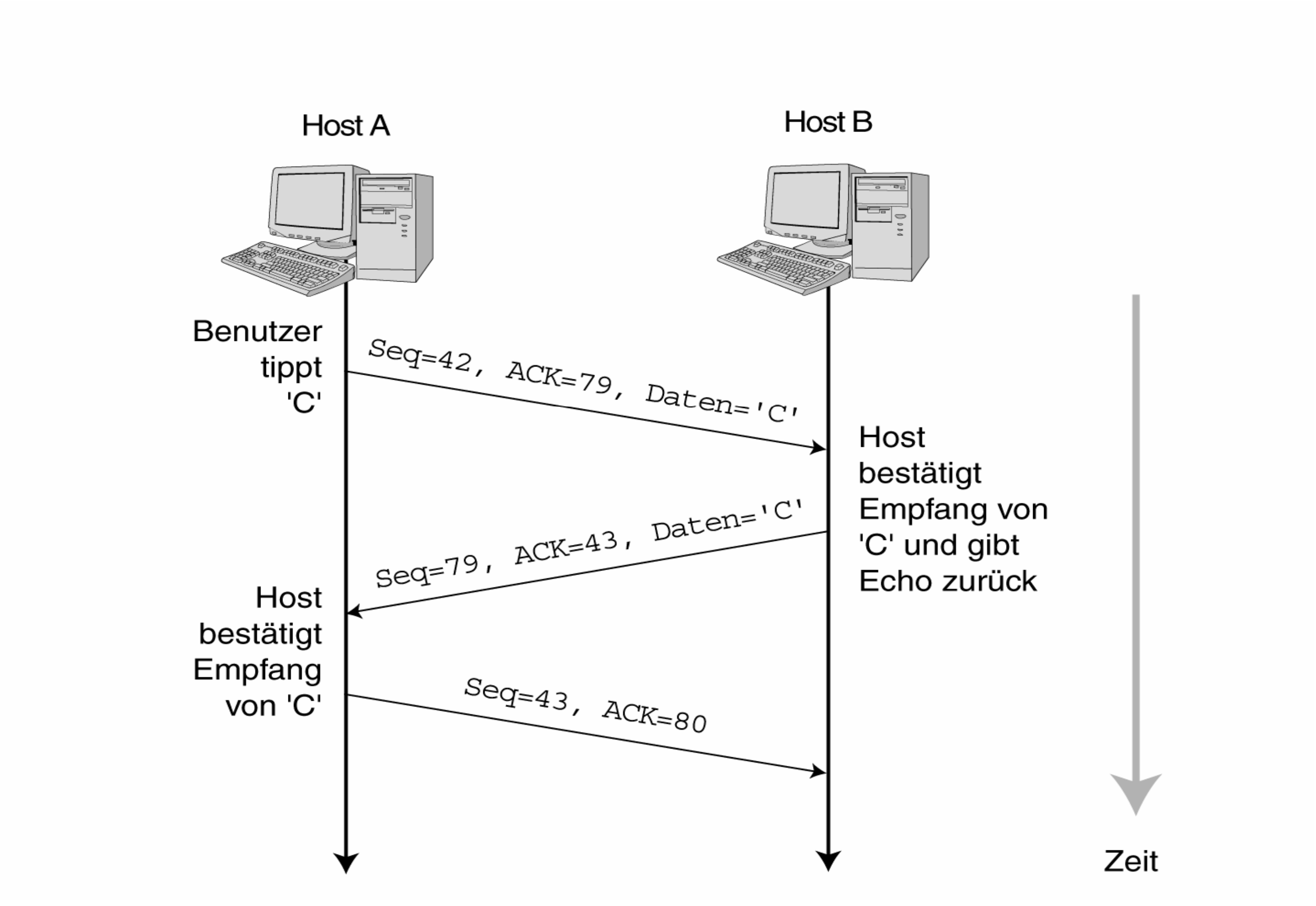

Host/end system
Host/end system
APDUs
AP
client
AP
server
TCP/UDP service primitives + parameters
TCP/UDP PDUs
TCP/UDP
TCP/UDP
Router/gateway
IP datagrams
IP datagrams
IP
IP
IP
LL
PL
LL
LL'
PL
PL'
LL'
PL'
Network 1
Network 2

TCP-Protokollablauf – Szenarien (1)


Client-Host
Server-Host
Verbindungsanfrage (SYN=1, seq=client_isn)
Verbindung gewährt (SYN=1, seq=server_isn,
ack=client_isn+1)
ACK (SYN=0, seq=client_isn+1)
ack=server_isn+1)
Zeit
Zeit

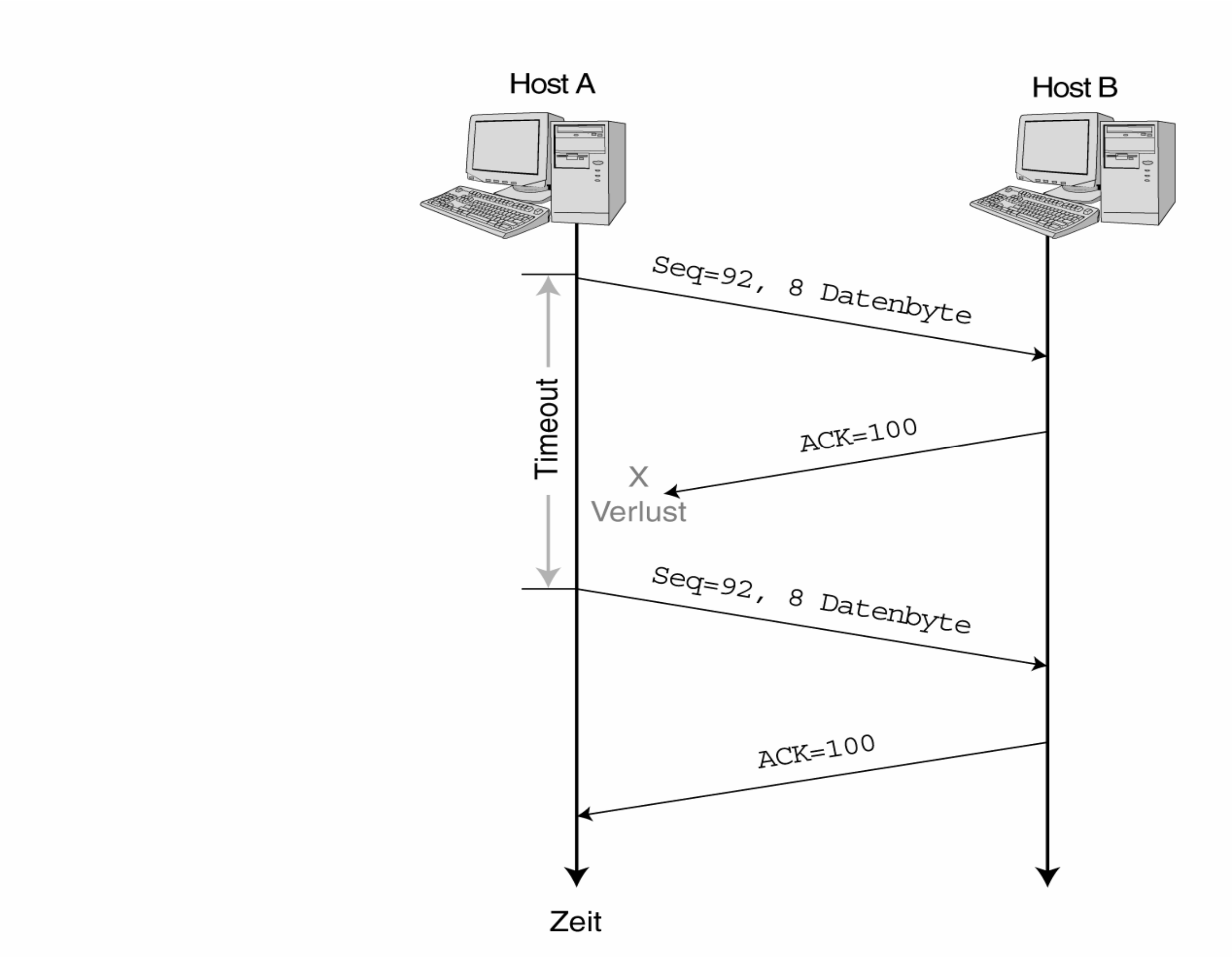
Verbindungsaufbau: 3 x Way - Handashake

TCP-Protokollablauf-Szenarien (2)


Host A
Host B
Benutzer tippt 'C'
Seq=42, ACK=79, Daten='C'
Host bestätigt Empfang von 'C' und gibt Echo zurück
Seq=79, ACK=43, Daten='C'
Host bestätigt Empfang von 'C'
Seq=43, ACK=80
Zeit

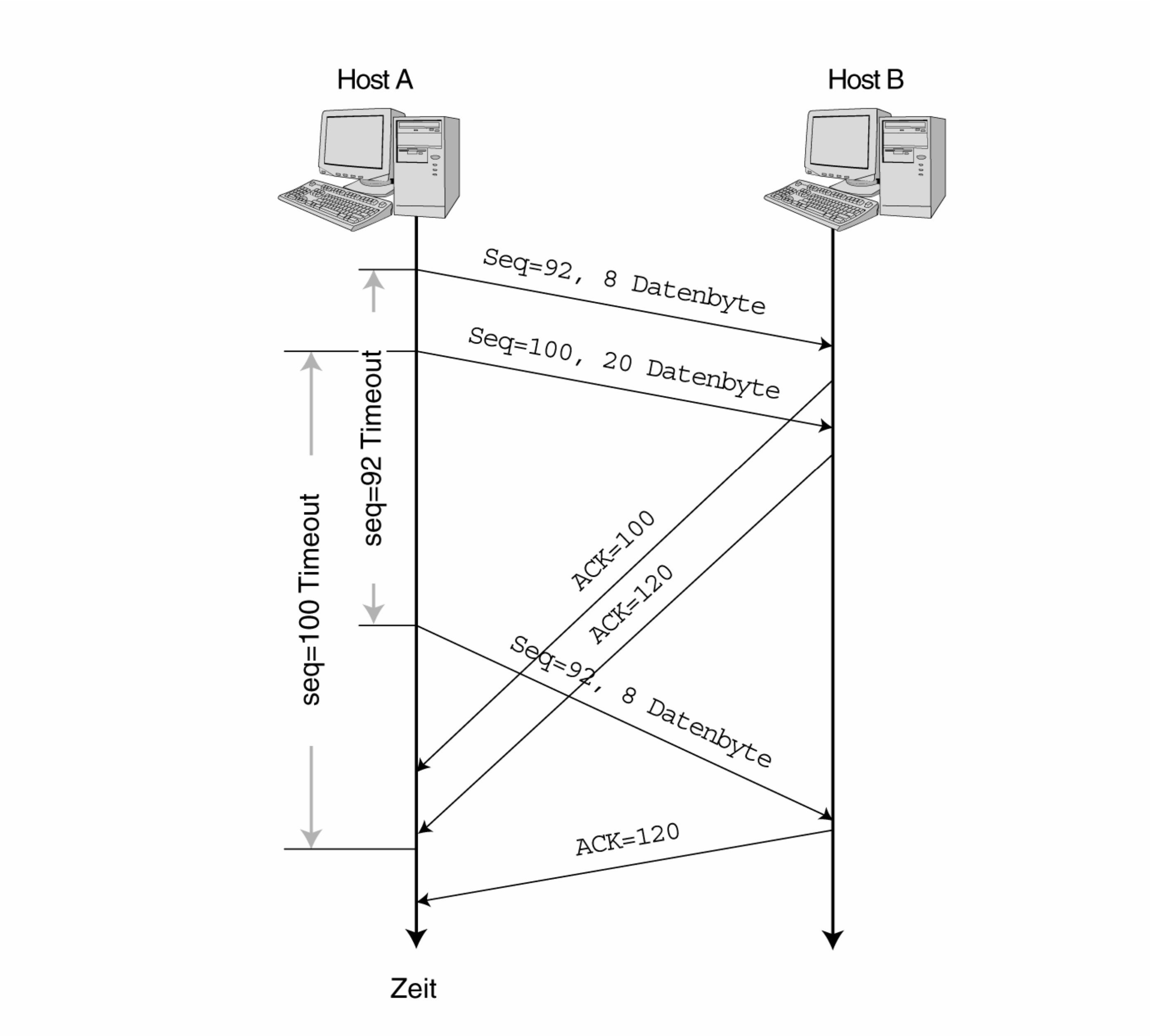
Datenübertragungs - Ablauf: SEQnr.= ACKnr. + # Bytes (Länge**)**

TCP-Protokollablauf-Szenarien (3)



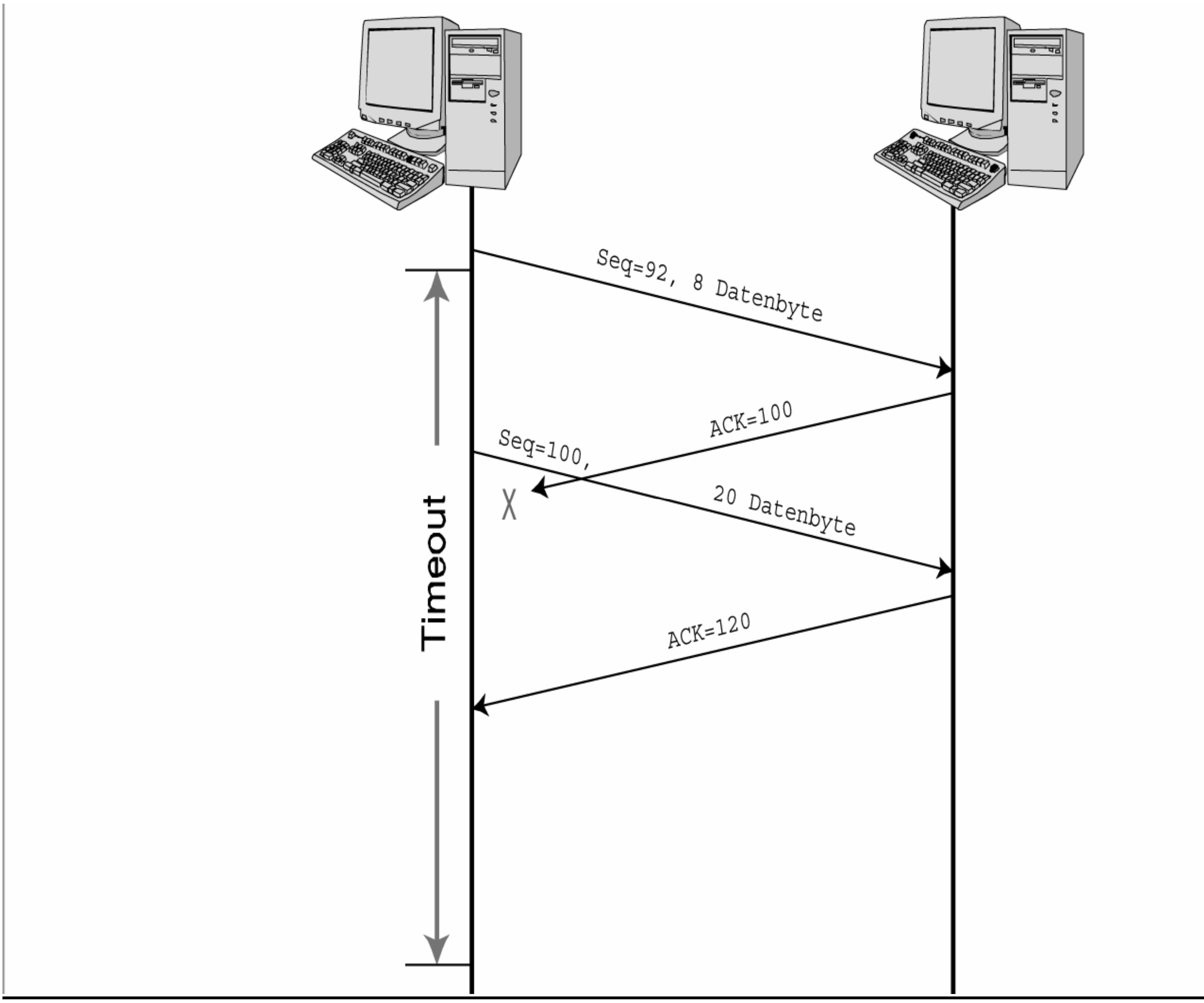
Fehlerkorrektur: Neuübertargung als Folge eines Fehlers (verlorene Bestätigung)

TCP-Protokollablauf-Szenarien (4)



Fehlerbehandlung: Datenblock wird nicht erneut übertragen weil seine ACK vor dem Timeout ankommt

TCP-Protokollablauf-Szenarien (4)



GO-BACK TO „N" –Prinzip: Durch eine kumulative Bestätigung wird die Neuübertragung vermieden

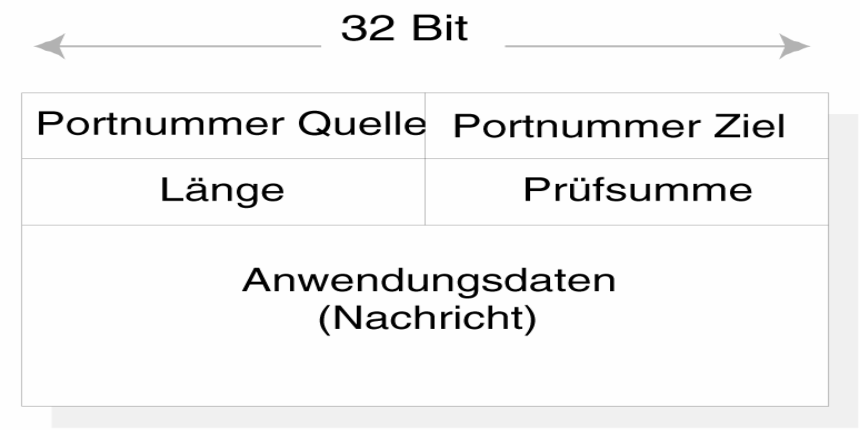
TCP operation

- A logical connection is first established between the two TCP entities and the sequence numbers are initialized
- During the data transfer, each TCP entity divides the submitted stream of bytes into *segments*
    - A segment may contain a single byte (e.g., a character from a terminal) or many bytes (e.g., file transfer)
    - A *maximum segment size (MSS)* is agreed when the connection is set up in order to
        - Minimize transmission errors
        - Avoid fragmentation
    - Default MSS is 536 bytes
- *Flow control* is used to ensure that no data is lost when a fast server is sending data to a slower host
- *Congestion control* is used to adjust the rate of entry of segments into the network to the rate at which segments are leaving

Connection record:

- The TCP in the client may support multiple concurrent connections involving different user APs
- The TCP in the server may support multiple connections for different clients
- Data structure created for each new connection to associate received segments with the correct connection
- Contents: *Connection identifier* (pair of socket addresses), agreed *MSS*, the *initial sequence number* for each direction, the *precedence value*, and *size of window*

UDP-Protokoll: Nachrichten-Format und Funktionsweise



32 Bit

Portnummer Quelle	Portnummer Ziel
Länge	Prüfsumme
Anwendungsdaten (Nachricht)	

Port Nr. Quelle / Ziel : ähnlich wie beim TCP
Länge : Paketlänge
Prüfsumme : 1er Kompl. der Summer aller 16 Bit Wörter im Segment

Dienste von UDP: bekannt auch als Datagramm-Protokoll

- ***kein Verbindungsaufbau : keine Verzögerungen***
- ***kein Verbindungszustand***
- ***keine Bestätigung***, d. h. keine Nummerierung der Meldungen
- ***geringer Overhead*** durch Packet-Header: 8 Bytes
- Unregulierte Sende-Rate: d.h. ***keine Überlast-Kontrolle***
- Anwendungen basierend auf UDP:
  - DNS, SNMP, RIP, NF, Internet-Phone, Audio/Video-Streaming

Principles of operation

- Each message submitted by the user AP is transferred directly in a single IP datagram
- There is no connection set up, no error or flow control
- The service provided to a user AP is an extension of the service provided by IP
- The service primitives and the protocol are simpler than those of TCP
- The source UDP
  - Adds a short header to the message received from the user AP to form a *UDP datagram*
  - Forwards the UDP datagram to the IP layer for transfer over the internet using, if necessary, fragmentation
- The destination IP and UDP

- IP determines from the *protocol* field in the header that the destination protocol is UDP and passes IP datagram content to the UDP
- UDP determines the destination user AP from a field in UDP datagram header and transfers contents of datagram to that AP
- Maximum size of a UDP datagram:
  - Maximum theoretical size is 65,507 (65,535 for IP datagram – 20 bytes in IP header – 8 bytes in UDP header)
  - However, most implementations support only 8,192 bytes
  - If fragmentation is to be avoided, the size of each application PDU should be limited to the MTU for that path minus the IP and UDP headers

```
     Sniffer Network Analyzer data from 11-Oct-102 at 09:30:48, unsaved capture data,
    Page 1
- - - - - - - - - - - - - - - - Frame 5 - - - - - - - - - - - - - - - - -
  SUMMARY:
     Delta T    Destination   Source             Summary
  5   0.0031  gigaserv.uni-.. [194.95.109.136)  DLC Ethertype=0800, size=62 bytes
                                                 IP  D=[131.234.25.10] S=[194.95.109.136]
                                                      LEN=28 ID=87
                                                TCP D=21 S=1036 SYN SEQ=2392861514
                                                      LEN=0  WIN=16384

DLC:  ----- DLC Header -----
DLC:
DLC:  Frame 5 arrived at  09:30:52.5536; frame size is 62 (003E hex) bytes.
DLC:  Destination = Station 00A08E30D27F
DLC:  Source      = Station 00065B75C343, RFHPCI136
DLC:  Ethertype  = 0800 (IP)
DLC:
IP:   ----- IP Header -----
IP:
IP:   Version = 4, header length = 20 bytes
IP:   Type of service = 00
IP:         000. .... = routine
IP:         ...0 .... = normal delay
IP:         .... 0... = normal throughput
IP:         .... .0.. = normal reliability
IP:   Total length    = 48 bytes
IP:   Identification  = 87
IP:   Flags           = 4X
IP:         .1.. .... = don't fragment
IP:         ..0. .... = last fragment
IP:   Fragment offset = 0 bytes
IP:   Time to live    = 128 seconds/hops
IP:   Protocol        = 6 (TCP)
IP:   Header checksum = 2D95 (correct)
IP:   Source address      = [194.95.109.136]
IP:   Destination address = [131.234.25.10], gigaserv.uni-paderborn.de
IP:   No options
IP:
TCP:  ----- TCP header -----
TCP:
TCP:  Source port             = 1036
TCP:  Destination port        = 21 (FTP)
TCP:  Initial sequence number = 2392861514
TCP:  Data offset             = 28 bytes
TCP:  Flags                   = 02
TCP:                ..0. .... = (No urgent pointer)
TCP:                ...0 .... = (No acknowledgment)
TCP:                .... 0... = (No push)
TCP:                .... .0.. = (No reset)
TCP:                .... ..1. = SYN
```

```
TCP:              .... ...0 = (No FIN)
TCP:  Window              = 16384
TCP:  Checksum             = B837 (correct)
TCP:
TCP:  Options follow
TCP:  Maximum segment size    = 1460
TCP:  No-op
TCP:  No-op
TCP:  Unknown option 4
TCP:  1 byte(s) of header padding
TCP:
```

Sniffer Network Analyzer data from 11-Oct-102 at 09:30:48, unsaved capture data, Page 2

```
- - - - - - - - - - - - - - - - Frame 6 - - - - - - - - - - - - - - - - -

      Delta T  Destination       Source      Summary
  6   0.0243  [194.95.109.136   gigaserv.uni-..  DLC Ethertype=0800, size=62 bytes
                                   IP  D=[194.95.109.136] S=[131.234.25.10] LEN=28
                                        ID=37079
                                  TCP D=1036 S=21 SYN ACK=2392861515
                                        SEQ=3168076761 LEN=0 WIN=8760

DLC:  ----- DLC Header -----
DLC:
DLC:  Frame 6 arrived at  09:30:52.5779; frame size is 62 (003E hex) bytes.
DLC:  Destination = Station 00065B75C343, RFHPCI136
DLC:  Source      = Station 00A08E30D27F
DLC:  Ethertype  = 0800 (IP)
DLC:
IP:   ----- IP Header -----
IP:
IP:   Version = 4, header length = 20 bytes
IP:   Type of service = 00
IP:         000. .... = routine
IP:         ...0 .... = normal delay
IP:         .... 0... = normal throughput
IP:         .... .0.. = normal reliability
IP:   Total length    = 48 bytes
IP:   Identification  = 37079
IP:   Flags           = 4X
IP:         .1.. .... = don't fragment
IP:         ..0. .... = last fragment
IP:   Fragment offset = 0 bytes
IP:   Time to live    = 246 seconds/hops
IP:   Protocol        = 6 (TCP)
IP:   Header checksum = 2714 (correct)
IP:   Source address      = [131.234.25.10], gigaserv.uni-paderborn.de
IP:   Destination address = [194.95.109.136]
```

```
IP:   No options
IP:
TCP:  ----- TCP header -----
TCP:
TCP:  Source port          = 21 (FTP)
TCP:  Destination port       = 1036
TCP:  Initial sequence number = 3168076761
TCP:  Acknowledgment number   = 2392861515
TCP:  Data offset          = 28 bytes
TCP:  Flags              = 12
TCP:              ..0. .... = (No urgent pointer)
TCP:              ...1 .... = Acknowledgment
TCP:              .... 0... = (No push)
TCP:              .... .0.. = (No reset)
TCP:              .... ..1. = SYN
TCP:              .... ...0 = (No FIN)
TCP:  Window               = 8760
TCP:  Checksum              = 1540 (correct)
TCP:
TCP:  Options follow
TCP:  No-op
TCP:  No-op
TCP:  Unknown option 4
TCP:  5 byte(s) of header padding
TCP:

Sniffer Network Analyzer data from 11-Oct-102 at 09:30:48, unsaved capture data, Page 3

- - - - - - - - - - - - - - - - Frame 7 - - - - - - - - - - - - - - - - -

      Delta T   Destination    Source           Summary
   7  0.0001  gigaserv.uni-..  [194.95.109.136  DLC Ethertype=0800, size=60 bytes
                                                IP  D=[131.234.25.10] S=[194.95.109.136]
                                                    LEN=20 ID=88
                                                TCP D=21 S=1036     ACK=3168076762
                                                      WIN=17520

DLC:  ----- DLC Header -----
DLC:
DLC:  Frame 7 arrived at  09:30:52.5780; frame size is 60 (003C hex) bytes.
DLC:  Destination = Station 00A08E30D27F
DLC:  Source      = Station 00065B75C343, RFHPCI136
DLC:  Ethertype  = 0800 (IP)
DLC:
IP:   ----- IP Header -----
IP:
IP:   Version = 4, header length = 20 bytes
IP:   Type of service = 00
IP:         000. .... = routine
IP:         ...0 .... = normal delay
IP:         .... 0... = normal throughput
```

```
IP:         .... .0.. = normal reliability
IP:   Total length    = 40 bytes
IP:   Identification  = 88
IP:   Flags           = 4X
IP:         .1.. .... = don't fragment
IP:         ..0. .... = last fragment
IP:   Fragment offset = 0 bytes
IP:   Time to live    = 128 seconds/hops
IP:   Protocol        = 6 (TCP)
IP:   Header checksum = 2D9C (correct)
IP:   Source address      = [194.95.109.136]
IP:   Destination address = [131.234.25.10], gigaserv.uni-paderborn.de
IP:   No options
IP:
TCP:  ----- TCP header -----
TCP:
TCP:  Source port            = 1036
TCP:  Destination port       = 21 (FTP)
TCP:  Sequence number        = 2392861515
TCP:  Acknowledgment number  = 3168076762
TCP:  Data offset            = 20 bytes
TCP:  Flags                  = 10
TCP:              ..0. .... = (No urgent pointer)
TCP:              ...1 .... = Acknowledgment
TCP:              .... 0... = (No push)
TCP:              .... .0.. = (No reset)
TCP:              .... ..0. = (No SYN)
TCP:              .... ...0 = (No FIN)
TCP:  Window                 = 17520
TCP:  Checksum               = 1FCC (correct)
TCP:  No TCP options
TCP:
```

Sniffer Network Analyzer data from 11-Oct-102 at 09:30:48, unsaved capture data, Page 1

```
- - - - - - - - - - - - - - - - Frame 48 - - - - - - - - - - - - - - - - -

        Delta T    Destination   Source          Summary
   48   0.0012  gigaserv.uni-.. [194.95.109.1..  DLC Ethertype=0800, size=60 bytes
                                  IP  D=[131.234.25.10] S=[194.95.109.136] LEN=20 ID=109
                                  TCP D=21 S=1036 FIN ACK=3168079573
                                       SEQ=2392861579 LEN=0 WIN=16822

DLC:  ----- DLC Header -----
DLC:
DLC:  Frame 48 arrived at  09:31:02.9334; frame size is 60 (003C hex) bytes.
DLC:  Destination = Station 00A08E30D27F
DLC:  Source      = Station 00065B75C343, RFHPCI136
DLC:  Ethertype  = 0800 (IP)
DLC:
IP:   ----- IP Header -----
IP:
IP:   Version = 4, header length = 20 bytes
IP:   Type of service = 00
IP:         000. .... = routine
IP:         ...0 .... = normal delay
IP:         .... 0... = normal throughput
IP:         .... .0.. = normal reliability
IP:   Total length    = 40 bytes
IP:   Identification  = 109
IP:   Flags           = 4X
IP:         .1.. .... = don't fragment
IP:         ..0. .... = last fragment
IP:   Fragment offset = 0 bytes
IP:   Time to live    = 128 seconds/hops
IP:   Protocol        = 6 (TCP)
IP:   Header checksum = 2D87 (correct)
IP:   Source address      = [194.95.109.136]
IP:   Destination address = [131.234.25.10], gigaserv.uni-paderborn.de
IP:   No options
IP:
TCP:  ----- TCP header -----
TCP:
TCP:  Source port            = 1036
TCP:  Destination port       = 21 (FTP)
TCP:  Sequence number        = 2392861579
TCP:  Acknowledgment number   = 3168079573
TCP:  Data offset            = 20 bytes
```

```
TCP:  Flags              = 11
TCP:            ..0. .... = (No urgent pointer)
TCP:            ...1 .... = Acknowledgment
TCP:            .... 0... = (No push)
TCP:            .... .0.. = (No reset)
TCP:            .... ..0. = (No SYN)
TCP:            .... ...1 = FIN
TCP:  Window              = 16822
TCP:  Checksum            = 174A (correct)
TCP:  No TCP options
TCP:
```

Sniffer Network Analyzer data from 11-Oct-102 at 09:30:48, unsaved capture data, Page 2

```
- - - - - - - - - - - - - - - - Frame 49 - - - - - - - - - - - - - - - - -

      Delta T  Destination      Source          Summary
  49  0.0066  [194.95.109.1.. gigaserv.uni-..  DLC Ethertype=0800, size=60 bytes
                                               IP  D=[194.95.109.136] S=[131.234.25.10]
                                                   LEN=20 ID=37099
                                               TCP D=1036 S=21 FIN ACK=2392861579
                                                     SEQ=3168079573 LEN=0 WIN=8760

DLC:  ----- DLC Header -----
DLC:
DLC:  Frame 49 arrived at  09:31:02.9401; frame size is 60 (003C hex) bytes.
DLC:  Destination = Station 00065B75C343, RFHPCI136
DLC:  Source      = Station 00A08E30D27F
DLC:  Ethertype  = 0800 (IP)
DLC:
IP:   ----- IP Header -----
IP:
IP:   Version = 4, header length = 20 bytes
IP:   Type of service = 10
IP:         000. .... = routine
IP:         ...1 .... = low delay
IP:         .... 0... = normal throughput
IP:         .... .0.. = normal reliability
IP:   Total length    = 40 bytes
IP:   Identification  = 37099
IP:   Flags           = 4X
IP:         .1.. .... = don't fragment
IP:         ..0. .... = last fragment
IP:   Fragment offset = 0 bytes
IP:   Time to live    = 246 seconds/hops
IP:   Protocol        = 6 (TCP)
IP:   Header checksum = 26F8 (correct)
IP:   Source address      = [131.234.25.10], gigaserv.uni-paderborn.de
IP:   Destination address = [194.95.109.136]
IP:   No options
IP:
```

```
TCP:  ----- TCP header -----
TCP:
TCP:  Source port             = 21 (FTP)
TCP:  Destination port        = 1036
TCP:  Sequence number         = 3168079573
TCP:  Acknowledgment number   = 2392861579
TCP:  Data offset             = 20 bytes
TCP:  Flags                   = 11
TCP:              ..0. .... = (No urgent pointer)
TCP:              ...1 .... = Acknowledgment
TCP:              .... 0... = (No push)
TCP:              .... .0.. = (No reset)
TCP:              .... ..0. = (No SYN)
TCP:              .... ...1 = FIN
TCP:  Window                  = 8760
TCP:  Checksum                = 36C8 (correct)
TCP:  No TCP options
TCP:
```