Transportschicht

NETZWERKTECHNIK / SEMESTER 1 UND 2

1

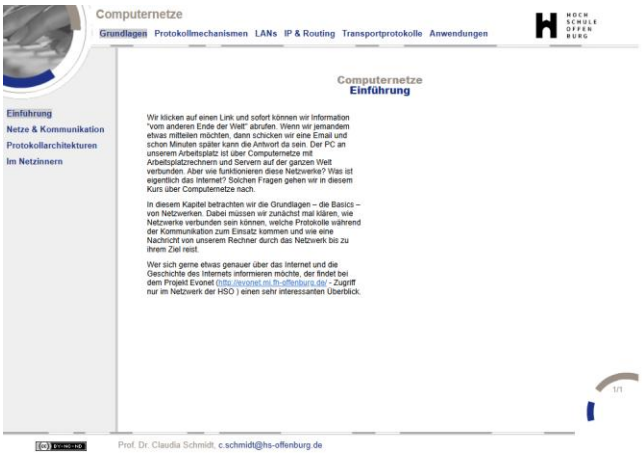
AGENDA

2

NETZWERKTECHNIK / SEMESTER 1 und 2

Computernetzwerke Einführung

https://mi-learning.mi.hs-offenburg.de/CNet/grundlagen/einfuehrung/index.html



tgm [Quelle: https://mi-learning.mi.hs-offenburg.de/CNet/grundlagen/einfuehrung/index.html - letzter Abruf 21.07.2025]

4

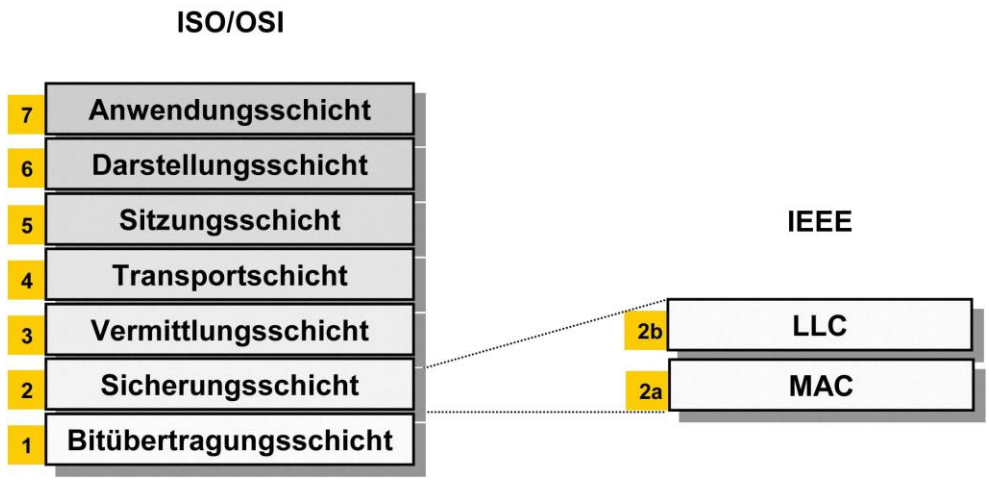
01

Transport von Daten

5

NETZWERKTECHNIK / SEMESTER 1 und 2

Einordnung im ISO/OSI Modell



[Quelle: https://www.airnet.de/cr1-gfe/de/html/TrFddiIEEE802_learningObject6.xml - letzter Abruf 21.07.2025]

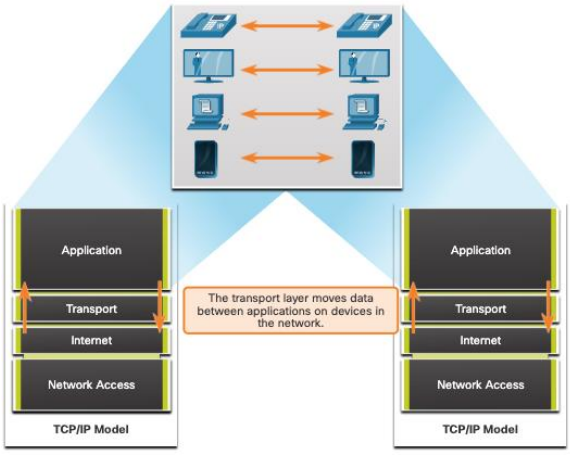
6

NETZWERKTECHNIK / SEMESTER 1 und 2

Rolle der Transportschicht

Die Transportschicht ist:

- verantwortlich für die **logische Kommunikation** zwischen Anwendungen, die auf verschiedenen Hosts ausgeführt werden.
- die **Verbindung zwischen der Anwendungsschicht und den unteren Schichten**, die für die Netzwerkübertragung verantwortlich sind.



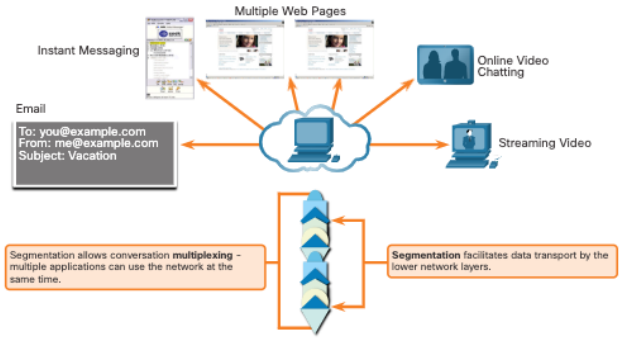
7

NETZWERKTECHNIK / SEMESTER 1 und 2

Verantwortung der Transportschicht

Die Transportschicht hat folgende Aufgaben:

- Verfolgen einzelner **Konversationen**
- **Segmentieren** von Daten und erneutes Zusammenfügen von Segmenten
- hinzufügen von **Kopfzeileninformationen**
- Identifizieren, Trennen und Verwalten mehrerer Konversationen
- Verwendet **Segmentierung und Multiplexing**, um die Verschachtelung verschiedener Kommunikationsgespräche im selben Netzwerk zu ermöglichen

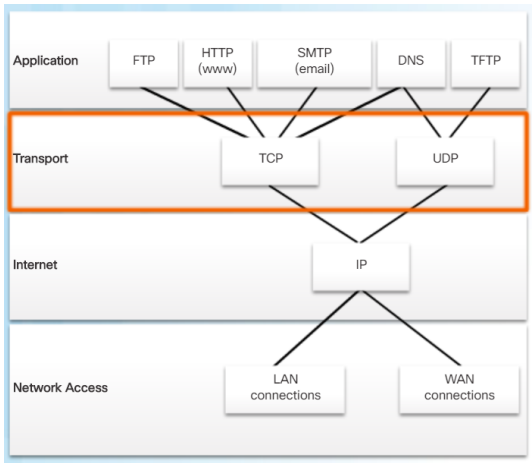


8

NETZWERKTECHNIK / SEMESTER 1 und 2

Protokolle der Transportschicht

- Internet Protokoll (IP) spezifiziert nicht, wie die Zustellung oder der Transport der Pakete erfolgt.
- Protokolle der Transportschicht geben an, wie Nachrichten zwischen Hosts übertragen werden, und sind für die **Verwaltung der Zuverlässigkeitsanforderungen einer Konversation verantwortlich**.
- Die Transportschicht umfasst die Protokolle **TCP** und **UDP**.



9

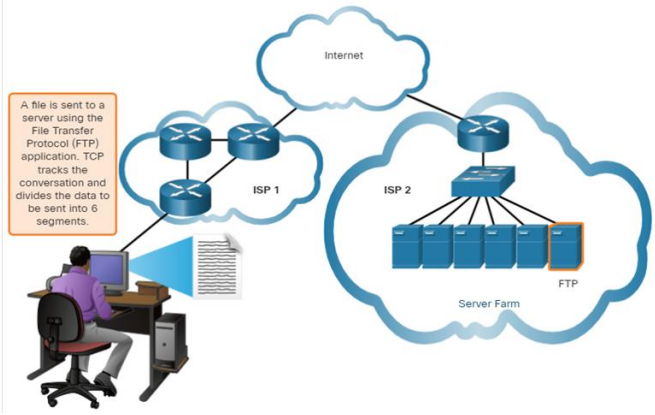
NETZWERKTECHNIK / SEMESTER 1 und 2

Transmission Control Protocol (TCP)

- TCP bietet **Zuverlässigkeit und Flusskontrolle**.

Grundlegende TCP-Operationen:

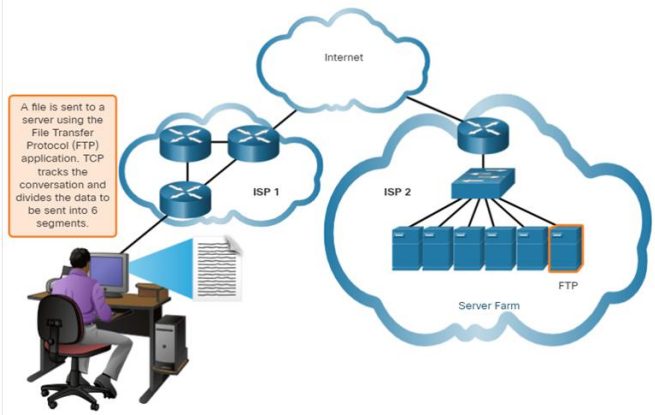
- **Nummerieren und verfolgen von Datensegmenten**, die von einer bestimmten Anwendung an einen bestimmten Host übertragen werden
- **Empfangene Daten bestätigen**
- **Erneute Übermitteln von nicht bestätigten Daten** nach einer bestimmten Zeit
- Daten, die möglicherweise in der falschen Reihenfolge eintreffen, **sequenzieren**.
- Daten mit einer effizienten, für den Empfänger **akzeptablen Rate senden**.



NETZWERKTECHNIK / SEMESTER 1 und 2

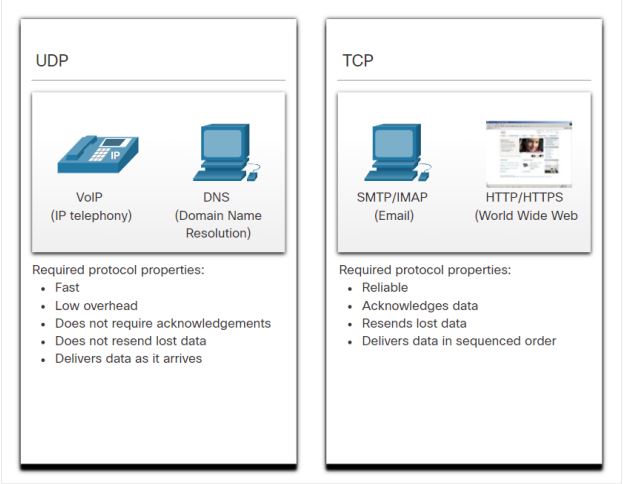
User Datagram Protocol (UDP)

- UDP bietet die grundlegenden Funktionen für die Bereitstellung von Datagrammen zwischen den entsprechenden Anwendungen, mit **sehr geringem Overhead und Datenprüfung**.
- UDP ist ein **verbindungsloses Protokoll**.
- UDP ist als B**est-Effort-Übermittlungsprotokoll** bekannt, da es keine Bestätigung gibt, dass die Daten am Ziel empfangen werden.



Das richtige Transport-Layer-Protokoll für die richtige Anwendung

- UDP wird auch von Request-and-Reply-Anwendungen verwendet, bei denen der Datenverbrauch minimal ist und die erneute Übertragung schnell erfolgen kann.
- Wenn es wichtig ist, dass alle Daten ankommen und in der richtigen Reihenfolge verarbeitet werden können, wird TCP als Transportprotokoll verwendet.



12

02

TCP-Überblick

13

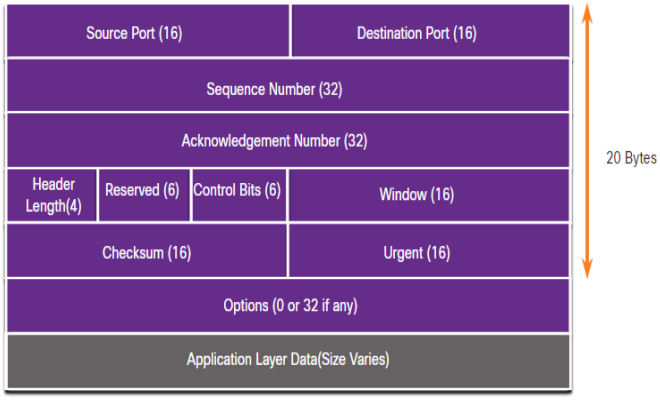
TCP Features

- Richtet eine Sitzung ein - TCP ist ein **verbindungsorientiertes Protokoll**, das eine permanente Verbindung (oder Sitzung) zwischen Quell- und Zielgeräten aushandelt und aufbaut, bevor Datenverkehr weitergeleitet wird.
- Gewährleistet eine zuverlässige Lieferung - Aus vielen Gründen ist es möglich, dass ein Segment beschädigt wird oder vollständig verloren geht, wenn es über das Netzwerk übertragen wird. TCP **stellt sicher, dass jedes Segment, das von der Quelle gesendet wird, am Ziel ankommt**.
- Bietet Lieferung in derselben Reihenfolge - Da Netzwerke möglicherweise mehrere Routen mit unterschiedlichen Übertragungsraten bereitstellen, können **Daten in der falschen Reihenfolge ankommen**.
- Unterstützt Flusskontrolle - Netzwerk-Hosts verfügen über begrenzte Ressourcen (d. h. Arbeitsspeicher und Rechenleistung). Wenn TCP erkennt, dass diese Ressourcen überlastet sind, kann es anfordern, dass die **sendende Anwendung die Geschwindigkeit des Datenflusses** reduziert.

14

TCP Header

- TCP ist ein zustandsbehaftetes Protokoll, was bedeutet, dass es den Status der Kommunikationssitzung verfolgt.
- TCP zeichnet auf, welche Informationen es gesendet hat und welche Informationen bestätigt wurden.



15

NETZWERKTECHNIK / SEMESTER 1 und 2

Felder des TCP Header

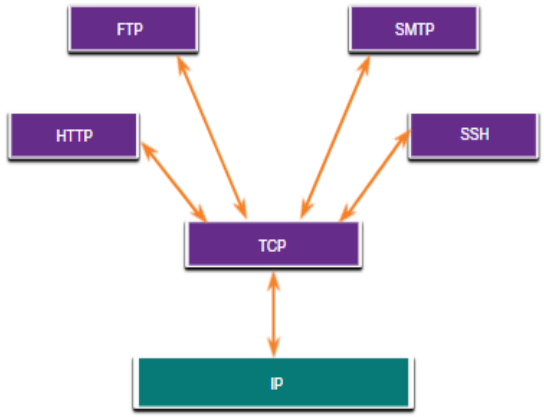
TCP Header Field	Description
Source Port	16-Bit-Feld, das zur Identifizierung der Quellanwendung anhand der Portnummer verwendet wird.
Destination Port	16-Bit-Feld, das zur Identifizierung der Zielanwendung anhand der Portnummer verwendet wird.
Sequence Number	32-Bit-Feld, das für die Datenreassemblierung verwendet wird.
Acknowledgment Number	32-Bit-Feld, das verwendet wird, um anzuzeigen, dass Daten empfangen wurden und das nächste Byte von der Quelle erwartet wird.
Header Length	4-Bit-Feld, das als "Datenoffset" bezeichnet wird und die Länge des TCP-Segmentheaders angibt.
Reserved	6-Bit-Feld, das für die zukünftige Verwendung reserviert ist.
Control bits	Verwendetes 6-Bit-Feld, das Bitcodes oder Flags enthält, die den Zweck und die Funktion des TCP-Segments angeben.
Window size	16-Bit-Feld, das verwendet wird, um die Anzahl der Bytes anzugeben, die gleichzeitig akzeptiert werden können.
Checksum	16-Bit-Feld, das für die Fehlerprüfung des Segmentkopfes und der Daten verwendet wird.
Urgent	16-Bit-Feld, das verwendet wird, um anzuzeigen, ob die enthaltenen Daten dringend sind.

16

NETZWERKTECHNIK / SEMESTER 1 und 2

Anwendungen die TCP nutzen

TCP übernimmt alle Aufgaben, die mit der Unterteilung des Datenstroms in Segmente, der Bereitstellung von Zuverlässigkeit, der Steuerung des Datenflusses und der Neuanordnung von Segmenten verbunden sind.



17

03

UDP-Überblick

18

UDP Features

Zu den UDP-Funktionen gehören die folgenden:

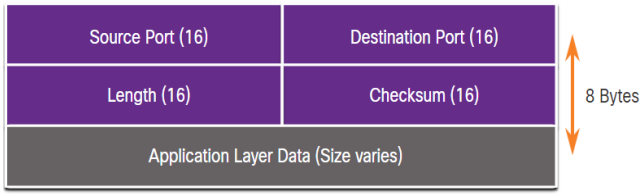
- Die Daten werden in der **Reihenfolge ihres Eingangs rekonstruiert**.
- Segmente, die verloren gehen, werden **nicht erneut gesendet**.
- Es gibt **keine Sitzungseinrichtung**.
- Der Sender wird **nicht über die Verfügbarkeit der Ressource** informiert.

19

NETZWERKTECHNIK / SEMESTER 1 und 2

UDP Header

Der UDP-Header ist **viel einfacher als der TCP-Header**, da er nur vier Felder hat und 8 Byte (d.h. 64 Bit) benötigt.



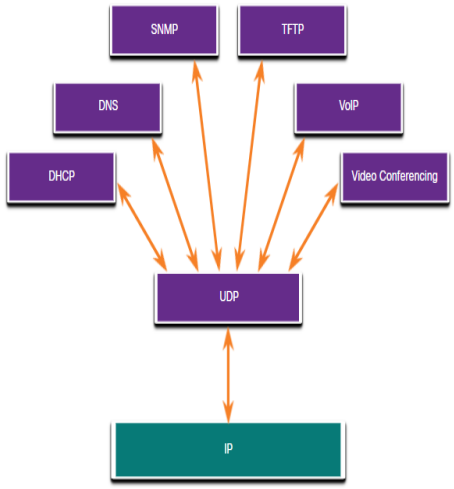
NETZWERKTECHNIK / SEMESTER 1 und 2

Felder des UDP Header

UDP Header Field	Description
Source Port	6-Bit-Feld, das zur Identifizierung der Quellanwendung anhand der Portnummer verwendet wird.
Destination Port	16-Bit-Feld, das zur Identifizierung der Zielanwendung anhand der Portnummer verwendet wird.
Length	16-Bit-Feld, das die Länge des UDP-Datagramm-Headers angibt.
Checksum	16-Bit-Feld, das für die Fehlerüberprüfung des Datagramm-Headers und der Daten verwendet wird.

Anwendungen die UDP nutzen

- **Live-Video- und Multimedia-Anwendungen** - Diese Anwendungen können einen gewissen Datenverlust tolerieren, erfordern jedoch nur eine geringe oder keine Verzögerung. Beispiele hierfür sind VoIP und Live-Streaming-Videos.
- **Einfache Anforderungs- und Antwortanwendungen**: Anwendungen mit einfachen Transaktionen, bei denen ein Host eine Anforderung sendet und möglicherweise keine Antwort erhält. Beispiele hierfür sind DNS und DHCP.
- **Anwendungen, die die Zuverlässigkeit selbst handhaben** - Unidirektionale Kommunikation, bei der Flusssteuerung, Fehlererkennung, Bestätigungen und Fehlerbehebung nicht erforderlich sind oder von der Anwendung verarbeitet werden können. Beispiele hierfür sind SNMP und TFTP.



tgm

04

Portnummern

NETZWERKTECHNIK / SEMESTER 1 und 2

Mehrere separate Kommunikationen

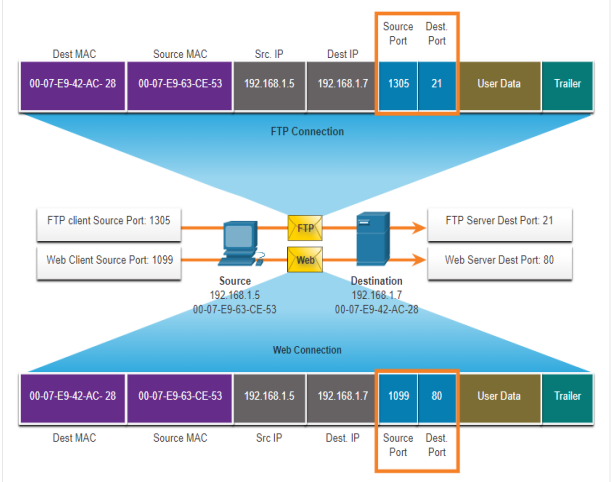
- TCP- und UDP-Transportschichtprotokolle verwenden **Portnummern, um mehrere, gleichzeitige Konversationen zu verwalten**.
- Die **Quellportnummer** ist mit der ursprünglichen Anwendung auf dem lokalen Host verknüpft, während die **Zielportnummer** mit der Zielanwendung auf dem Remotehost verknüpft ist.

24

NETZWERKTECHNIK / SEMESTER 1 und 2

Socket Paare

- Die Quell- und Zielports werden innerhalb des Segments platziert.
- Die Segmente werden dann in einem IP-Paket gekapselt.
- Die Kombination aus der **Quell-IP-Adresse und der Quellportnummer oder der Ziel-IP-Adresse und der Zielportnummer wird als Socket bezeichnet**.
- Sockets ermöglichen es **mehreren Prozessen, die auf einem Client ausgeführt werden, sich voneinander zu unterscheiden**, und mehreren Verbindungen zu einem Serverprozess, um voneinander zu unterscheiden.



25

Gruppen von Portnummern

Port Group	Number Range	Description
Well-known Ports	**0 to 1,023**	▪ Diese Portnummern sind für gängige oder beliebte Dienste und Anwendungen wie Webbrowser, E-Mail-Clients und RAS-Clients reserviert. ▪ Definierte bekannte Ports für gängige Serveranwendungen ermöglichen es Clients, den zugehörigen Dienst leicht zu identifizieren.
Registered Ports	**1,024 to 49,151**	▪ Diese Portnummern werden von der IANA einer anfordernden Entität zur Verwendung mit bestimmten Prozessen oder Anwendungen zugewiesen. ▪ Bei diesen Prozessen handelt es sich in erster Linie um einzelne Anwendungen, die ein Benutzer installiert hat, und nicht um allgemeine Anwendungen, die eine bekannte Portnummer erhalten. ▪ Cisco hat beispielsweise den Port 1812 für den Authentifizierungsprozess des RADIUS-Servers registriert.
Private and/or **Dynamic Ports**	**49,152 to 65,535**	▪ Diese Ports werden auch als ephemere Ports bezeichnet. ▪ Das Betriebssystem des Clients weist in der Regel dynamisch Portnummern zu, wenn eine Verbindung zu einem Dienst initiiert wird. ▪ Der dynamische Port wird dann verwendet, um die Clientanwendung während der Kommunikation zu identifizieren.

26

„Well-Known“ Portnummern

Port Number	Protocol	Application
20	TCP	File Transfer Protocol (FTP) - Data
21	TCP	File Transfer Protocol (FTP) - Control
22	TCP	Secure Shell (SSH)
23	TCP	Telnet
25	TCP	Simple Mail Transfer Protocol (SMTP)
53	UDP, TCP	Domain Name Service (DNS)
67	UDP	Dynamic Host Configuration Protocol (DHCP) - Server
68	UDP	Dynamic Host Configuration Protocol - Client
69	UDP	Trivial File Transfer Protocol (TFTP)
80	TCP	Hypertext Transfer Protocol (HTTP)
110	TCP	Post Office Protocol version 3 (POP3)
143	TCP	Internet Message Access Protocol (IMAP)
161	UDP	Simple Network Management Protocol (SNMP)
443	TCP	Hypertext Transfer Protocol Secure (HTTPS)

27

netstat Kommando

Unerklärliche TCP-Verbindungen können eine große Sicherheitsbedrohung darstellen. **netstat** ist ein wichtiges Werkzeug zur Überprüfung von Verbindungen.

```
C:\> netstat
Active Connections
Proto Local Address              Foreign Address                      State
TCP   192.168.1.124:3126         192.168.0.2:netbios-ssn          ESTABLISHED
TCP   192.168.1.124:3158         207.138.126.152:http
       ESTABLISHED
TCP   192.168.1.124:3159         207.138.126.169:http
       ESTABLISHED
TCP   192.168.1.124:3160         207.138.126.169:http
       ESTABLISHED
TCP   192.168.1.124:3161         sc.msn.com:http
       ESTABLISHED
TCP   192.168.1.124:3166         www.cisco.com:http               ESTABLISHED
```


28

05

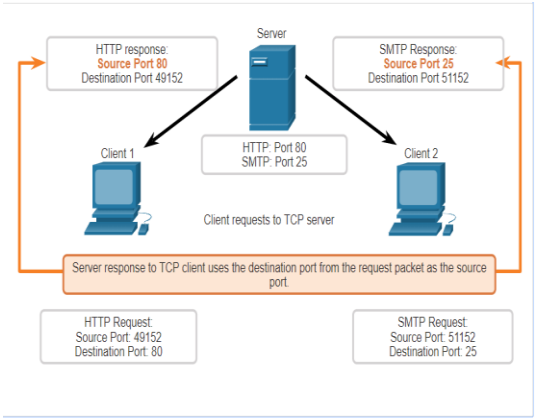
Verbindungsaufbau und Abbau

29

NETZWERKTECHNIK / SEMESTER 1 und 2

TCP Server Prozess

- **Jeder Anwendungsprozess, der auf einem Server ausgeführt wird, ist so konfiguriert, dass er eine Portnummer verwendet.**
- Einem einzelnen Server können nicht zwei Dienste zugewiesen werden, die derselben Portnummer innerhalb derselben Transportschichtdienste zugewiesen sind.
- Eine aktive Serveranwendung, die einem bestimmten Port zugewiesen ist, gilt als offen, was bedeutet, dass die Transportschicht an diesen Port adressierte Segmente akzeptiert und verarbeitet.
- Jede eingehende Clientanforderung, die an den richtigen Socket adressiert ist, wird akzeptiert, und die Daten werden an die Serveranwendung übergeben.

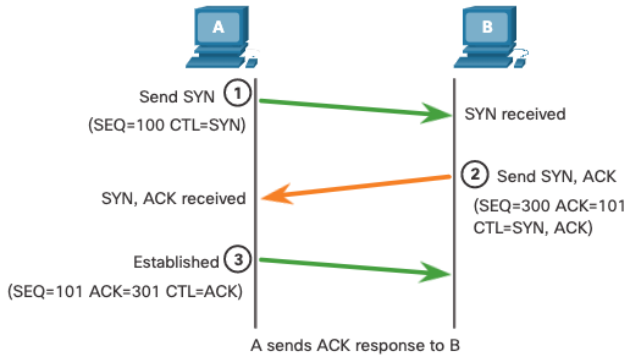


30

NETZWERKTECHNIK / SEMESTER 1 und 2

TCP Verbindungsaufbau

- **Schritt 1:** Der initiierende Client fordert eine Client-zu-Server-Kommunikationssitzung mit dem Server an.
- **Schritt 2:** Der Server bestätigt die Client-zu-Server-Kommunikationssitzung und fordert eine Server-zu-Client-Kommunikationssitzung an.
- **Schritt 3:** Der initiierende Client bestätigt die Server-zu-Client-Kommunikationssitzung.

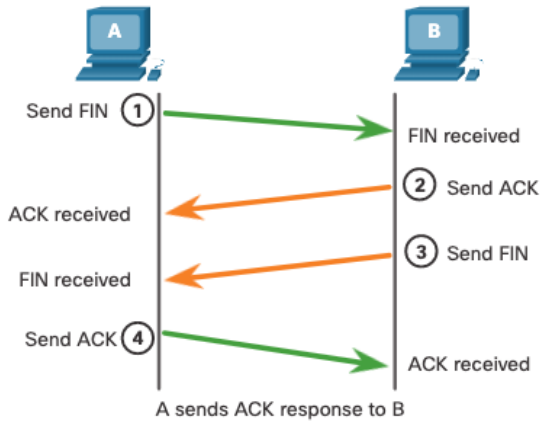


31

NETZWERKTECHNIK / SEMESTER 1 und 2

Session Termination

- **Schritt 1:** Wenn der Client keine Daten mehr im Stream senden kann, sendet er ein Segment mit dem FIN-Flag.
- **Schritt 2:** Der Server sendet eine ACK, um den Empfang der FIN zu bestätigen und die Sitzung vom Client zum Server zu beenden.
- **Schritt 3:** Der Server sendet eine FIN an den Client, um die Server-zu-Client-Sitzung zu beenden.
- **Schritt 4:** Der Client antwortet mit einer ACK, um die FIN vom Server zu bestätigen.



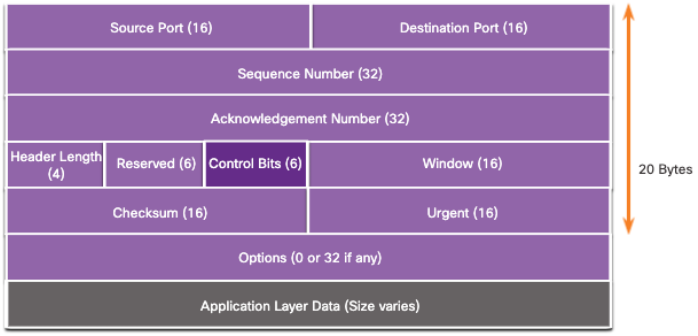
NETZWERKTECHNIK / SEMESTER 1 und 2

TCP 3-Way Handshake

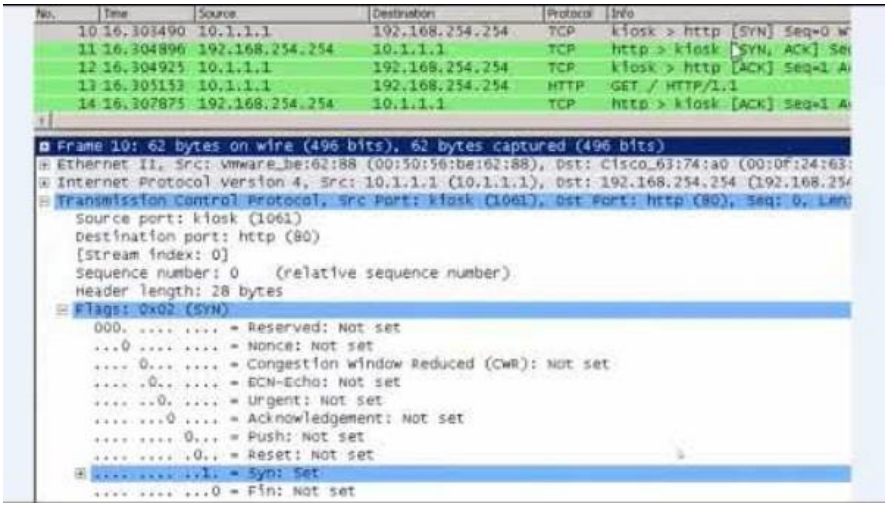
- Es stellt fest, dass das **Zielgerät im Netzwerk vorhanden** ist.
- Er überprüft, ob das **Zielgerät über einen aktiven Dienst verfügt** und Anforderungen an die Zielportnummer akzeptiert, die der initiierende Client verwenden möchte.
- Er informiert das Zielgerät, dass der Quellclient beabsichtigt, eine **Kommunikationssitzung über diese Portnummer** einzurichten.
- Nach Abschluss der Kommunikation werden die Sitzungen geschlossen und die Verbindung beendet. Die Verbindungs- und Sitzungsmechanismen ermöglichen die TCP-Zuverlässigkeitsfunktion.

TCP 3-Way Handshake Analyse

- **URG** - Urgent pointer field significant
- **ACK** - Acknowledgment flag used in connection establishment and session termination
- **PSH** - Push function
- **RST** - Reset the connection when an error or timeout occurs
- **SYN** - Synchronize sequence numbers used in connection establishment
- **FIN** - No more data from sender and used in session termination



Video - TCP 3-Way Handshake Analyse

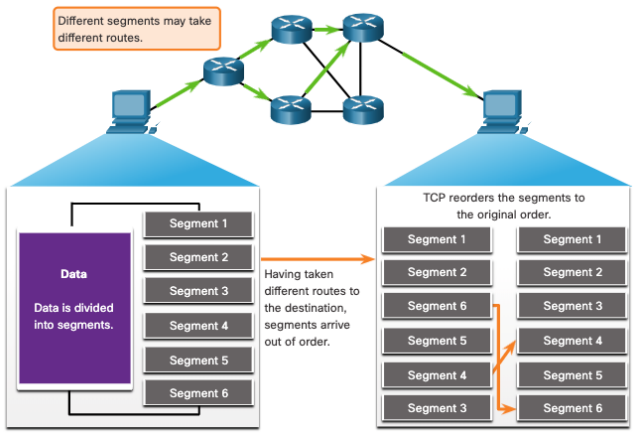


06

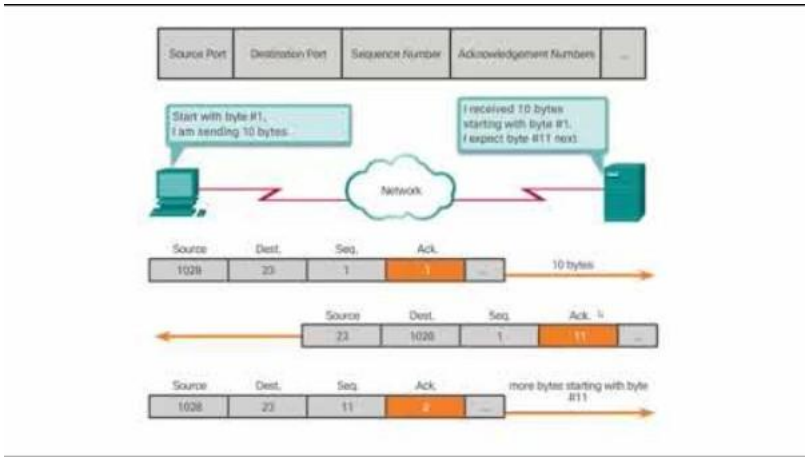
Zuverlässigkeit und Flußkontrolle

TCP-Zuverlässigkeit - Garantierte und geordnete Übertragung

- TCP kann auch dazu beitragen, den Paketfluss aufrechtzuerhalten, damit die Geräte nicht überlastet werden.
- Es kann vorkommen, dass TCP-Segmente nicht an ihrem Ziel ankommen oder nicht in der richtigen Reihenfolge ankommen.
- **Alle Daten müssen empfangen werden, und die Daten in diesen Segmenten müssen wieder in die ursprüngliche Reihenfolge eingefügt werden.**
- Um dieses Ziel zu erreichen, werden im Header jedes Pakets **Sequenznummern** zugewiesen.

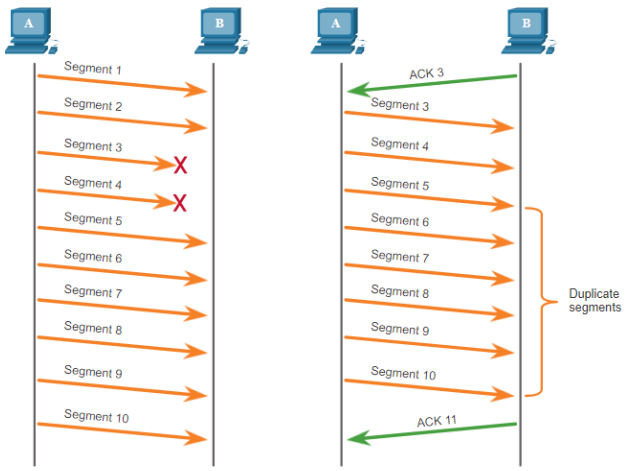


Video - TCP Reliability - Sequence Numbers und Acknowledgments



TCP Reliability – Data Loss und Retransmission

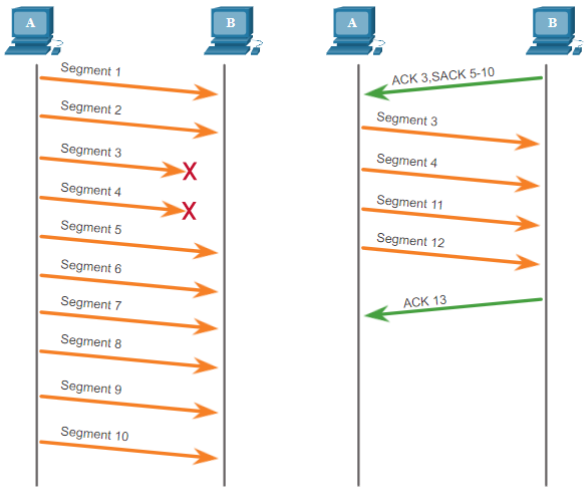
- Gelegentlich kommt es zu **Datenverlusten**.
- TCP bietet Methoden zur Verwaltung dieser Segmentverluste.
- Dazu gehört ein Mechanismus zur **erneuten Übertragung von Segmenten für nicht bestätigte Daten**.



NETZWERKTECHNIK / SEMESTER 1 und 2

TCP Reliability – Data Loss and Retransmission

- Hostbetriebssysteme verwenden heute in der Regel eine optionale TCP-Funktion namens **Selective Conknowledgment (SACK)**, die während des Drei-Wege-Handshakes ausgehandelt wird.
- Wenn beide Hosts SACK unterstützen, kann der Empfänger explizit bestätigen, welche Segmente (Bytes) empfangen wurden, einschließlich etwaiger diskontinuierlicher Segmente.

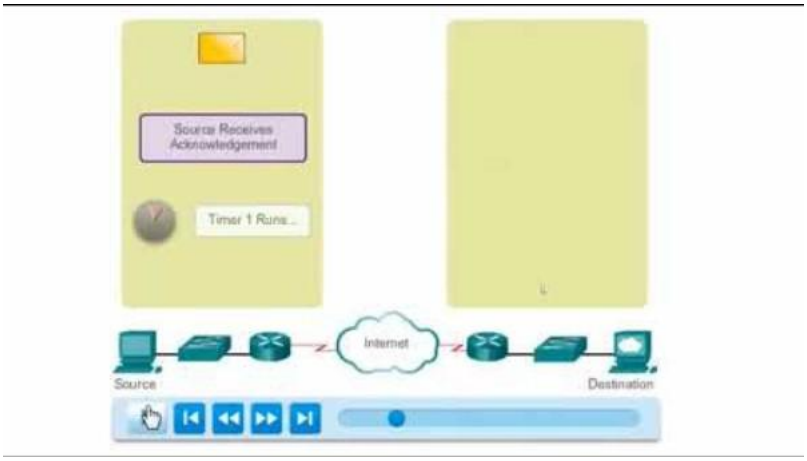


tgm

40

NETZWERKTECHNIK / SEMESTER 1 und 2

Video - TCP Reliability – Data Loss und Retransmission



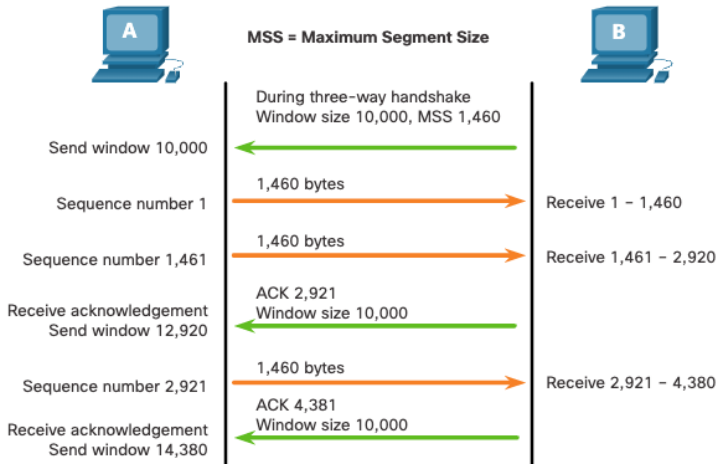
tgm

41

NETZWERKTECHNIK / SEMESTER 1 und 2

TCP Flow Control – Window Size und Acknowledgments

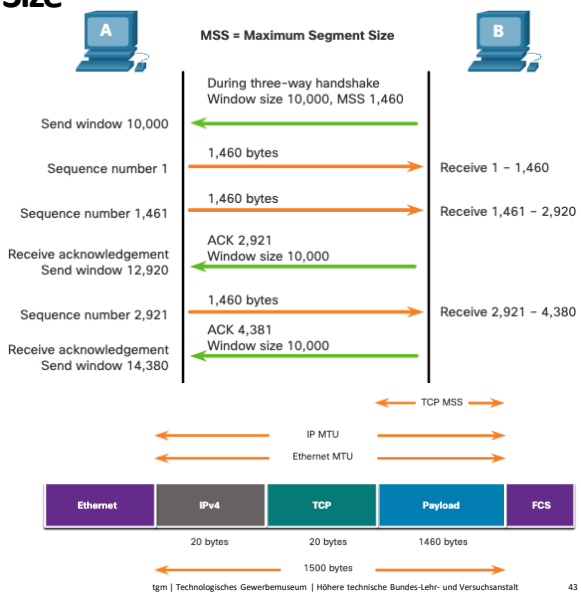
- **Die Flusssteuerung ist die Datenmenge, die das Ziel zuverlässig empfangen und verarbeiten kann.**
- Die Flusssteuerung trägt dazu bei, die Zuverlässigkeit der TCP-Übertragung aufrechtzuerhalten, indem die **Geschwindigkeit des Datenflusses zwischen Quelle und Ziel für eine bestimmte Sitzung angepasst** wird.



NETZWERKTECHNIK / SEMESTER 1 und 2

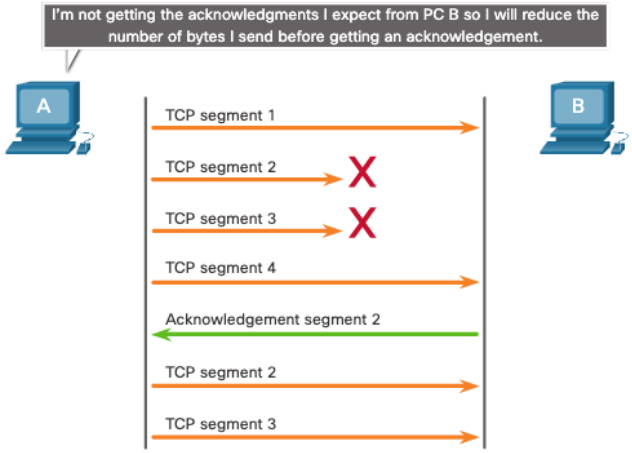
TCP Flow Control – Maximum Segment Size

- Die maximale Segmentgröße (MSS) ist die **maximale Datenmenge, die das Zielgerät empfangen kann**.
- Eine **gängige MSS-Datei ist 1.460 Byte** bei Verwendung von IPv4.
- Ein Host bestimmt den Wert seines MSS-Feldes, indem er die IP- und TCP-Header von der Ethernet Maximum Transmission Unit (MTU) subtrahiert, die standardmäßig 1500 Byte beträgt.
- 1500 minus 40 (20 Byte für den IPv4-Header und 20 Byte für den TCP-Header) lässt 1460 Byte übrig.



TCP Flow Control – Congestion Avoidance

- Wenn in einem Netzwerk eine Überlastung auftritt, führt dies dazu, dass Pakete vom überlasteten Router verworfen werden.
- Um Überlastungen zu vermeiden und zu kontrollieren, verwendet TCP mehrere Überlastungsbehandlungsmechanismen, Timer und Algorithmen.



44

07

UDP Kommunikation

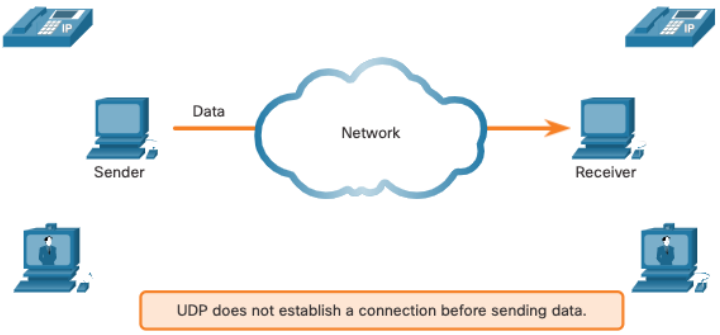
45

45

NETZWERKTECHNIK / SEMESTER 1 und 2

Geringer UDP-Overhead im Vergleich zu Zuverlässigkeit

UDP stellt keine Verbindung her. UDP bietet einen geringen Overhead für den Datentransport, da es über einen kleinen Datagramm-Header und keinen zusätzlichen Overhead für die Netzwerkverwaltung verfügt.

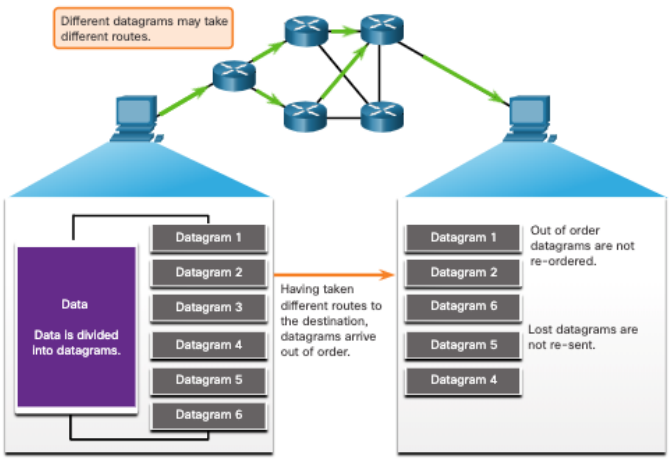


46

NETZWERKTECHNIK / SEMESTER 1 und 2

Zusammenbau von UDP-Datagrammen

- UDP verfolgt Sequenznummern nicht wie TCP.
- UDP hat keine Möglichkeit, die Datagramme in ihrer Übertragungsreihenfolge neu anzuordnen.
- UDP setzt die Daten einfach in der Reihenfolge des Empfangs wieder zusammen und leitet sie an die Anwendung weiter.

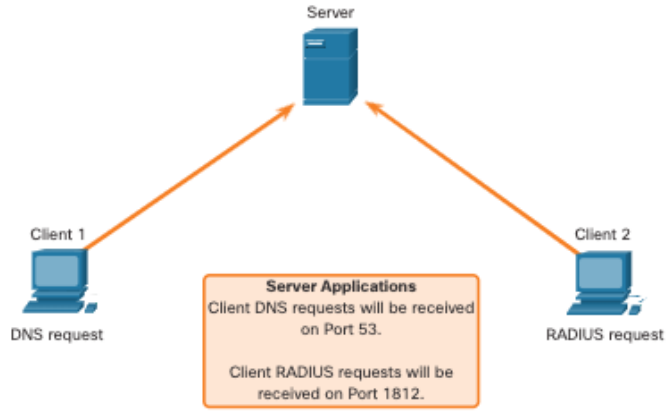


47

NETZWERKTECHNIK / SEMESTER 1 und 2

UDP-Serverprozesse und Anforderungen

- UDP-basierten Serveranwendungen werden bekannte oder **registrierte Portnummern** zugewiesen.
- UDP empfängt ein Datagramm, das für einen dieser Ports bestimmt ist, und leitet die Anwendungsdaten anhand seiner Portnummer an die entsprechende Anwendung weiter.

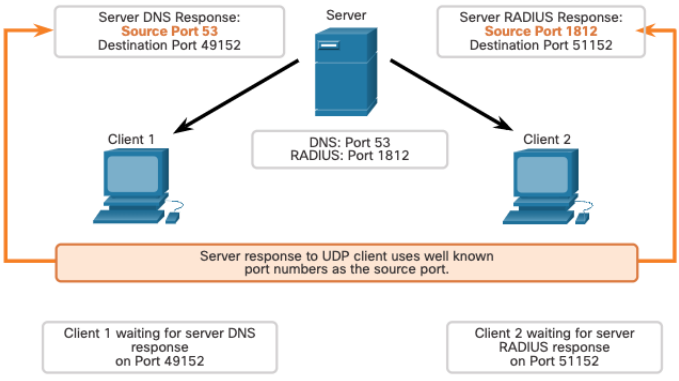


tgm

NETZWERKTECHNIK / SEMESTER 1 und 2

UDP-Client Prozesse

- Der UDP-Clientprozess wählt dynamisch eine Portnummer aus dem Bereich der Portnummern aus und verwendet diese als Quellport für die Konversation.
- Der Zielport ist in der Regel die bekannte oder registrierte Portnummer, die dem Serverprozess zugewiesen ist.
- Nachdem ein Client die Quell- und Zielports ausgewählt hat, wird dasselbe Portpaar im Header aller Datagramme in der Transaktion verwendet.



tgm