

KAPITEL 4:

TRANSPORTSCHICHT: TCP/UDP

Lerninhalte:

- Verständnis prinzipieller Eigenschaften von Diensten der Transportschicht:
 - Multiplexing/ Demultiplexing
 - Segmentierung
 - Zuverlässiger Transport von Daten
 - Vermeidung/Verringerung von Paketverlusten durch Überlast im Netz
- Praktische Realisierung
 - UDP: verbindungslose Übertragung von Datagrammen
 - TCP: gesicherte Datenübertragung über TCP-Verbindungen
 - verschiedene TCP Varianten

4.1 Multiplexing

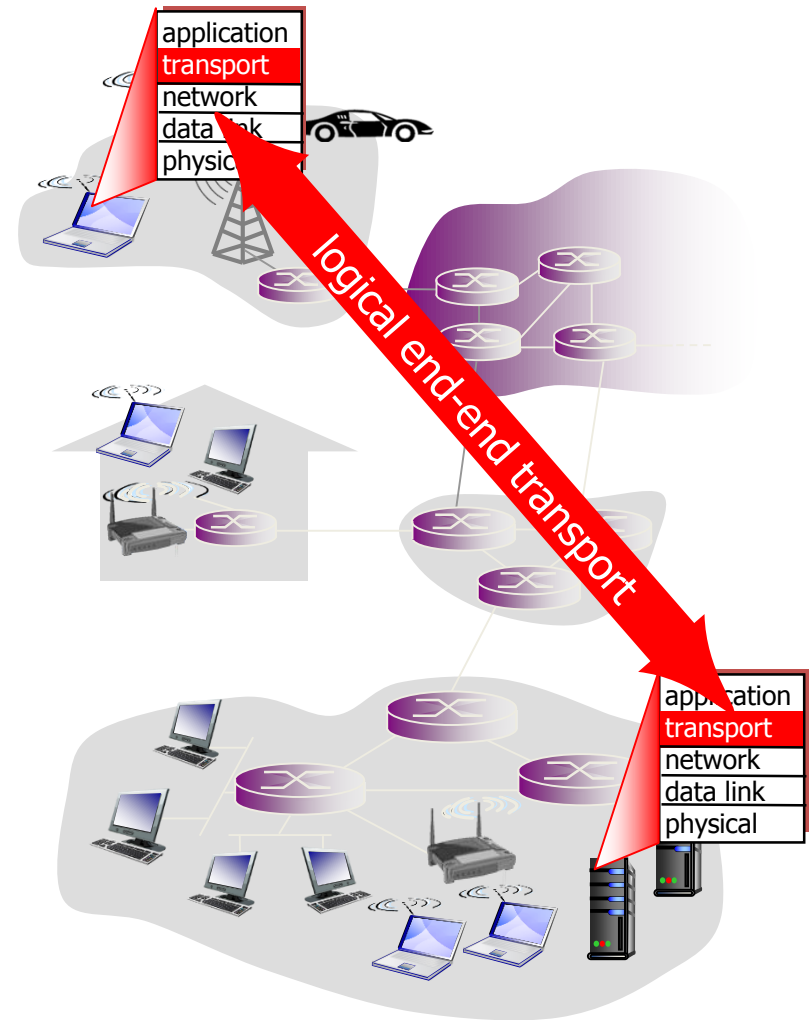
4.2 UDP

4.3 TCP

4.4 Zusammenfassung

Dienste und Protokolle der Transportschicht

- Bieten logische Kommunikation zwischen Anwendungsprozessen auf verschiedenen Hosts
- Transportprotokolle laufen auf Endsystemen und nicht im Netz
 - Sender: teilt Nachrichten in Segmente, gibt Segmente an Netzwerkschicht weiter
 - Empfänger: setzt Segmente von der Netzwerkschicht wieder zu Nachrichten zusammen
- Protokolle der Transportschicht
 - Internet: TCP und UDP
 - weitere:
 - SCTP (Stream Control Transmission Protocol)



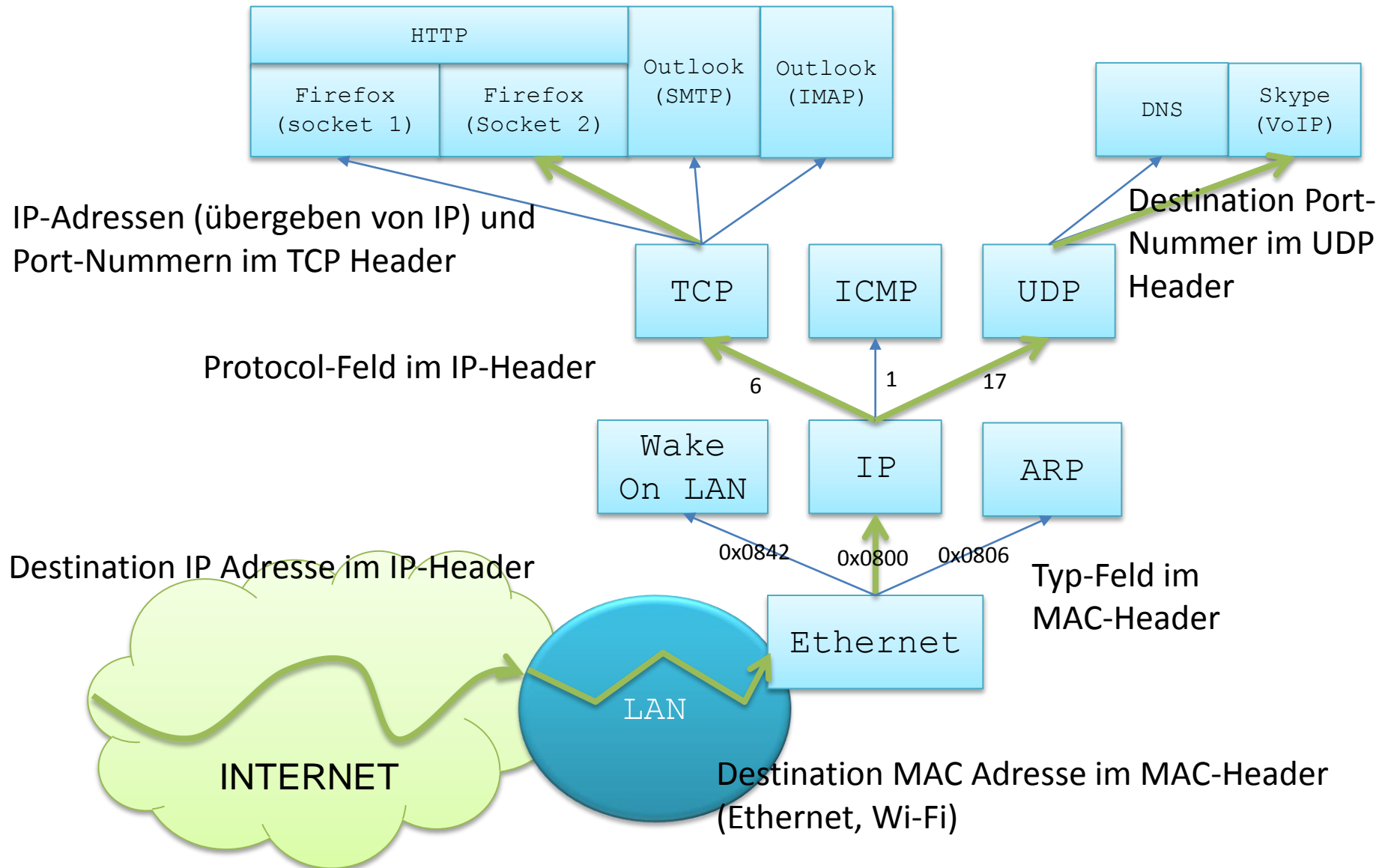
TCP-Dienste:

- **Zuverlässiger** Transport von Daten zwischen Sender und Empfänger
 - alle Daten ohne Verlust in gleicher Reihenfolge
- **Flusskontrolle:** Sender überflutet Empfänger nicht mit Daten
- **Überlastkontrolle:** Drosseln des Senders bei Überlast im Netz
- **Keinerlei Garantien** für Qualität der Übertragung (Dauer, Durchsatz, Sicherheit)
- **Verbindungsorientierung:** Herstellen einer Verbindung zwischen Client und Server

UDP-Dienste:

- **Unzuverlässiger** Transport von Daten zwischen Sender und Empfänger
 - Verlust und Änderung der Reihenfolge möglich
- **Keine** Verbindungsorientierung, Zuverlässigkeit, Flusskontrolle, Überlastkontrolle, Garantien für Qualität der Übertragung

Wo steht das Ziel in einem Paket?

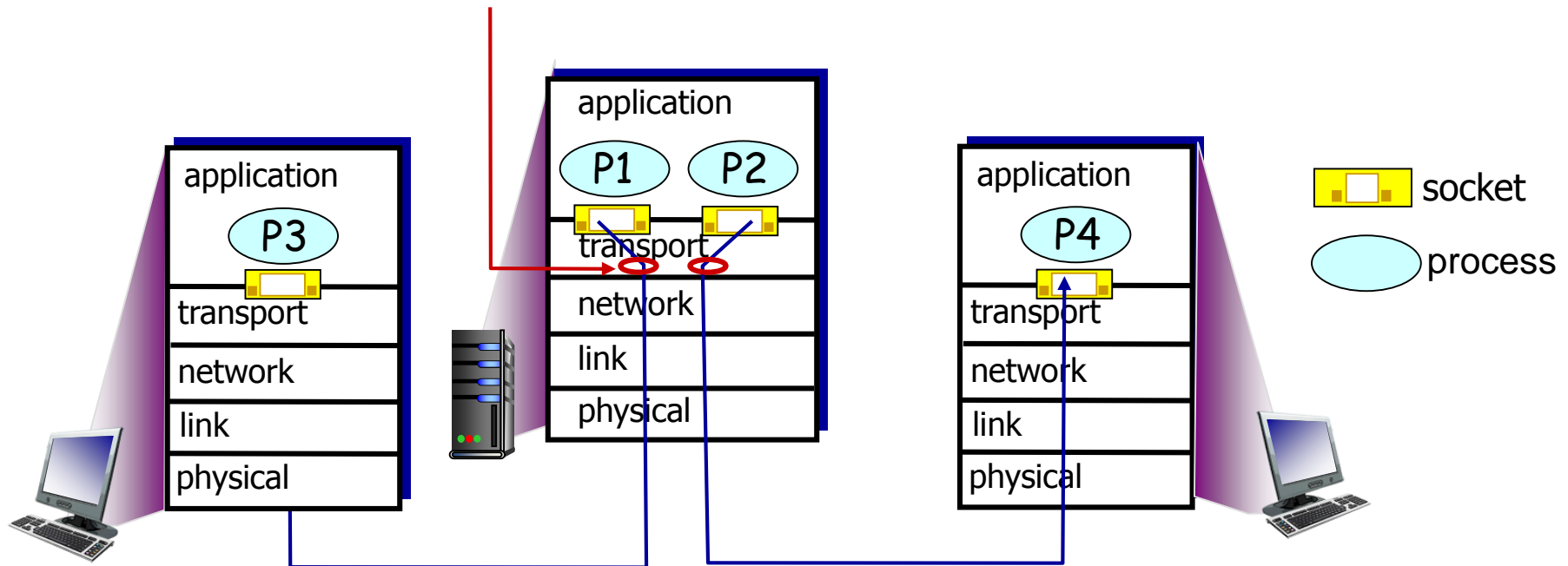


Transportschicht: Multiplex

- Multiplex/De-Multiplex: Zusammenführen/Trennen mehrerer Datenströme auf einen Datenstrom
- Viele Datenströme zwischen Anwendungen und Transportprotokoll, ein Datenstrom zwischen Transportprotokoll und IP

Multiplex am Sender:

Abfertigung von Daten vieler Sockets, Hinzufügen des Transport-Header (horizontale Kommunikation, De-Multiplex)

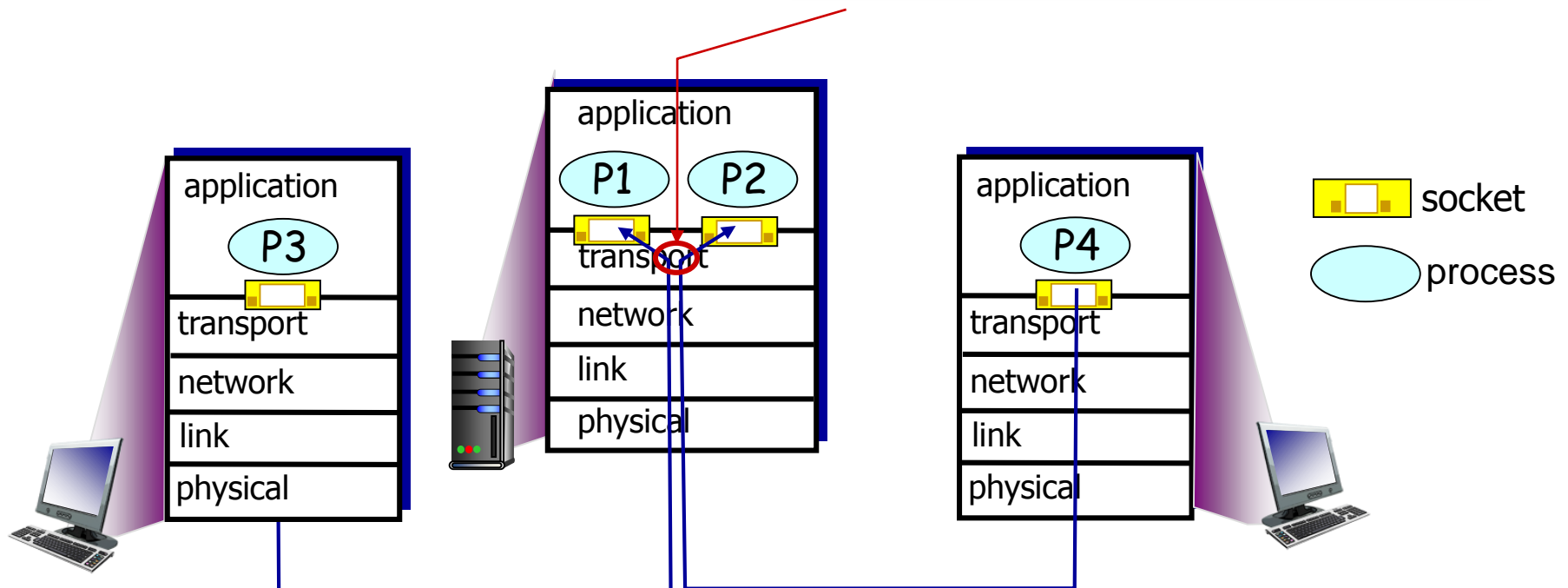


Transportschicht: De-Multiplex

- Multiplex/De-Multiplex: Zusammenführen/Trennen mehrerer Datenströme auf einen Datenstrom
- Viele Datenströme zwischen Anwendung und Transportprotokoll, ein Datenstrom zwischen Transportprotokoll und IP

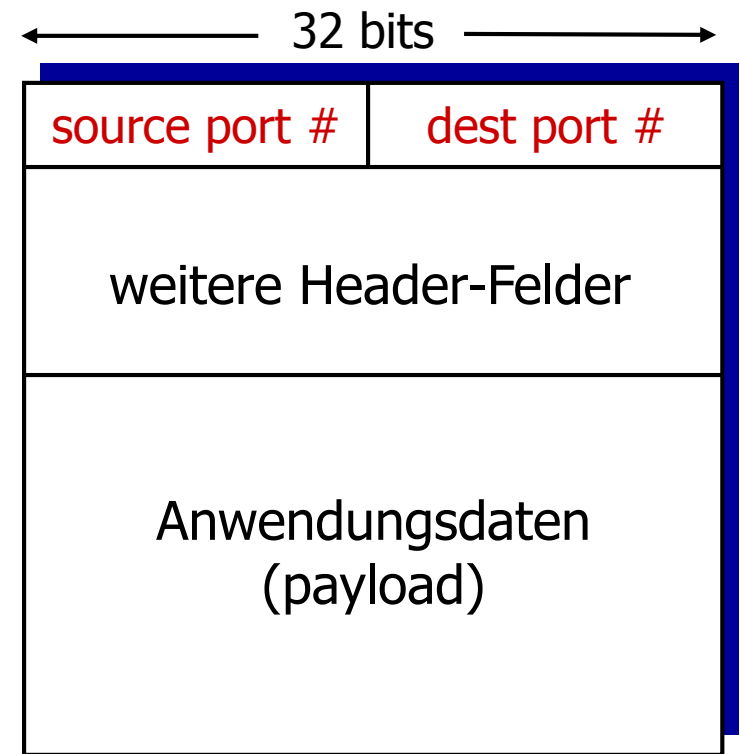
De-Multiplex am Empfänger:

Ausliefern der Daten an den richtigen Socket, Identifikation über Transport-Header



Wie De-Multiplex funktioniert

- Transport-Protokoll identifiziert richtiges Socket über IP Adressen und Ports
- Host empfängt IP Datagramm
 - im IP Header stehen Source und Destination IP Adresse
 - jedes Datagramm enthält ein Segment der Transportschicht
 - im Transport-Header jedes Segments stehen Source und Destination Port
- im Host werden Segmente über IP Adressen und Port Nummern an das richtige Socket (den richtigen Prozess) übergeben



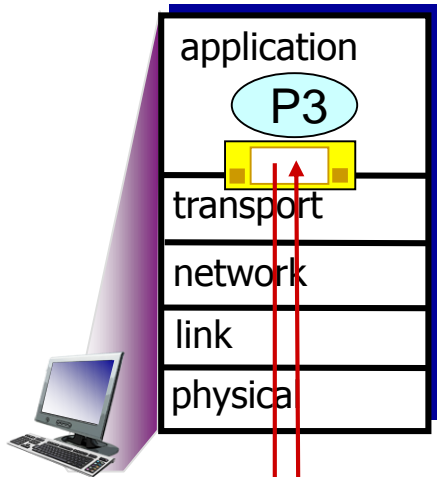
TCP/UDP Segment Format

UDP - Demux

- UDP identifiziert Socket NUR über Destination Port
- Pakete mit gleichem Destination Port aber unterschiedlichen IP-Adressen/Source Ports gehen an gleiches Socket

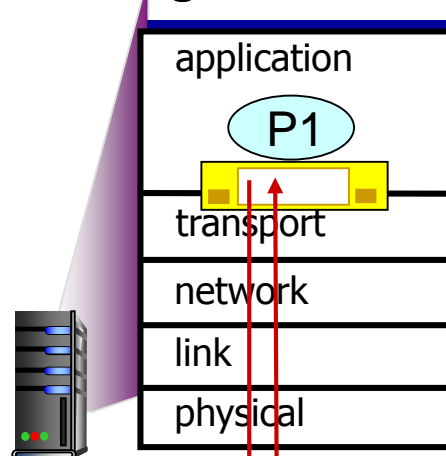
DatagramSocket

```
mySocket2 = new  
DatagramSocket (9157);
```



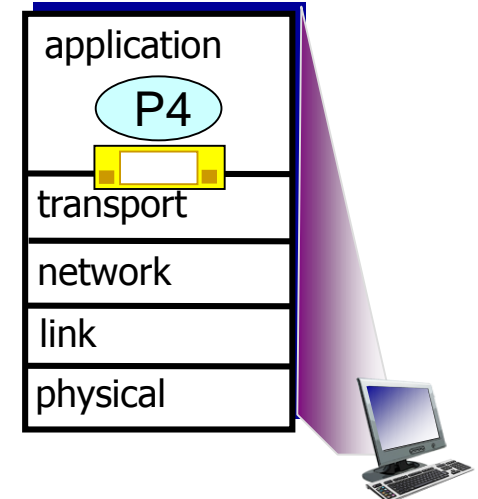
DatagramSocket

```
serverSocket = new  
DatagramSocket (6428);
```



DatagramSocket

```
mySocket1 = new  
DatagramSocket (5775);
```

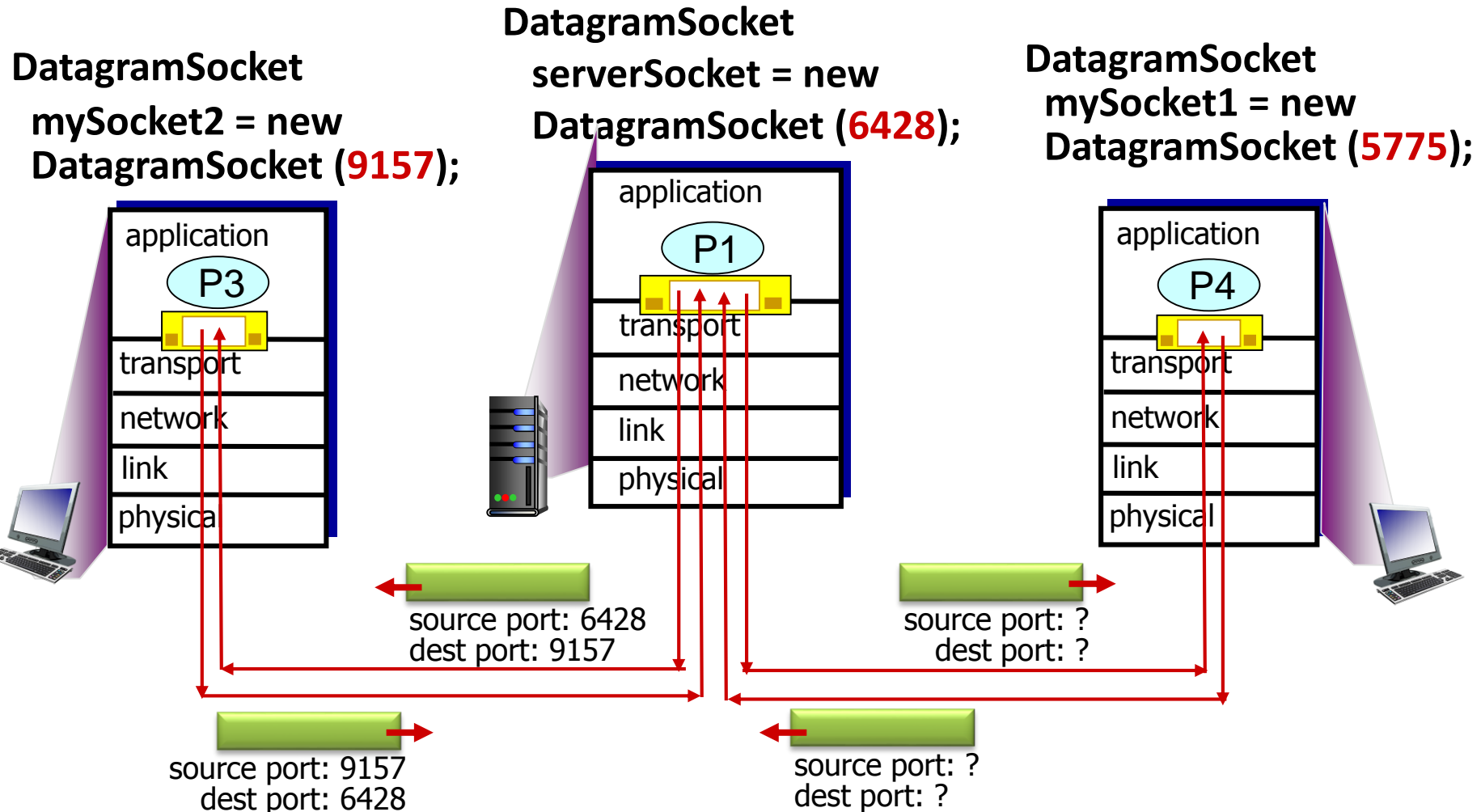


source port: 6428
dest port: 9157

source port: 9157
dest port: 6428

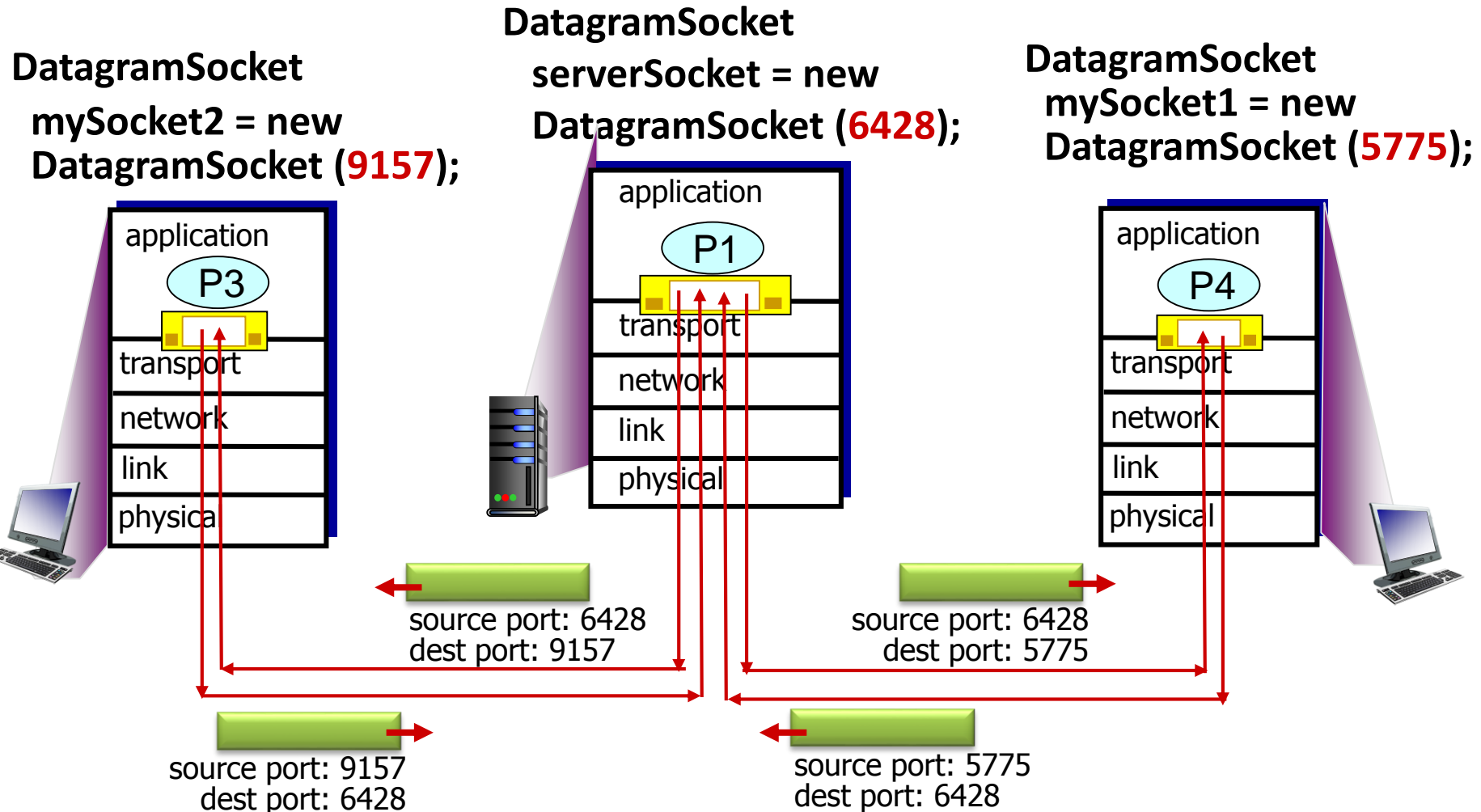
UDP - Demux

- UDP identifiziert Socket NUR über Destination Port
- Pakete mit gleichem Destination Port aber unterschiedlichen IP-Adressen/Source Ports gehen an gleiches Socket

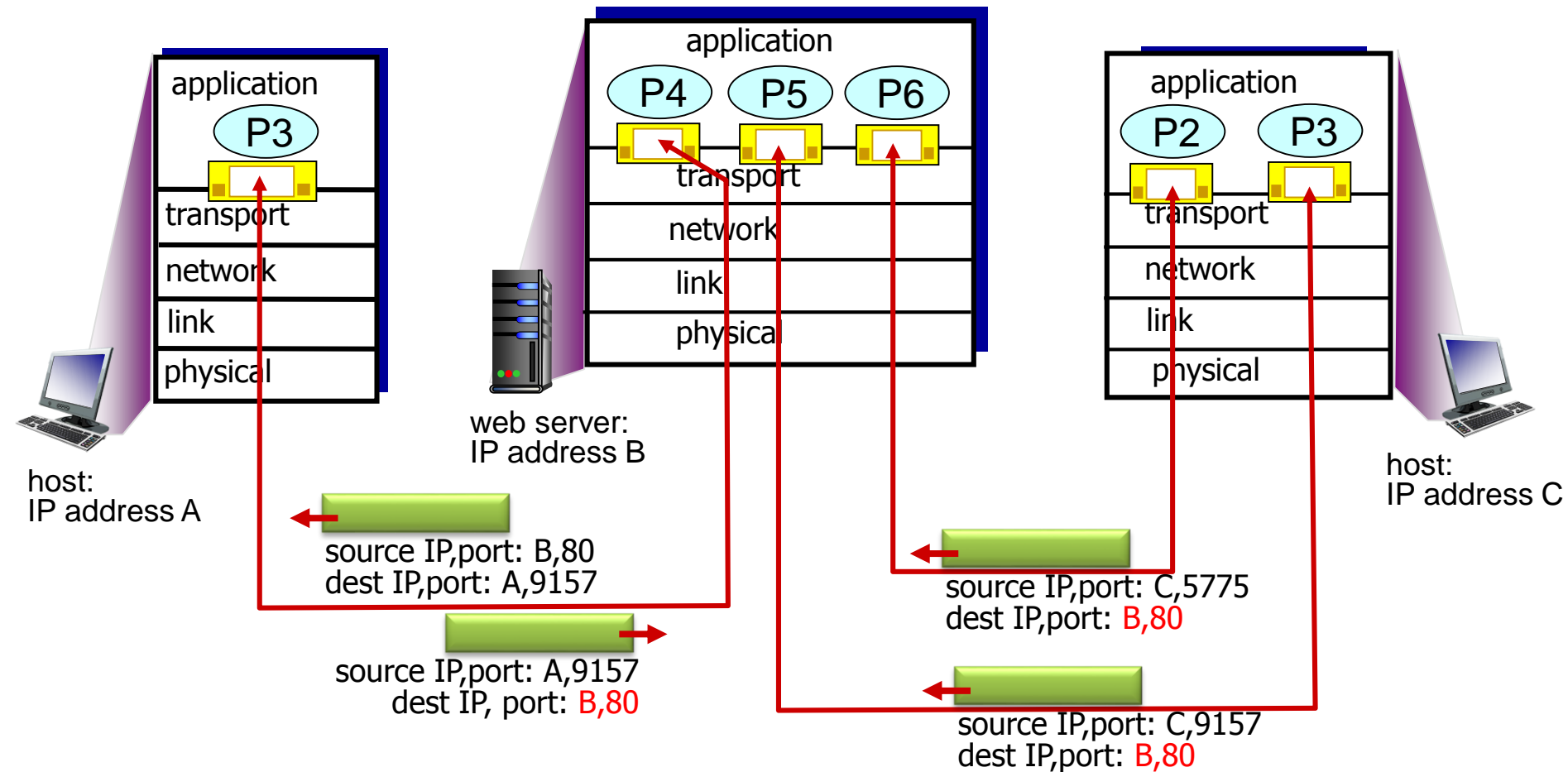


UDP - Demux

- UDP identifiziert Socket NUR über Destination Port
- Pakete mit gleichem Destination Port aber unterschiedlichen IP-Adressen/Source Ports gehen an gleiches Socket



TCP Demux



three segments, all destined to IP address: B,
dest port: 80 are demultiplexed to *different* sockets

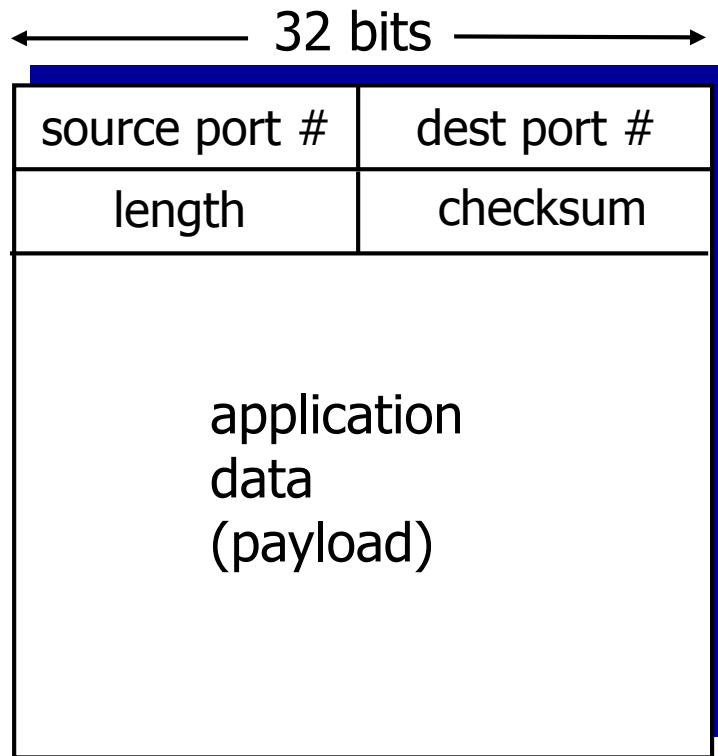
4.1 Multiplexing

4.2 UDP

4.3 TCP

4.4 Zusammenfassung

UDP (User Datagram Protocol): Paketformat und Protokoll



UDP segment format

length: Länge in Bytes inkl. Header
16 Bit → maximale Paketgröße 65536 Bytes
checksum: ermöglicht Erkennung von Bit-Fehler

- Minimales Transportprotokoll
 - zustandslos
 - kein Handshake zum Verbindungsaufbau
 - jedes UDP-Segment wird unabhängig von allen anderen behandelt
 - UDP-Sender hat "Aufgabe erledigt", sobald Nachricht übertragen ist
- Verbindungsloser "Best-Effort"-Dienst, UDP-Segmente können:
 - verloren gehen
 - in falschen Reihenfolge an die Anwendung ausgeliefert werden
- einige UDP-Anwendungen
 - DNS, IP-Telefonie (Sprache, Video), IP-Multicast/Broadcast, SNMP