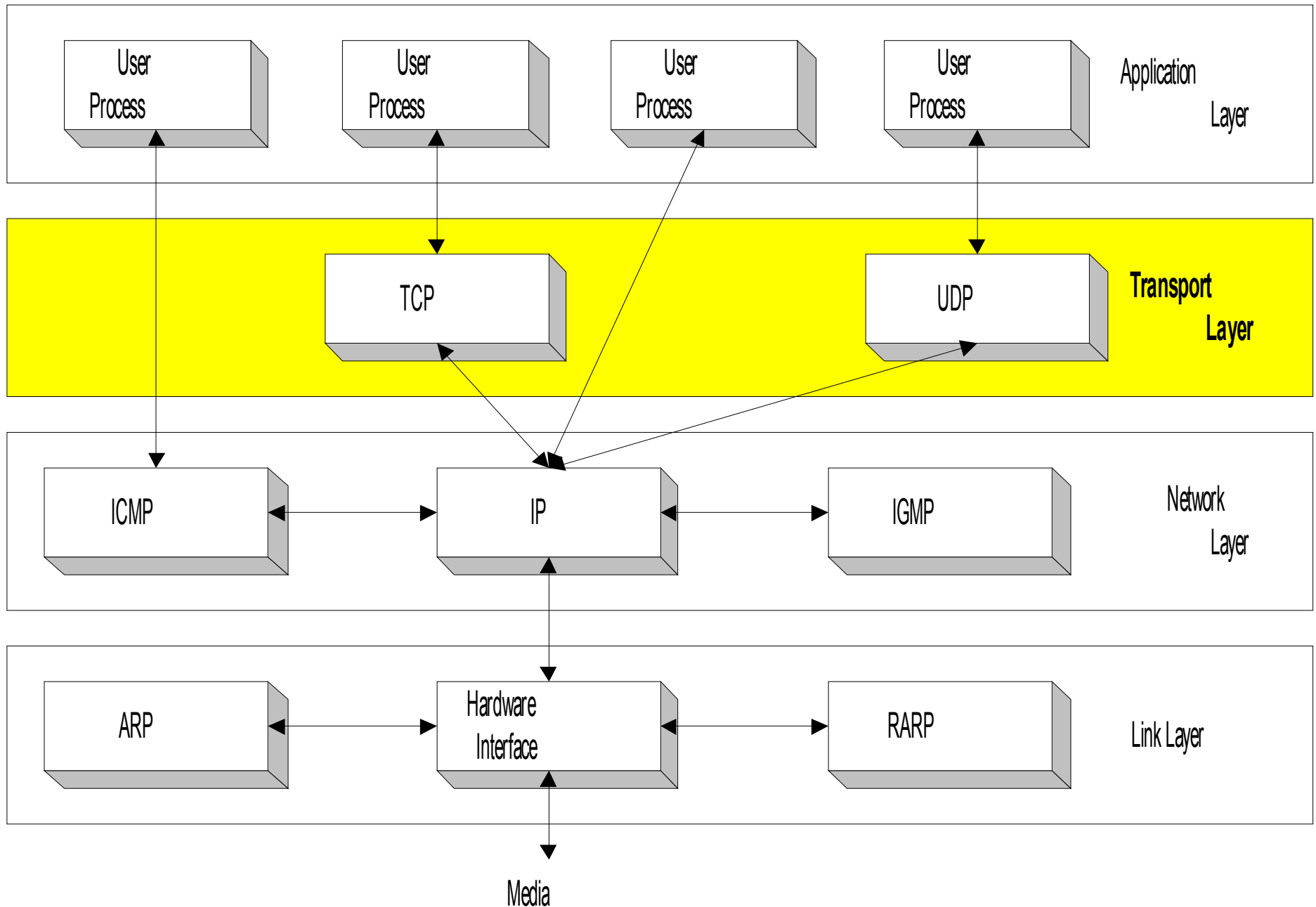


UDP-, MTU- und IP- Fragmentierung

Jörn Stuphorn
stuphorn@rvs.uni-bielefeld.de

- 13. April 2005 Unix-Umgebung
- 20. April 2005 Unix-Umgebung
- 27. April 2005 Unix-Umgebung
- 4. Mai 2005 ARP, ICMP, ping
- 11. Mai 2005 IP-Adressen & Subnetzmasken
- 18. Mai 2005 Einführung in Bridging, Routing, ...
- 25. Mai 2005 IOS, Spanning-Tree
- 1. Juni 2005 IOS Befehle, Bridging, Routing
- 8. Juni 2005 Statisches Routing
- 15. Juni 2005 *UDP-, MTU- und IP-Fragmentierung***
- 22. Juni 2005 TCP-Verbindungen und -Datenfluss*
- 29. Juni 2005 DHCP und NTP*
- 6. Juli 2005 NAT und Firewalls*
Verschlüsselung, Vertraulichkeit,
- 13. Juli 2005 Authentisierung*
- 20. Juli 2005 Sichere Anwendungen*

Routing



- Die beiden verbreiteten Transport-Layer Protokolle des Internets:

UDP - User Datagram Protocol

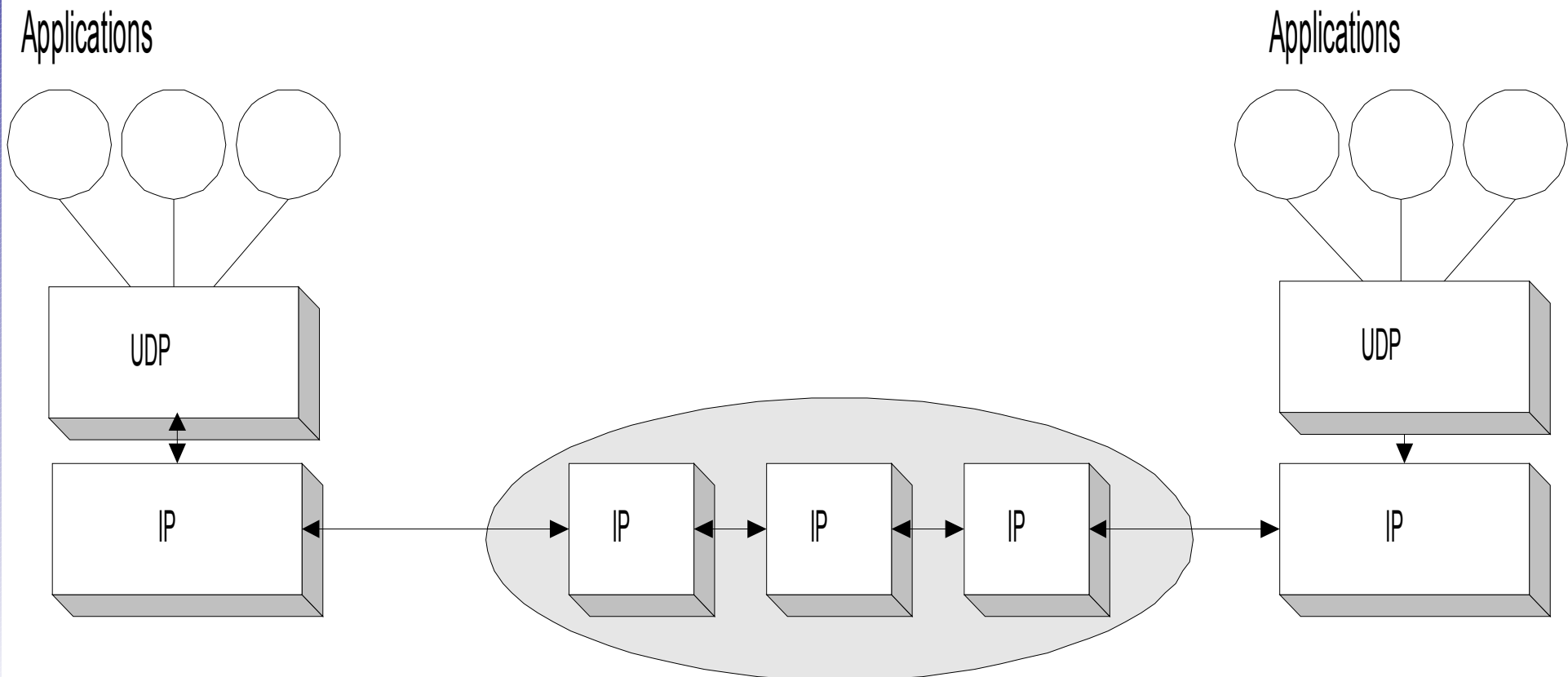
- Datagramm orientiert
- Unzuverlässig, Verbindungslos
- Einfach
- Unicast und Multicast Übertragung
- für einige Anwendungen nützlich (z.B. Multimedia)
- oft für Dienste verwendet
 - Netzwerk Management (SNMP), Routing (RIP), Namensauflösung (DNS), etc.

TCP - Transmission Control Protocol

- Stream orientiert
- Zuverlässig, Verbindungsorientiert
- Komplex
- Nur Unicast Übertragung
- Für die meisten Internet Anwendungen verwendet::
 - Web (http), eMail (smtp), Dateiübertragung (ftp), Terminal (telnet), etc.

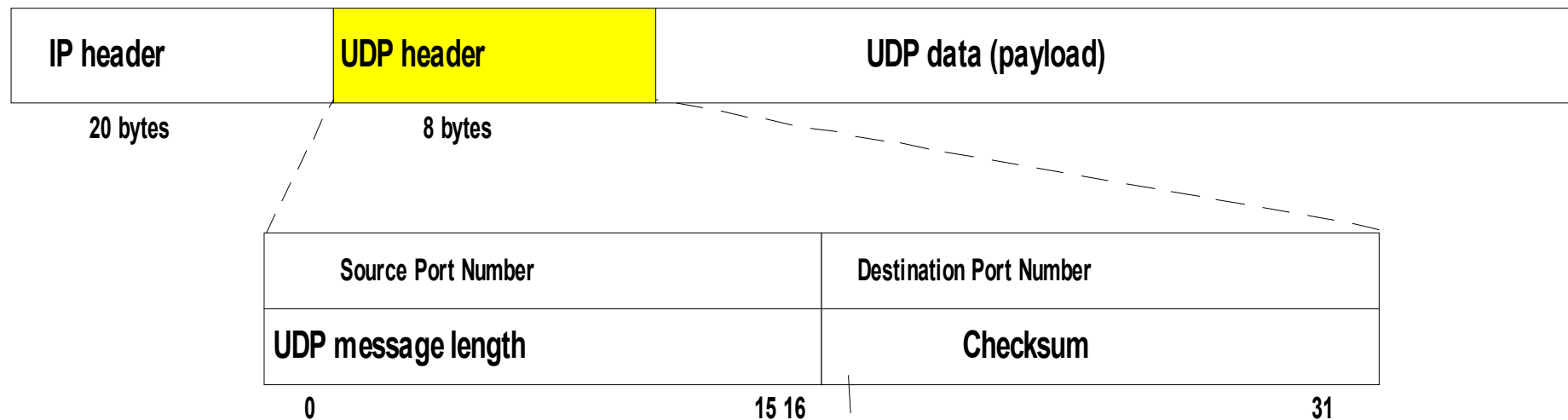
UDP

- Unterstützt unzuverlässige Übertragung von Datagrammen
- UDP erweitert Host-to-Host Übertragung von IP-Datagramm um Application-to-Application Service
- UDP fügt multiplexing und demultiplexing hinzu



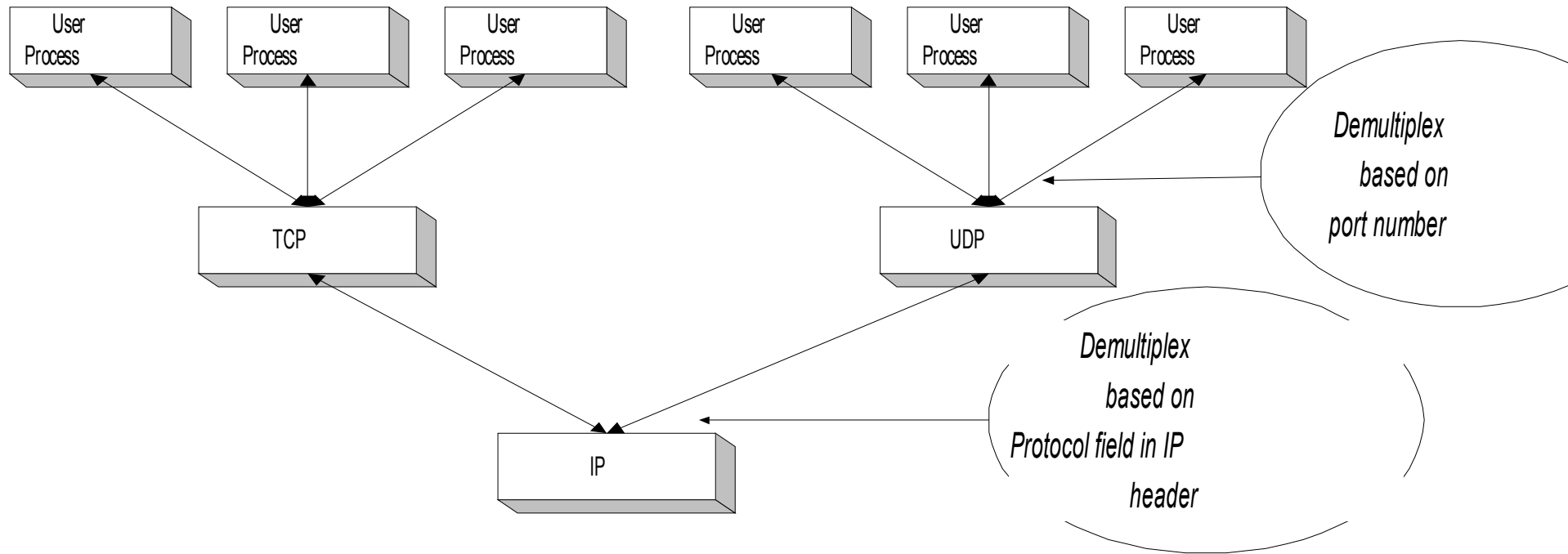
UDP Header

- Port Nummern identifizieren sendende und empfangende Anwendung
 - Maximaler Port: 65535 (16bit)
- Nachrichtenlänge zwischen 8 und 65535 Byte
- Checksum für UDP Header und UDP Daten



Port Nummern

- UDP und TCP verwenden Port Nummern um Anwendungen zu identifizieren
- Global Eindeutige Adresse im Transport-Layer:
 - IP Adresse
 - Port Nummer



UDP Experiment

- Packetgenerator: hping3
- Exercise 4
 - `sock -u -i -ni -w10000 remote_host echo`

 - Beobachte die übertragenen Pakete
 - Beschreibe die Pakete und die Felder, die für Fragmentierung wichtig sind

- Exercise 5
 - Rufe den folgende Befehl mit variierenden SIZE auf
 - `sock -u -i -n1 -w<SIZE> remote_host echo`

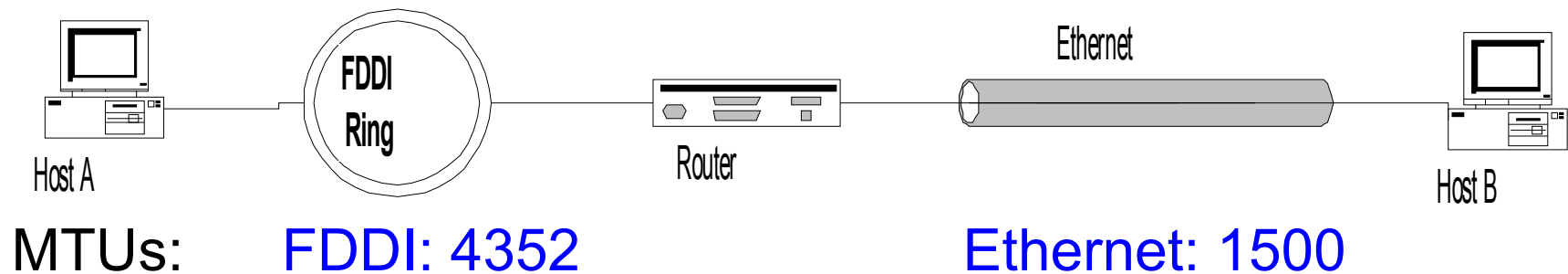
 - Wie groß ist die maximale Größe eines UDP Datagramms wenn Fragmentierung erlaubt ist?

- DataLink Protokoll hat begrenzte Rahmengröße
- Rahmengröße wirkt sich auf maximale Datenmenge, die ein IP Datagramm kapseln kann, aus.
- MTU (Maximum Transmission Unit)
 - MTUs for various data link layers:

Ethernet: 1500	FDDI: 4352
802.3: 1492	ATM AAL5: 9180
802.5: 4464	PPP: 296
- Was passiert, wenn IP Datagramm größer als MTU ist?
 - Datagramm wird fragmentiert
- Was passiert, wenn Route über Netzwerke mit unterschiedlichen MTUs läuft?
 - Die niedrigste MTU wird als Pfad MTU (Path MTU) verwendet.

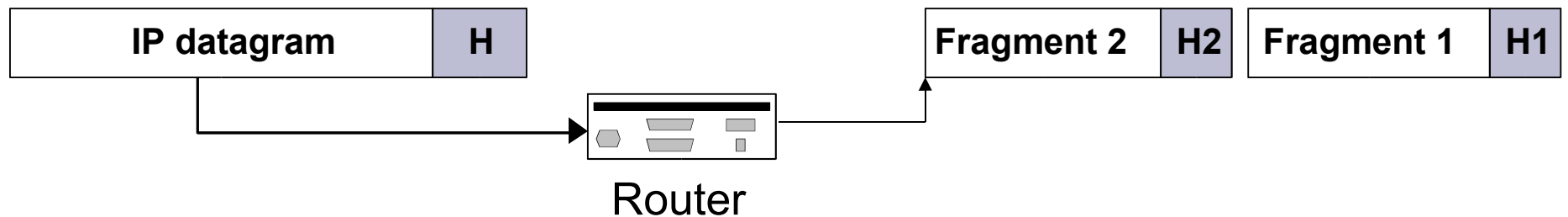
IP Fragmentation (1)

- Host A sendet ein großes IP Datagramm and Host B
- Wie geht der Router damit um?
- IP Router splittet das Datagramm in mehrere Fragmente
- Die Größe aller Fragmente (bis auf das Letzte) ist ein Vielfaches von 8Byte.



IP Fragmentation (2)

- Fragmentierung kann auf beim Sender oder einer Zwischenstation passieren.
- Das gleiche Datagramm kann öfters fragmentiert werden.
- Zusammensetzen der empfangenen Fragmente erfolgt nur beim Empfänger



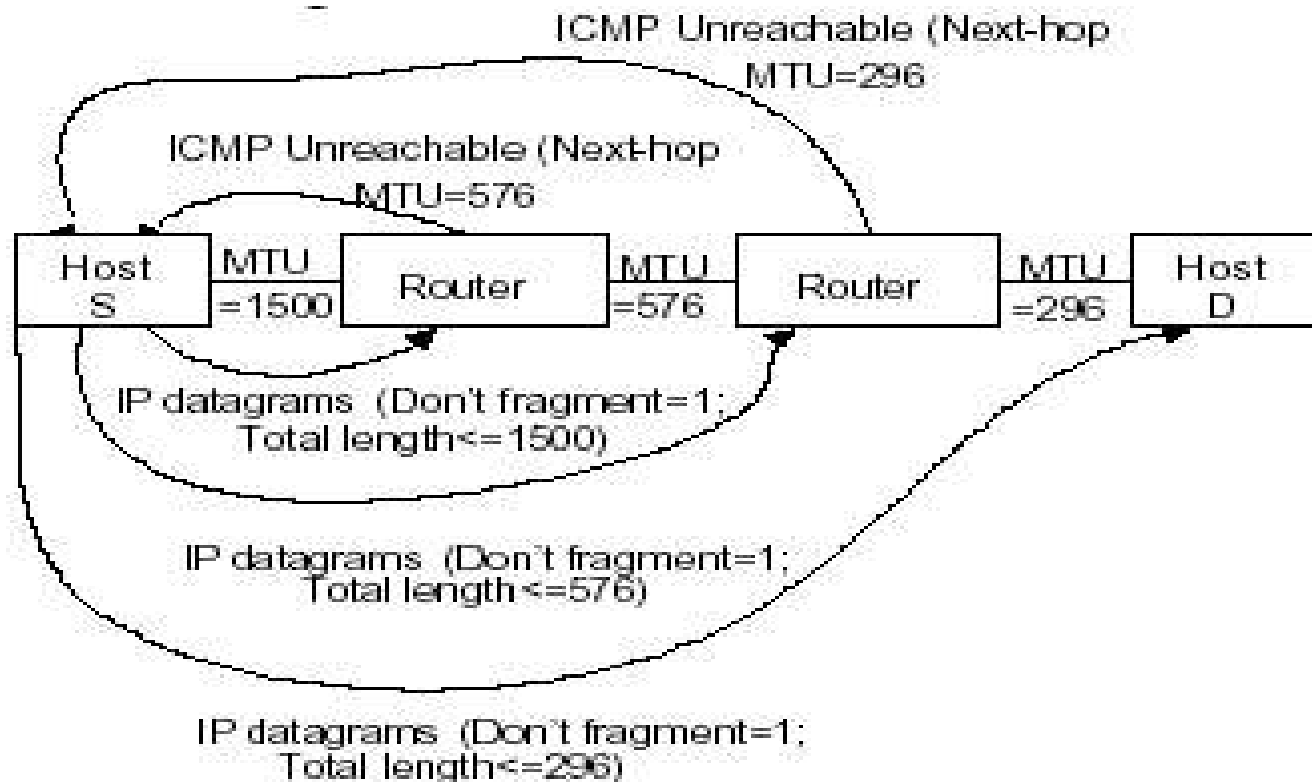
IP Header Informationen

- Die folgenden Header Felder betreffen Fragmentierung:
 - *Identification* für alle Fragmente gleich
 - *Flag* beinhaltet ein „more fragments“ bit (ein „don't fragment bit“ existiert auch)
 - *Fragment offset* zeigt den offset (Vielfaches von 8Byte) an
 - *Total length* gibt die Größe des Fragments an.

version (4 bits)	header length	Type of Service	Total Length (in bytes)	
Identification			flags	Fragment Offset
TTL Time-to-Live (8 bits)		Protocol (8 bits)	Header Checksum (16 bits)	
.....				

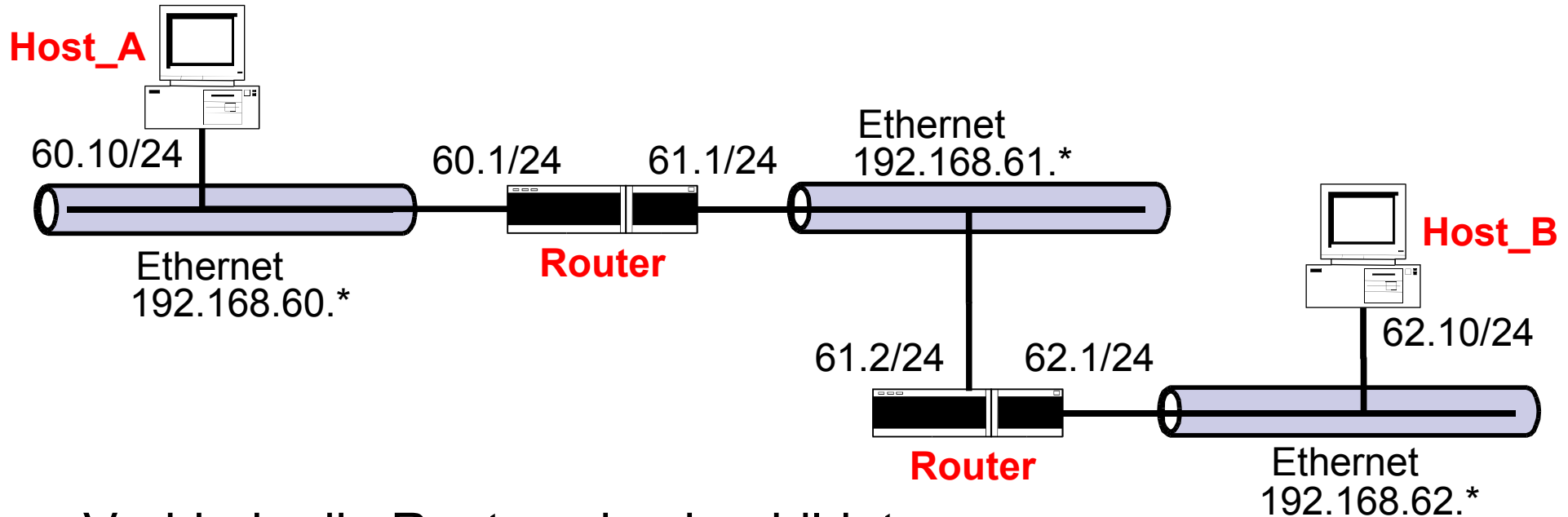
Pfad MTU Ermittlung

- Ein host sendet eine Folge von IP datagrammen mit unterschiedlichen Größen und dem „don't fragment“-Flag gesetzt.



Experiment MTU Ermittlung

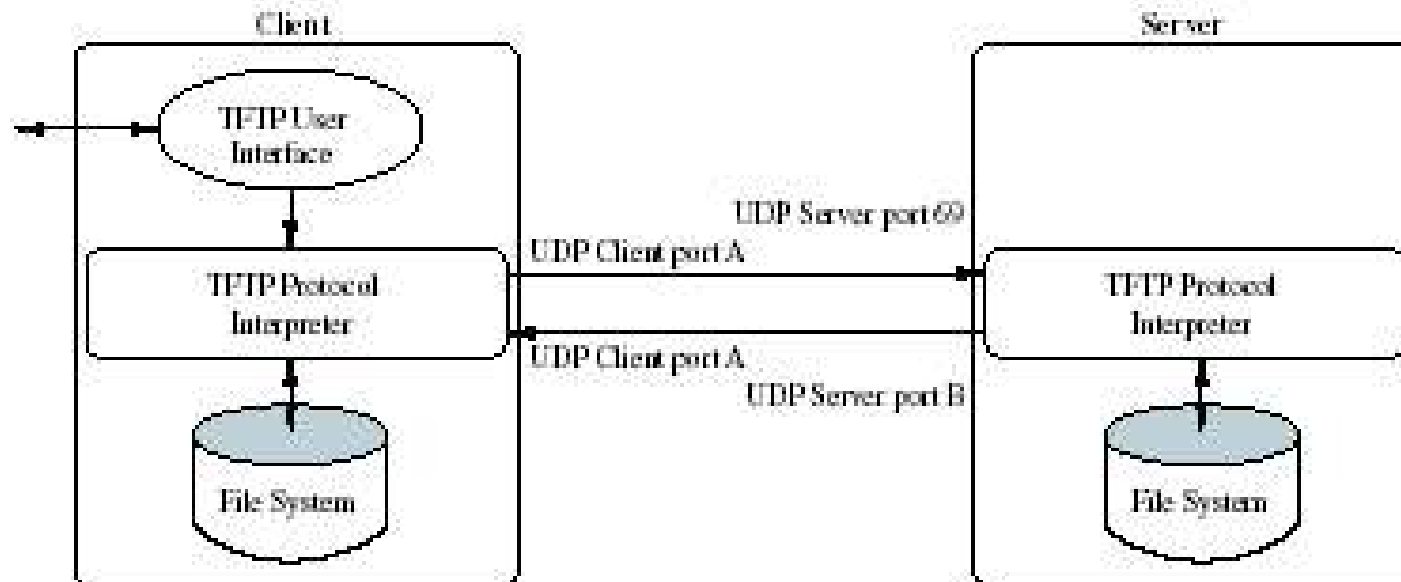
■ Exercise 6



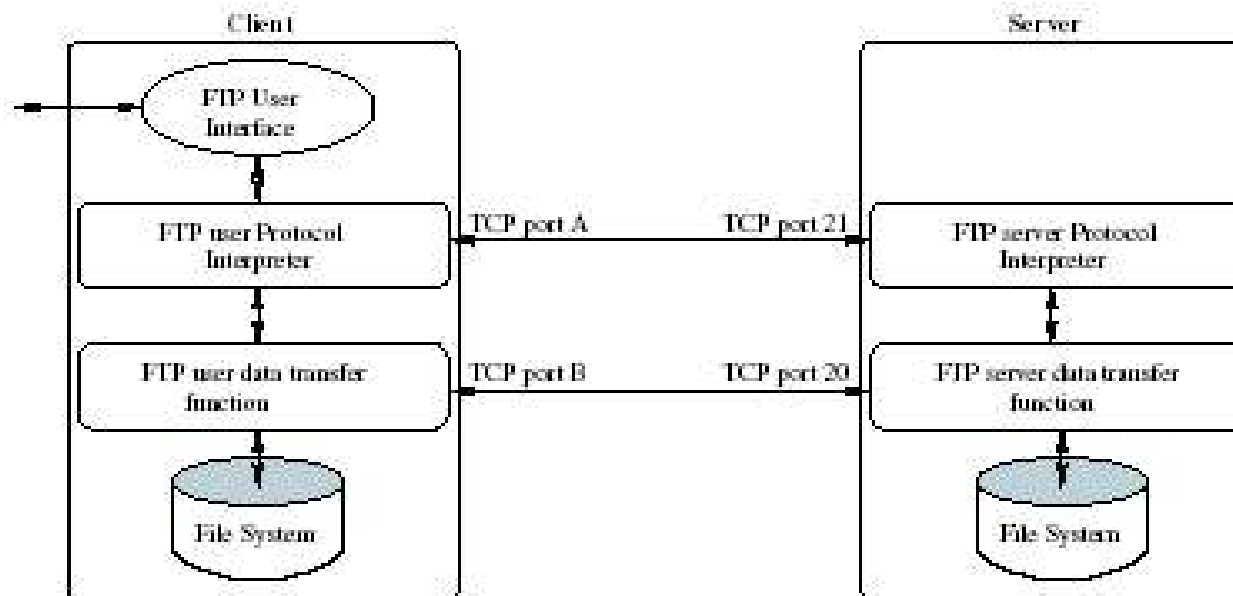
- Verbinde die Router wie abgebildet
- Telnet zu jedem Router (RIP aktivieren)
- Router2: eth1 MTU auf 500 Byte setzen
- Teste Verbindung mit ping von Host_A nach Host_B
- Starte sock Server auf Host_B `sock -u -s 5555`
- Host_A: `sock -i -u -n10 -w1200 -p remote_host 5555`
- Host_A: `sock -i -u -n10 -w1200 -p5 remote_host 5555`

- Die meisten Netzwerkanwendung sind als Client/Server Systeme konzipiert
 - Ein Server bietet die Netzwerkdienst an
 - Clients greifen auf diese Dienste zurück
- Server benutzen bekannte Ports und laufen rund um die Uhr (im optimalen Fall)
- Client benutzt einen kurzlebigen Port, der nach Verbindungsabbau wieder frei gegeben wird
- Wenn ein Client einen Dienst anfordert, der nicht angeboten wird:
 - UDP: ICMP port unreachable Nachricht wird zurückgegeben
 - TCP: Verbindung wird zurückgesetzt.

- Trivial File Transfer Protocol
- Verwendet UDP (nicht verbindungsorientiert und unzuverlässig)
 - Unregelmäßige Übertragung kleiner Dateien
 - Durchsatz ist nicht übermäßig wichtig
- Verwendet stop-and-wait Flow Windows Control Algorithmus
 - Warte auf ACK bevor nächstes Paket gesendet wird
 - Ein verlorenes Paket erzeugt Timeout und Retransmission
- Konzipiert für plattenlose Systeme (Thin-Clients) um Konfigurationsinformationen während Bootvorgang abzurufen

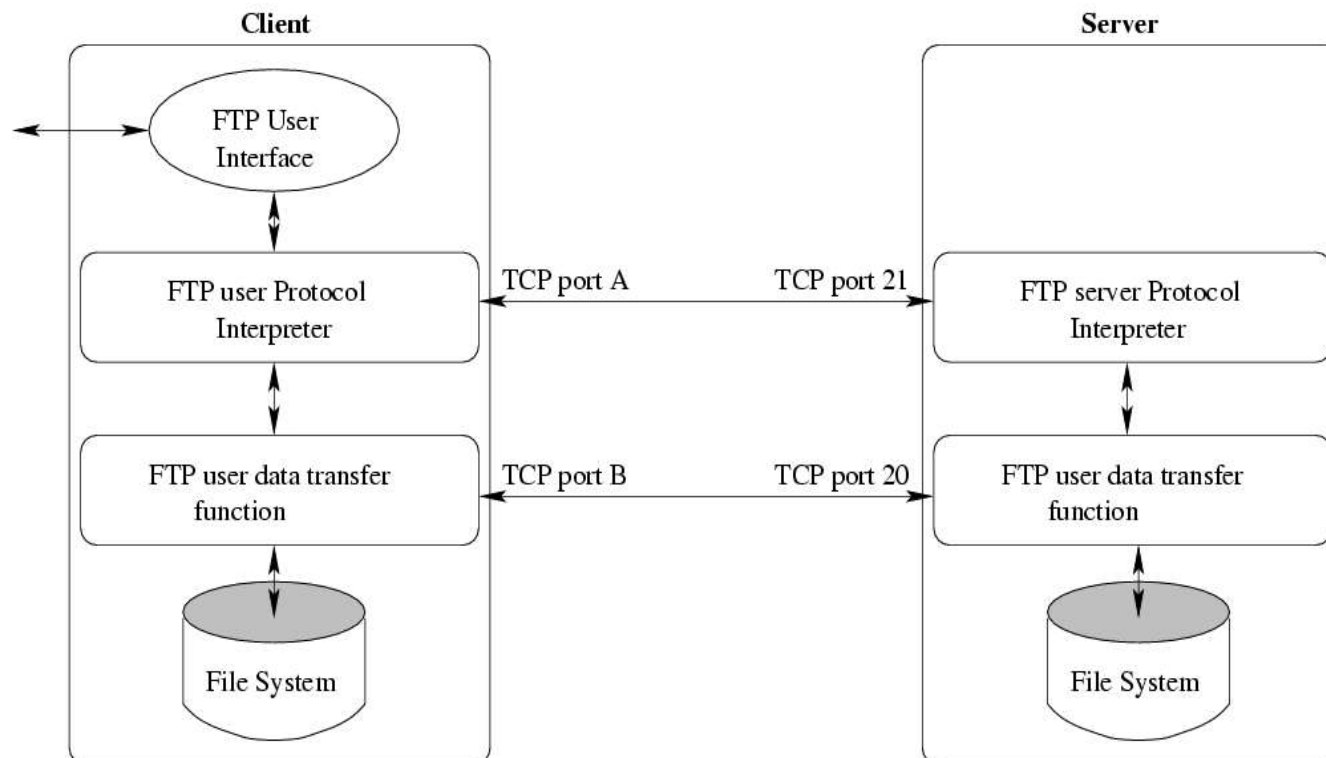


- File Transfer Protocol
- Verwendet 2 TCP Verbindungen
 - Kontrolle: Port 21 auf dem Server
 - Daten: Port 20 auf dem Server



Verbindungs Management

- Kontrollverbindung bleibt während client/server Verbindung aktiv
- Verbindungsaufbau der Datenverbindung wird vom Client gesteuert
- Client wählt einen vorübergehenden Port für sein Ende der Datenverbindung
- Client sendet PORT Kommando an Server über Kontrollverbindung
- Server überträgt Daten an Client-Port
- Ermöglicht viele FTP-Sitzungen von vielen Clients zu einem Server



- Exercise 8
 - Übertragung einer großen Datei mit TFTP
 - Beobachte Transfer mit ethereal

 - Kopiere Datei von yoda nach windu mit TFTP

 - Welche Nachrichten wurden übertragen?

- Exercise 9
 - Übertragung einer großen Datei mit FTP
 - Beobachte Transfer mit ethereal

 - Kopiere Datei von yoda nach windu mit FTP

 - Welche Nachrichten wurden übertragen?

- Exercise 7
 - Übertragung einer großen Datei mit TFTP
 - Beobachte Transfer mit ethereal

 - Kopiere Datei von yoda nach windu mit FTP
 - Notiere Übertragungsdauer
 - Kopiere Datei von yoda nach windu mit TFTP
 - Notiere Übertragungsdauer

 - Welche Variante ist schneller?
 - Warum?