



# Modul 14: Transportschicht

Unterlagen für Instruktoen

Einführung in Netzwerke v7.0  
(ITN)



# Was in diesem Modul zu erwarten ist

- Um das Lernen zu erleichtern, können folgende Funktionen innerhalb der GUI in dieses Modul aufgenommen werden:

Funktion	Beschreibung
Animationen	Den Lernenden mit neuen Fertigkeiten und Konzepten in Kontakt zu bringen.
Videos	Den Lernenden mit neuen Fertigkeiten und Konzepten in Kontakt zu bringen.
Überprüfen Sie Ihr Verständnis (C heck Y our U nderstanding)	Pro Thema Online-Quiz, um Lernenden dabei zu helfen, das Verständnis von Inhalten zu messen.
Interaktive Aktivitäten	Eine Vielzahl von Formaten, die Lernenden dabei helfen, das Verständnis von Inhalten zu ermitteln.
Syntaxprüfer	Kleine Simulationen, die Lernenden der Cisco-Befehlszeile aussetzen, um Konfigurationsfähigkeiten zu üben.
PT-Aktivität	Simulations- und Modellierungsaktivitäten, die dazu dienen, Fähigkeiten zu erforschen, zu erwerben, zu stärken und zu erweitern.

# Was in diesem Modul zu erwarten ist (Fortsetzung)

- Um das Lernen zu erleichtern, können folgende Funktionen in diesem Modul enthalten sein:

Funktion	Beschreibung
Praxisorientierte Übungen	Labs für die Arbeit mit realer Hardware.
Schulungsaktivitäten	Diese finden Sie auf der Seite Instructor Resources. Klassenaktivitäten sollen das Lernen, die Unterrichtsdiskussion und die Zusammenarbeit erleichtern.
Modulquizzes	Selbsteinschätzungen, die Konzepte und Fähigkeiten integrieren, die in der Reihe von Themen des Moduls erlernt wurden.
Modulzusammenfassung	Kurze Zusammenfassung des Modulinhalts.



# Modul 14: Transportschicht

Einführung in Netzwerke v7.0  
(ITN)



# Modulziele

## Modultitel: Transportschicht

**Modulziel:** Vergleichen Sie die Operationen von Transportschichtprotokollen bei der Unterstützung der End-to-End-Kommunikation.

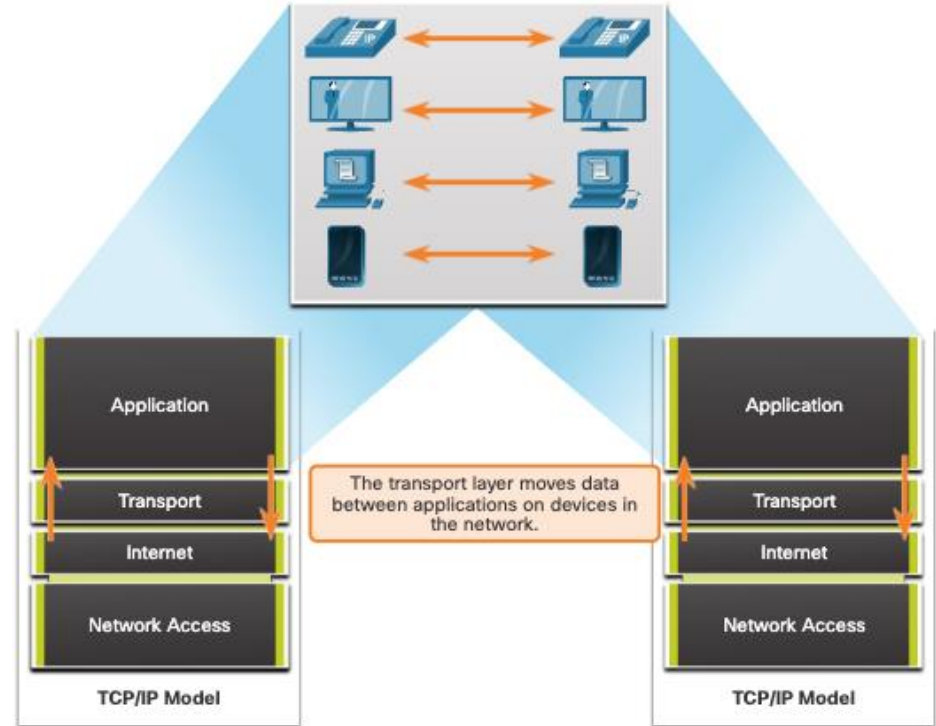
Thema	Ziel
Transport von Daten	Erläutern Sie den Zweck der Transportschicht bei der Verwaltung des Transports von Daten in der End-to-End-Kommunikation.
TCP Überblick	Erläutern Sie die Eigenschaften von TCP.
UDP Übersicht	Erläutern Sie die Merkmale von UDP.
Port-Nummern	Erläutern Sie, wie TCP und UDP Portnummern verwenden.
TCP-Kommunikationsprozess	Erläutern Sie, wie TCP-Sitzungseinrichtungs- und Abschlussprozesse eine zuverlässige Kommunikation ermöglichen.
Zuverlässigkeit und Flusskontrolle	Erläutern Sie, wie TCP-Protokolldateneinheiten übertragen und bestätigt werden, um die Lieferung zu gewährleisten.
UDP-Kommunikation	Vergleichen Sie die Vorgänge von Transportschichtprotokollen bei der Unterstützung der End-to-End-Kommunikation.

# 14.1 Transport von Daten

# Transport der Datenrolle der Transportschicht

Die Transportschicht ist:

- verantwortlich für die logische Kommunikation zwischen Anwendungen, die auf verschiedenen Hosts ausgeführt werden.
- Die Transportschicht bildet den Link zwischen der Anwendungsschicht und den unteren Schichten, die für die Netzwerkübertragung verantwortlich sind.

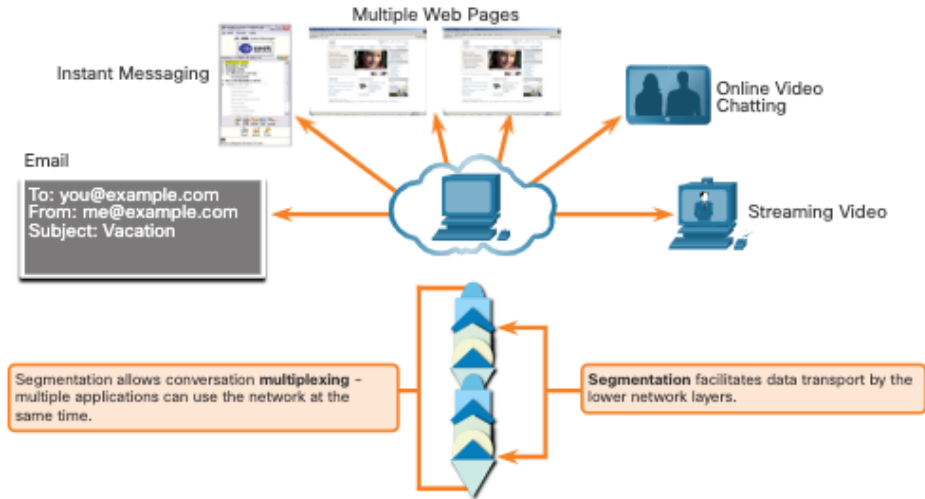


# Aufgaben der Transportschicht

## Zuständigkeiten der Transportschicht

Die Transportschicht hat folgende Zuständigkeiten:

- Verfolgung individueller Konversationen
- Segmentieren von Daten und erneutes Zusammensetzen zu Segmenten
- Fügt Header-Informationen hinzu
- Identifizieren, Separieren und Verwalten mehrerer Unterhaltungen
- Verwendet Segmentierung und Multiplexing, um verschiedene Kommunikationsgespräche im selben Netzwerk miteinander zu verbinden

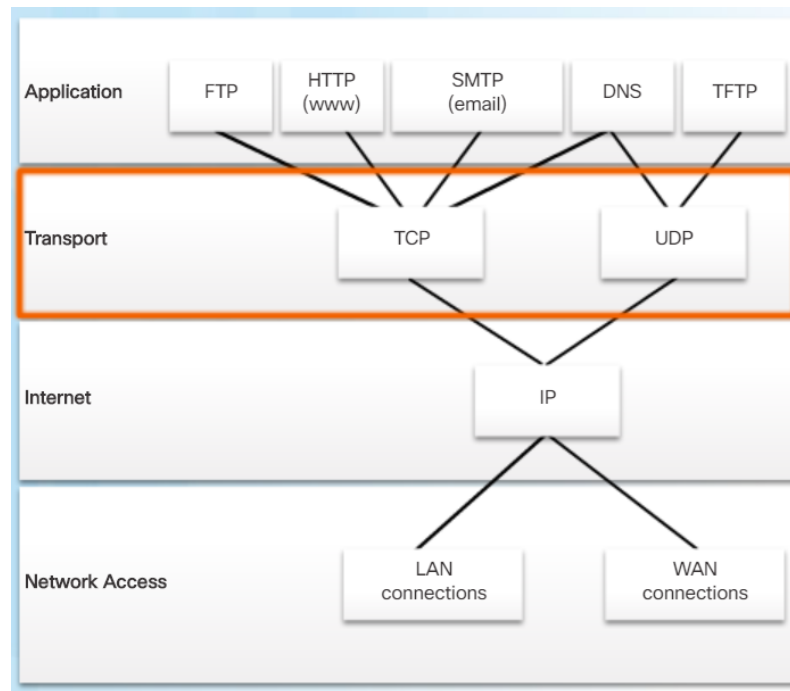




# Aufgaben der Transportschicht

## Zuständigkeiten der Transportschicht

- Es legt nicht fest, wie die Zustellung oder der Transport der Pakete erfolgt.
- Transportschichtprotokolle geben an, wie Nachrichten zwischen Hosts übertragen werden sollen, und sind für die Verwaltung der Zuverlässigkeitsanforderungen einer Unterhaltung verantwortlich.
- Die Transportschicht beinhaltet TCP- und UDP-Protokolle.

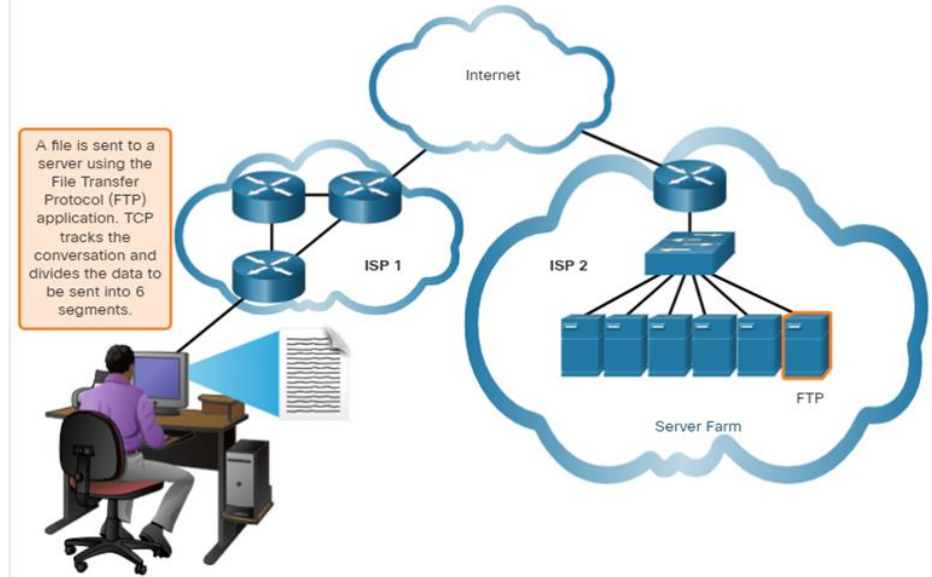


# Transport von Daten

## Transmission Control Protocol

TCP bietet Zuverlässigkeit und Flusskontrolle. TCP-Basisoperationen :

- Nummerierung und Nachverfolgung von Datensegmenten, die von einer bestimmten Anwendung an einen bestimmten Host übermittelt werden
- Bestätigung empfangener Daten
- Erneute Übertragung unquittierten Daten nach einem bestimmten Zeitraum
- Sequenzdaten, die in falscher Reihenfolge ankommen
- Sendet Daten mit einer effizienten Rate, die für den Empfänger akzeptabel ist

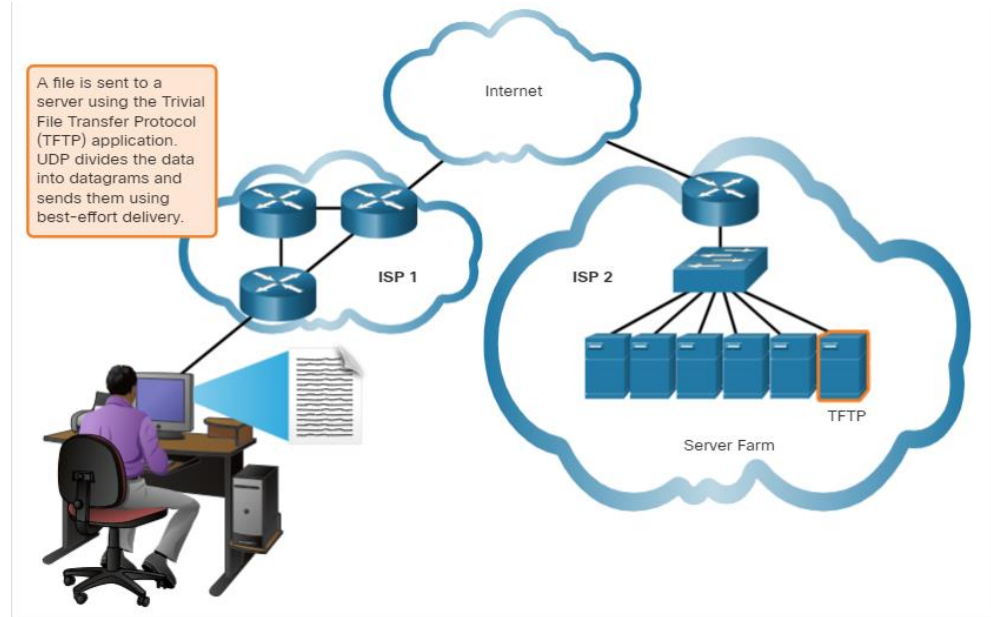


# Transport von Data

## User Datagram Protocol (UDP)

UDP bietet nur die Grundfunktionen für die Übertragung von Datensegmenten zwischen den entsprechenden Anwendungen, mit sehr geringem Overhead und nur einer geringfügigen Datenprüfung.

- UDP ist ein verbindungsloses Protokoll.
- UDP wird auch als best-effort delivery-Protocol bezeichnet, da es keine Bestätigung gibt, dass die Daten am Ziel empfangen wurden.



# Das richtige Transportschichtprotokoll für die richtige Anwendung

UDP wird auch von Anforderungs- und Antwort-Anwendungen verwendet, bei denen die Datenmengen minimal sind und das wiederholte Senden schnell erfolgen.

Wenn es wichtig ist, dass alle Daten eintreffen und in der richtigen Reihenfolge verarbeitet werden können, wird TCP als Transportprotokoll verwendet.

## UDP



VoIP  
(IP telephony)



DNS  
(Domain Name Resolution)

### Required protocol properties:

- Fast
- Low overhead
- Does not require acknowledgements
- Does not resend lost data
- Delivers data as it arrives

## TCP



SMTP/IMAP  
(Email)



HTTP/HTTPS  
(World Wide Web)

### Required protocol properties:

- Reliable
- Acknowledges data
- Resends lost data
- Delivers data in sequenced order

# 14.2 TCP — Übersicht

# TCP-Funktionen

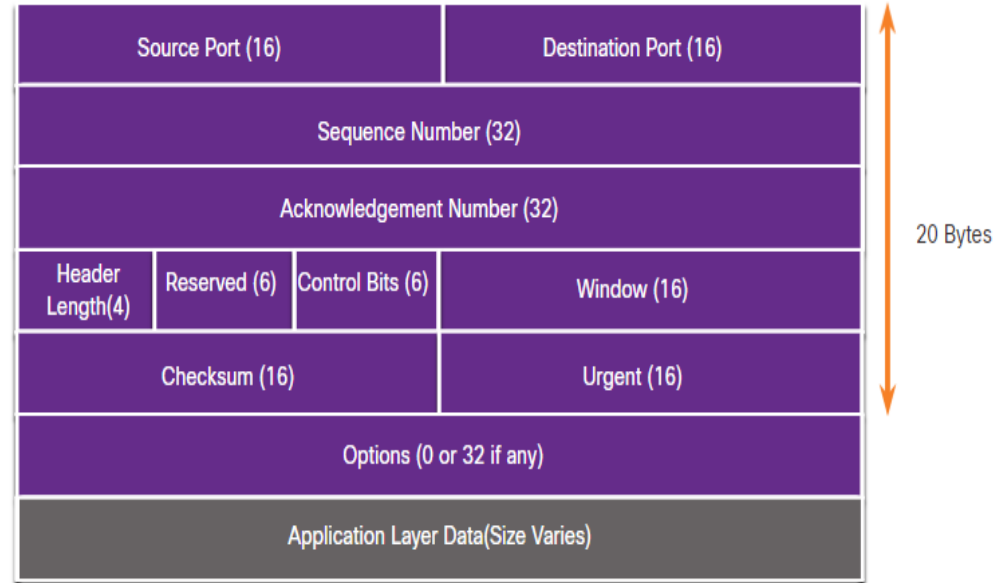
- **Einrichten einer Verbindung** - Ein verbindungsorientiertes Protokoll handelt eine permanente Verbindung (oder Sitzung) zwischen Quell- und Zielgeräten aus und baut diese auf, um Datenverkehr zu übertragen.
- **Zuverlässige Zustellung**- Es gibt viele Gründe dafür, warum ein Segment bei der Übertragung im Netzwerk beschädigt werden oder vollständig verloren gehen kann. TCP stellt sicher, dass jedes Segment, das von der Quelle gesendet wird, am Ziel ankommt.
- **Zustellung in derselben Reihenfolge** - Da Netzwerke mehrere Routen mit unterschiedlichen Übertragungsraten bereitstellen können, kommen Daten möglicherweise in der falschen Reihenfolge am Ziel an.
- **Flusskontrolle** - Netzwerk-Hosts verfügen über begrenzte Ressourcen, was Speicher oder Verarbeitungsleistung betrifft. Wenn TCP feststellt, dass diese Ressourcen überlastet sind, kann das Protokoll die sendende Anwendung dazu auffordern, die Datenflussrate zu reduzieren.

# TCP-Übersicht

## TCP-Header

TCP ist ein Stateful-Protokoll, was bedeutet, dass es den Status der Kommunikationssitzung verfolgt.

Um den Status einer Sitzung zu verfolgen, zeichnet TCP auf, welche Informationen gesendet und welche Informationen bestätigt wurden.



# TCP-Übersicht

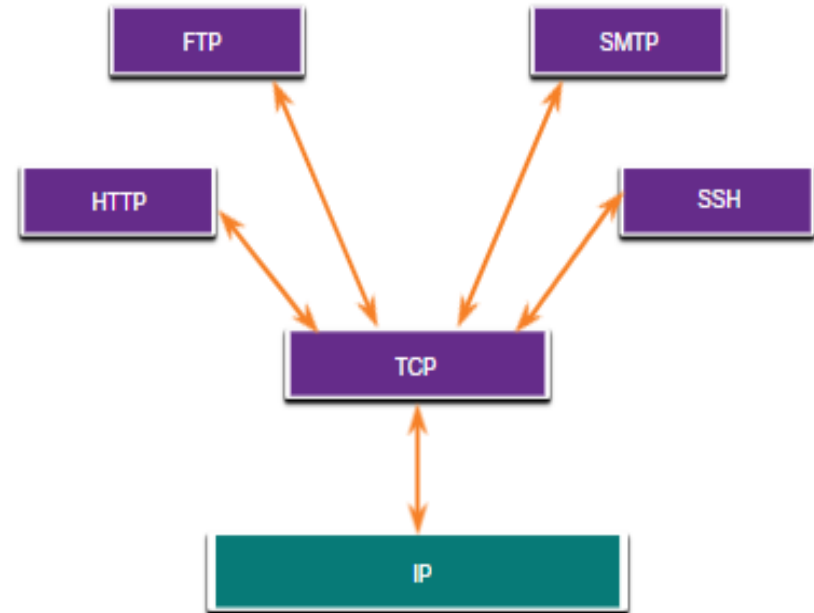
## TCP-Headerfelder

TCP-Header-Feld	Beschreibung
Quellport	Ein 16-Bit-Feld, das zur Identifizierung der Quellanwendung anhand der Portnummer verwendet wird.
Ziel-Port	Ein 16-Bit-Feld, das zur Identifizierung der Zielanwendung anhand der Portnummer verwendet wird.
Folgenummer	Ein 32-Bit-Feld, das für die Wiederausammenbauung von Daten verwendet wird.
Acknowledgement number (Bestätigungsnummer)	Ein 32-Bit-Feld, das verwendet wird, um anzuzeigen, dass Daten empfangen wurden und das nächste Byte von der Quelle erwartet wird.
Länge des Headers	Ein 4-Bit-Feld, das als „data offset-Wert“ bezeichnet wird, das die Länge des TCP-Segmentheaders angibt.
Reserviert	Ein 6-Bit-Feld, das für die zukünftige Verwendung reserviert ist.
Steuer-Bits	Control Bits (Steuer-Bits) (6 Bit) – Beinhalten Bit-Codes oder Flags, die den Zweck und die Funktion des TCP-Segments angeben.
Window size (Fenstergröße)	Ein 16-Bit-Feld, das verwendet wird, um die Anzahl der Bytes anzugeben, die vom Empfänger
Checksum (Prüfsumme)	Ein 16-Bit-Feld, das zur Fehlerüberprüfung des Segmentheaders und der Daten verwendet wird.
Dringend	Ein 16-Bit-Feld, das verwendet wird, um anzuzeigen, ob die enthaltenen Daten dringend sind.



# Anwendungen, die TCP verwenden

TCP verarbeitet alle Aufgaben zur Unterteilung des Datenstroms in Segmente und sorgt so für Zuverlässigkeit, die Steuerung des Datenflusses und das erneute Ordnen von Segmenten.



# 14.3 UDP Übersicht

# UDP-Übersicht

## UDP-Funktionen

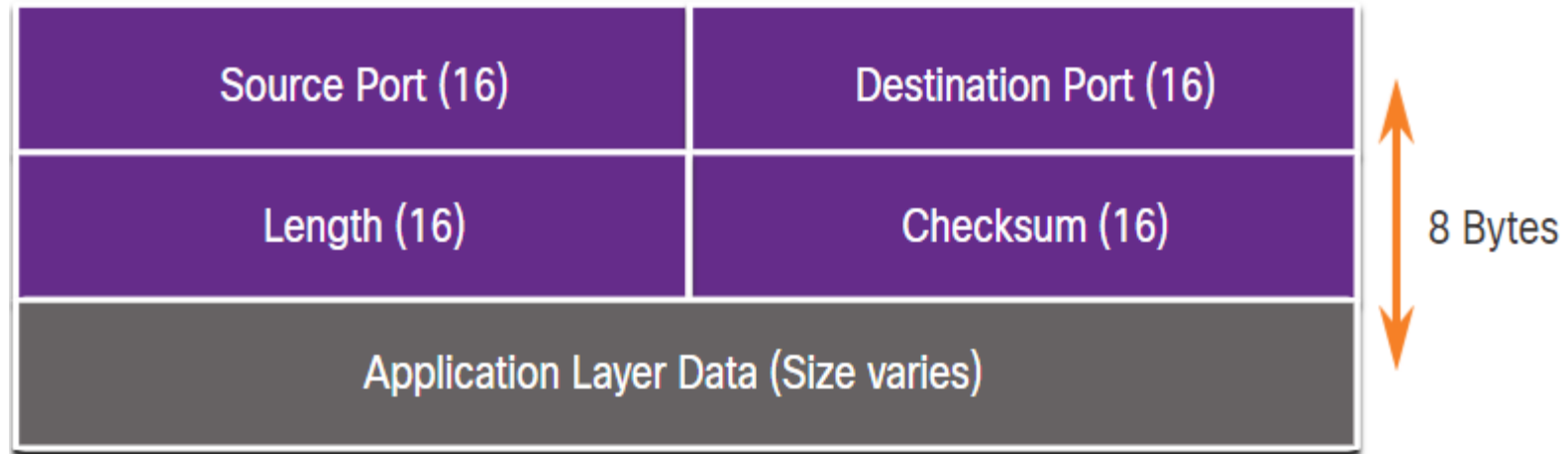
Zu den UDP-Funktionen gehören die folgenden:

- Die Daten werden in der Reihenfolge rekonstruiert, in der sie empfangen werden.
- Alle verlorenen Segmente werden nicht erneut versetzt.
- Es gibt keine Sitzungseinrichtung.
- Das Senden wird nicht über die Verfügbarkeit von Ressourcen informiert.

# UDP Übersicht

## UDP-Header

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



# UDP-Übersicht

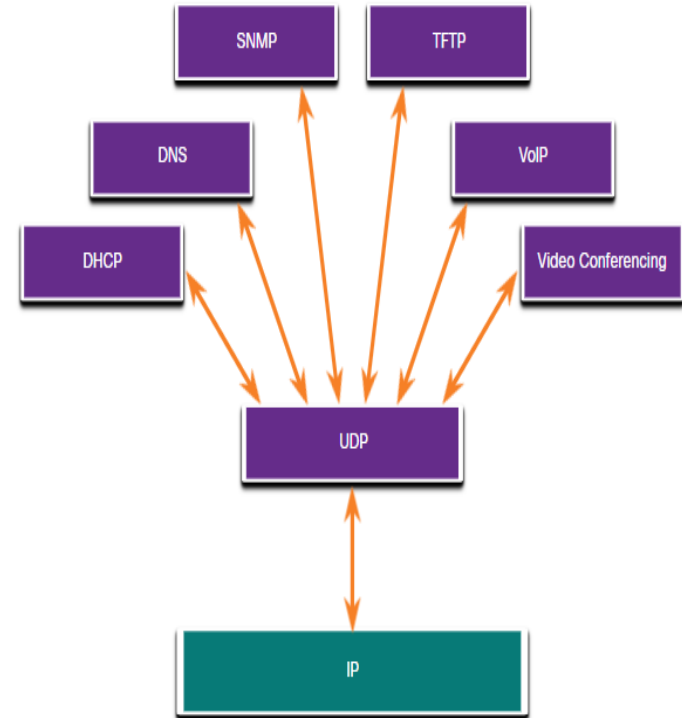
## UDP-Headerfelder

Die Tabelle identifiziert und beschreibt die vier Felder in einem UDP-Header.

UDP-Headerfeld	Beschreibung
Quellport	Ein 16-Bit-Feld, das zur Identifizierung der Quellanwendung anhand der Portnummer verwendet wird.
Ziel-Port	Ein 16-Bit-Feld, das zur Identifizierung der Zielanwendung anhand der Portnummer verwendet wird.
Länge	Ein 16-Bit-Feld, das die Länge des UDP-Datagram-Headers angibt.
Checksum (Prüfsumme)	Ein 16-Bit-Feld, das für die Fehlerüberprüfung des Datagram-Headers und der Daten verwendet wird.

# Anwendungen, die UDP verwenden

- Live-Video- und Multimedia-Anwendungen – Diese Anwendungen können einen Datenverlust tolerieren, lassen jedoch nur geringfügige oder keine Verzögerungen zu. Beispiele hierfür sind VoIP und Live-Video-Streaming.
- Einfache Anfrage- und Antwort-Anwendungen – Anwendungen mit einfachen Transaktionen, bei denen ein Host eine Anfrage sendet und eine Antwort erhalten kann oder auch nicht. Beispiele hierfür sind DNS und DHCP.
- Anwendungen, die selbst für Zuverlässigkeit sorgen – Unidirektionale Kommunikation, bei der Flusskontrolle, Fehlererkennung, Bestätigungen und Fehlerbehebung nicht erforderlich sind oder von der Anwendung übernommen werden können. Beispiele hierfür sind SNMP und TFTP.




# 14.4 Portnummern

# Mehrere getrennte Kommunikationen

Die Protokolle der TCP- und UDP-Transportschicht verwenden Portnummern, um mehrere gleichzeitige Gespräche zu verwalten.

Die Quellportnummer ist der Ursprungsanwendung auf dem lokalen Host zugeordnet, während die Zielportnummer der Zielanwendung auf dem Remote-Host zugeordnet ist.



Source Port (16)

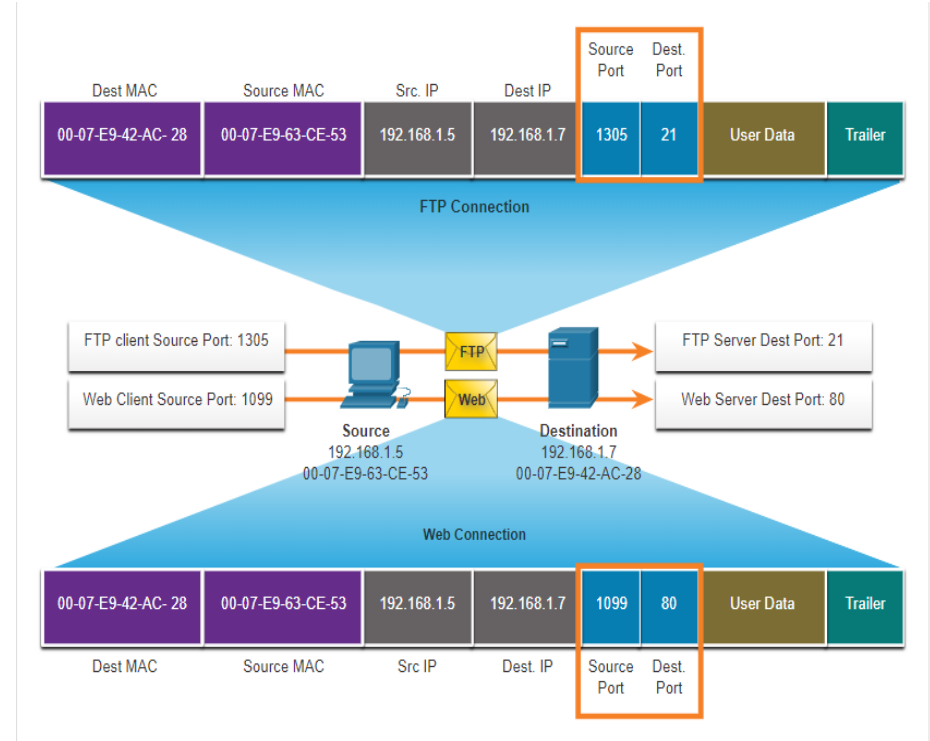
Destination Port (16)



# Anschlussnummern

## Sockelpaare

- Die Quell- und Zielports werden in das Segment eingefügt.
- Die Segmente werden dann in ein IP-Paket eingekapselt.
- Die Kombination aus Quell-IP-Adresse und Quellport-Nummer oder aus Ziel-IP-Adresse und Zielport-Nummer wird als Socket bezeichnet.
- Durch Sockets können mehrere Prozesse, die auf einem Client ausgeführt werden, sowie mehrere Verbindungen mit einem Serverprozess voneinander unterschieden werden.



# Port-Nummer-Gruppen

Port-Gruppe	Nummernbereich	Beschreibung
Well-Known-Ports	0 bis 1.023	<ul style="list-style-type: none"><li>•Diese Port-Nummern sind für allgemeine oder beliebte Dienste und Anwendungen wie Webbrowser, E-Mail-Clients und Remote-Zugriffs-Clients reserviert.</li><li>•Definierte bekannte Ports für allgemeine Serveranwendungen ermöglichen es Clients, den zugeordneten Dienst ganz einfach zu identifizieren.</li></ul>
Registrierte Ports	1.024 bis 49.151	<ul style="list-style-type: none"><li>•Diese Port-Nummern werden einem Antragsteller von der IANA zugewiesen, um sie für spezifische Prozesse oder Anwendungen zu verwenden.</li><li>•Diese Prozesse sind in erster Linie einzelne Anwendungen, die ein Benutzer anstelle gängiger Anwendungen installiert hat, die normalerweise eine Well-Known-Port-Nummer erhalten.</li><li>•Beispielsweise hat Cisco Port 1812 für den RADIUS-Serverauthentifizierungsprozess registriert.</li></ul>
Private und/oder dynamische Ports	49.152 bis 65.535	<ul style="list-style-type: none"><li>•Diese Ports werden auch als <i>flüchtige Ports</i> bezeichnet.</li><li>•Das Betriebssystem des Clients weist normalerweise Portnummern dynamisch zu, wenn eine Verbindung zu einem Dienst initiiert wird.</li><li>•Der dynamische Port wird dann verwendet, um die Client-Anwendung bei der Kommunikation zu identifizieren.</li></ul>

# Portnummern

## Portnummerngruppen (Fortsetzung)

### Well-Known-Port-Nummern

Port-Nummer	Protokoll	Anwendung
20	TCP	File Transfer Protocol (Daten)
21	TCP	File Transfer Protocol (Steuerung)
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)

## Port-Nummern

# Der Befehl netstat

Unbekannte TCP-Verbindungen können eine ernsthafte Sicherheitsbedrohung darstellen. Netstat ist ein wichtiges Werkzeug, um Verbindungen zu überprüfen.

```
C:\> netstat
Aktive Verbindungen
Proto Lokale Adresse Fremdadresse Status
TCP 192.168.1. 124:3126 192.168.0.2:netbios-ssn aufgebaut
TCP 192.168.1. 124:3158 207.138.126.152:http aufgebaut
TCP 192.168.1. 124:3159 207.138.126.169:http aufgebaut
TCP 192.168.1. 124:3160 207.138.126.169:http aufgebaut
TCP 192.168.1. 124:3161 sc.msn.com:http aufgebaut
TCP 192.168.1. 124:3166 www.cisco.com:http aufgebaut
```

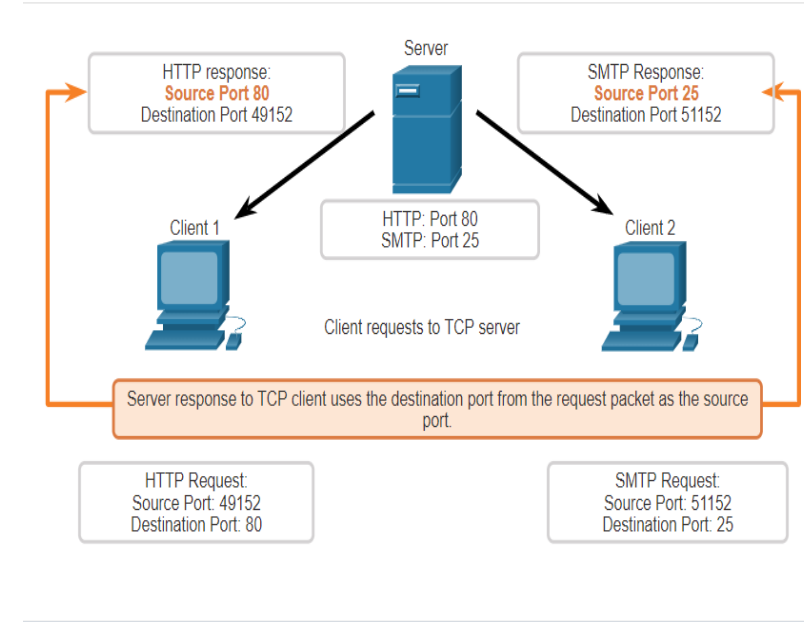
# 14.5 TCP- Kommunikationsprozess

# TCP-Kommunikationsprozess

## TCP Server-Prozesse

Jeder Anwendungsprozess, der auf einem Server ausgeführt wird, ist so konfiguriert, dass er eine Portnummer verwendet.

- Ein einzelner Server kann auf derselben Transportschicht nicht zwei Dienste aufweisen, denen die gleiche Port-Nummer zugewiesen ist.
- Eine aktive Serveranwendung, der ein bestimmter Port zugeordnet ist, gilt als offen, d. h., dass sie die Transportschicht-Segmente akzeptiert und verarbeitet, die an diesen Port adressiert sind.
- Jede eingehende Client-Anforderung, die an den richtigen Socket adressiert ist, wird akzeptiert und die Daten werden an die Serveranwendung übergeben.



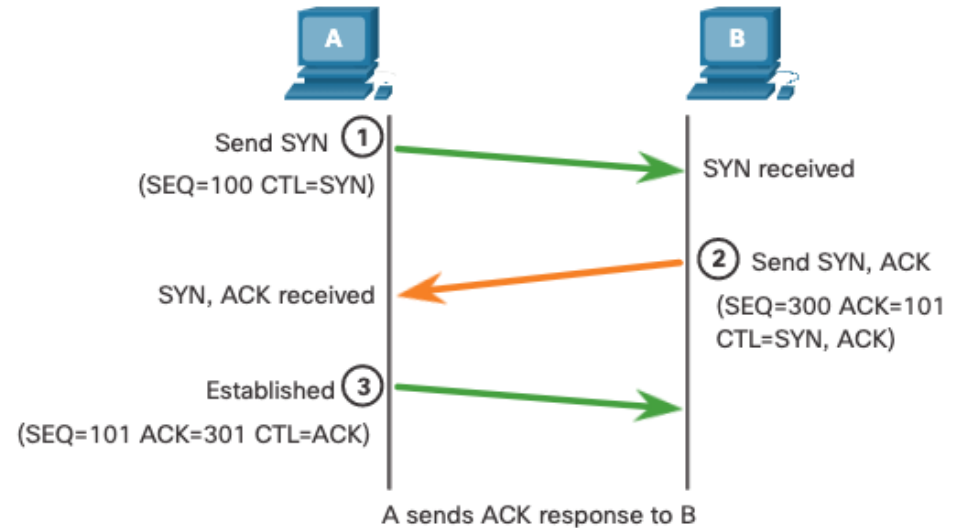
# TCP-Kommunikationsprozess

## TCP-Verbindungsaufbau

Schritt 1 – Der initiiierende Client fordert eine Client-Server-Kommunikationssitzung mit dem Server an.

Schritt 2: Der Server bestätigt die Client-Server-Kommunikationssitzung und fordert eine Server-Client-Kommunikationssitzung an.

Schritt 3 – Der initiiierende Client bestätigt die Server-Client-Kommunikationssitzung.



# TCP-Kommunikationsprozess

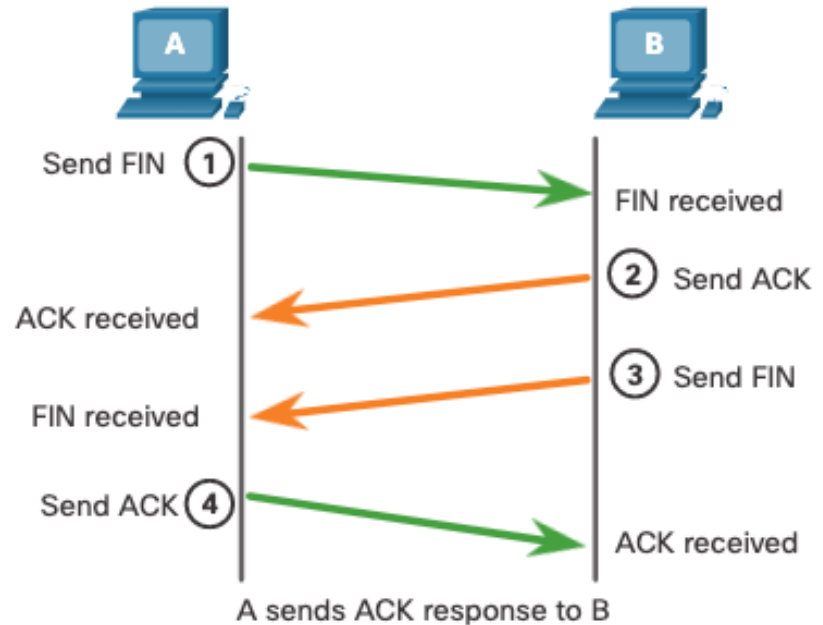
## Sitzungsbeendigung

Schritt 1: Wenn der Client keine Daten mehr im Stream senden muss, sendet er ein Segment mit dem FIN-Flag.

Schritt 2: Der Server sendet ein ACK-Flag, um den Erhalt des FIN-Flags zu bestätigen und die Client-Server-Sitzung zu beenden.

Schritt 3: Der Server sendet ein FIN-Flag an den Client, um die Server-Client-Sitzung zu beenden.

Schritt 4: Der Client antwortet mit einem ACK-Flag, um das FIN-Flag vom Server zu bestätigen.





# TCP-Drei-Wege-Handshake-Analyse

### Funktionen des Drei-Wege-Handshake:

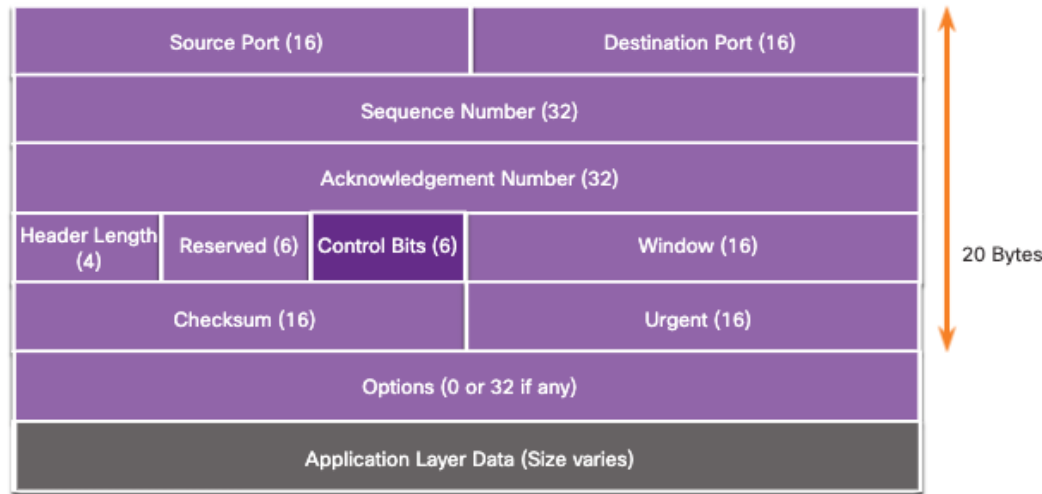
- Er stellt fest, ob das Zielgerät im Netzwerk vorhanden ist.
- Er überprüft, ob das Zielgerät über einen aktiven Dienst verfügt und Anforderungen auf der Zielport-Nummer akzeptiert, die der initiiierende Client zu verwenden beabsichtigt.
- informiert das Zielgerät, dass der Quell-Client beabsichtigt, eine Kommunikationssitzung mit dieser Port-Nummer herzustellen.

Nach Abschluss der Kommunikation werden die Sitzungen geschlossen und die Verbindung wird beendet. Die Verbindungs- und Sitzungsmechanismen ermöglichen die Zuverlässigkeitsfunktion von TCP.

## TCP-Dreiwege-Handshake-Analyse (Fortsetzung)

Die sechs Kontrollbit-Flags lauten wie folgt:

- **URG** - Dringendes Zeigerfeld signifikant
- **ACK** - Acknowledgment-Flag für Verbindungsaufbau und Sitzungsbeendigung
- **PSH** - Push-Funktion
- **RST** - dient dazu, eine Verbindung zurückzusetzen, wenn ein Fehler oder eine Zeitüberschreitung auftritt.
- **SYN** - Synchronisieren von Sequenznummern, die im Verbindungsaufbau verwendet werden
- **FIN** - Keine weiteren Daten mehr vom Absender und wird bei der Sitzungsbeendigung verwendet



# Video TCP 3-Wege-Handshake

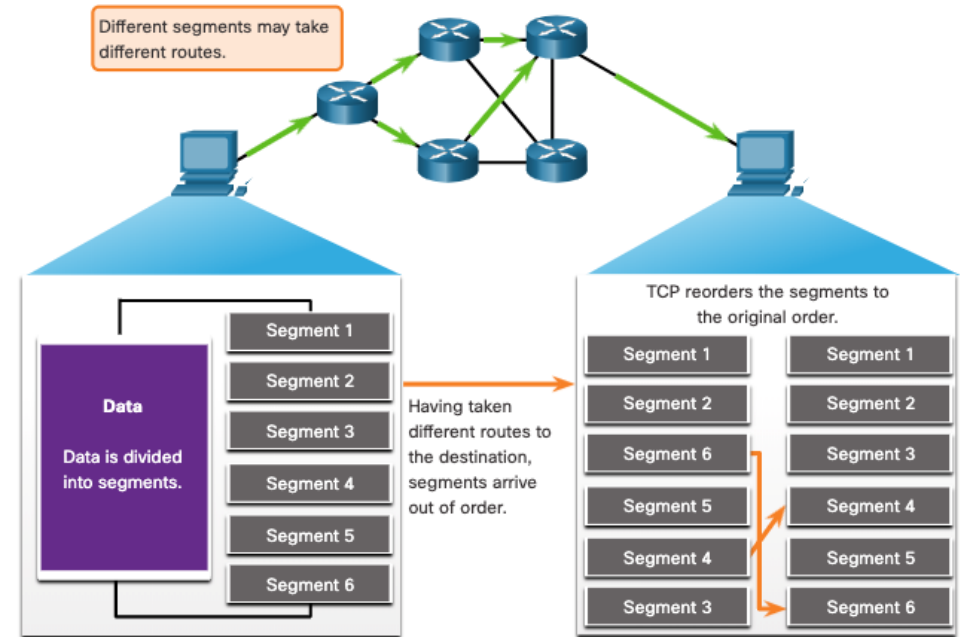
Das Video behandelt Folgendes:

- TCP-Drei-Wege-Handshakes
- Beendigung einer TCP-Konversation

# 14.6 Zuverlässigkeit und Flusskontrolle

# TCP Zuverlässigkeit - Garantierte und geordnete Zustellung

- TCP kann auch dazu beitragen, den Paketfluss aufrechtzuerhalten, damit Geräte nicht überlastet werden.
- Es kann vorkommen, dass TCP-Segmente nicht an ihrem Ziel ankommen.
- Alle Daten müssen empfangen werden und die Daten in diesen Segmenten müssen in die ursprüngliche Reihenfolge zusammengesetzt werden.
- Hierzu werden Sequenznummern im Header jedes Pakets zugewiesen.



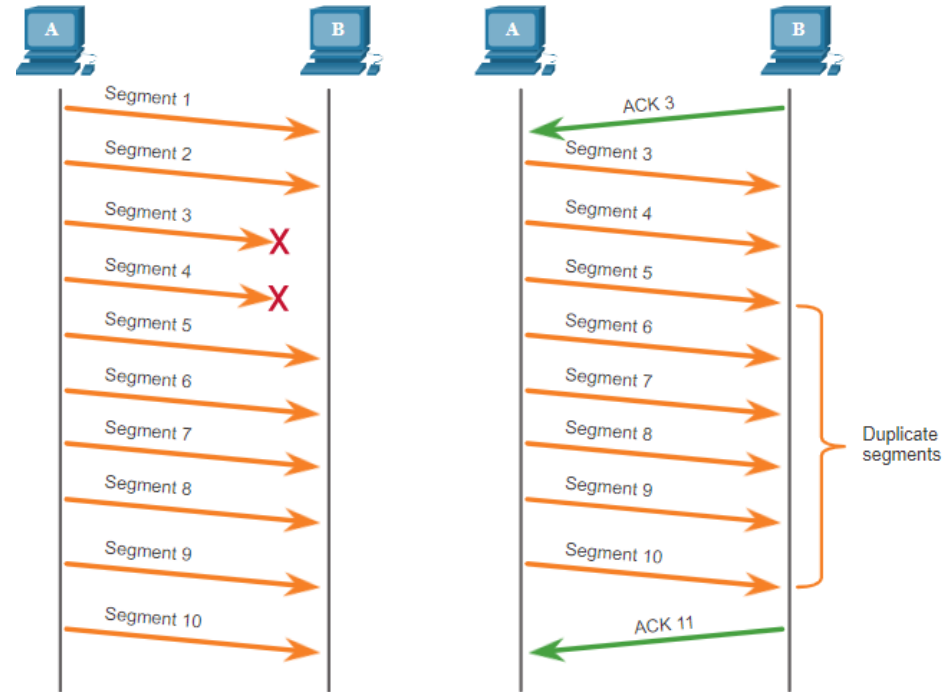
# Video – TCP-Zuverlässigkeit – Sequenznummern und Bestätigungen

Dieses Video zeigt ein vereinfachtes Beispiel für die TCP-Operationen .

# TCP Zuverlässigkeit — Datenverlust und Weiterleitung

Unabhängig davon, wie gut ein Netzwerk gestaltet ist, kommt es gelegentlich zu Datenverlusten.

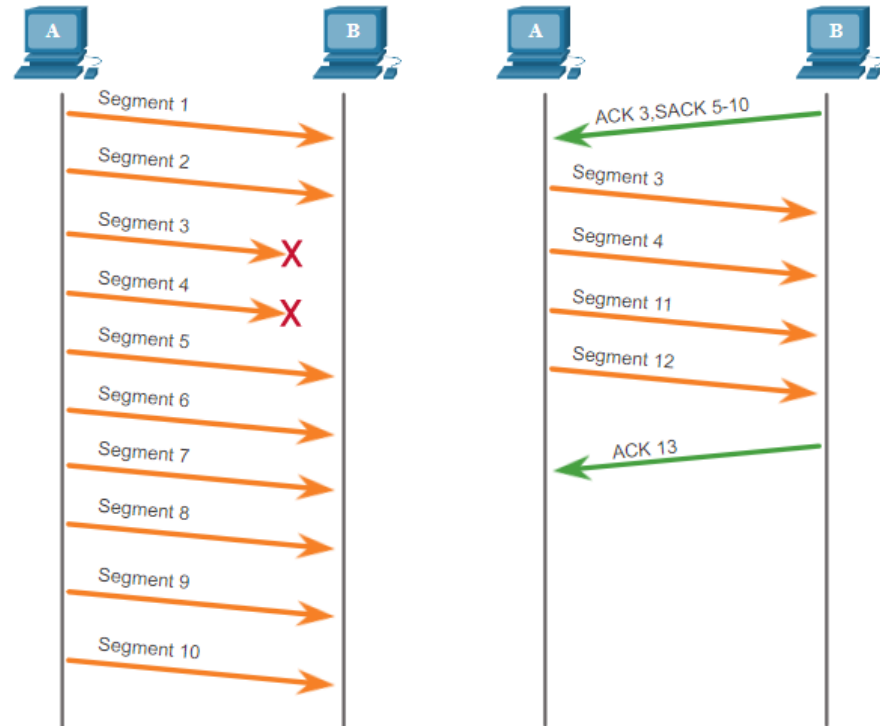
TCP bietet Methoden zur Verwaltung dieser Segmentverluste. Dazu gehört ein Mechanismus für die erneute Übertragung von Segmenten mit nicht bestätigten Daten.



# TCP Zuverlässigkeit — Datenverlust und Weiterübertragung (Fortsetzung)

Hostbetriebssysteme verwenden heute typischerweise eine optionale TCP-Funktion namens Selective Acknowledgment (SACK), die während des Dreiwege-Handshakes ausgehandelt wird.

Wenn beide Hosts SACK unterstützen, kann der Empfänger explizit bestätigen, welche Segmente (Bytes) empfangen wurden, einschließlich aller diskontinuierlichen Segmente.





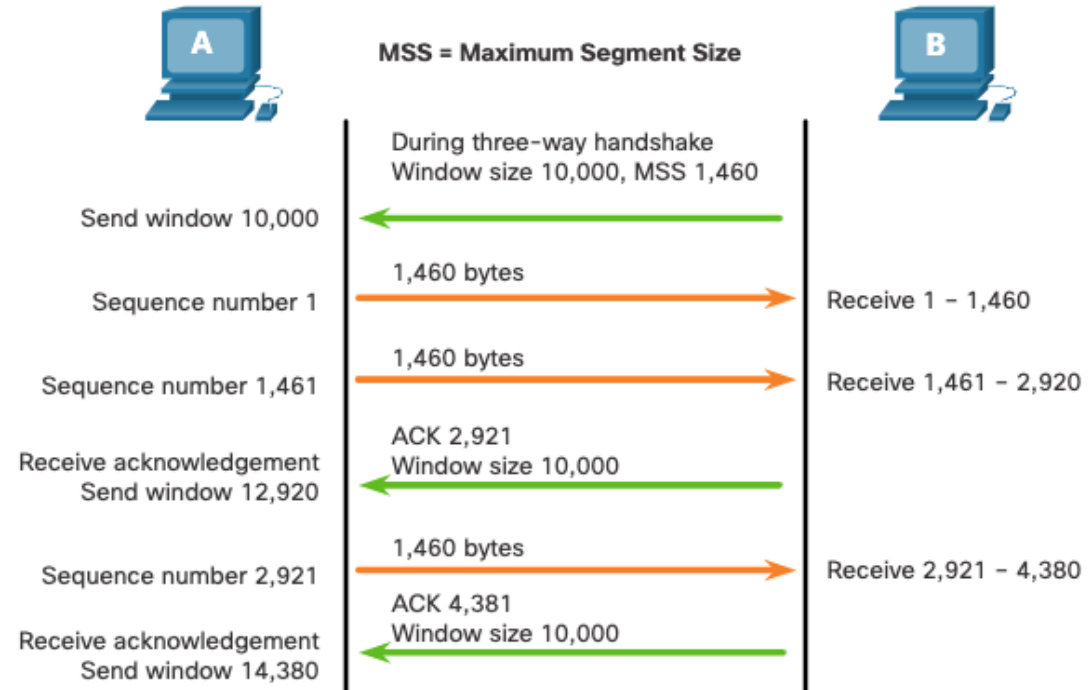
# TCP Zuverlässigkeit — Datenverlust und erneute Übertragung

Dieses Video zeigt den Prozess des erneuten Versenden von Segmenten, die ursprünglich nicht vom Ziel empfangen werden.

# TCP-Flusskontrolle – Fenstergröße und Bestätigungen

TCP bietet auch Mechanismen für die Flusssteuerung wie folgt:

- Flusssteuerung regelt die Datenmenge, die das Ziel zuverlässig empfangen und verarbeiten kann.
- Die Flusskontrolle trägt zur Zuverlässigkeit der TCP-Übertragung bei, da die Datenflussrate zwischen Quelle und Ziel für eine bestimmte Sitzung angepasst wird.

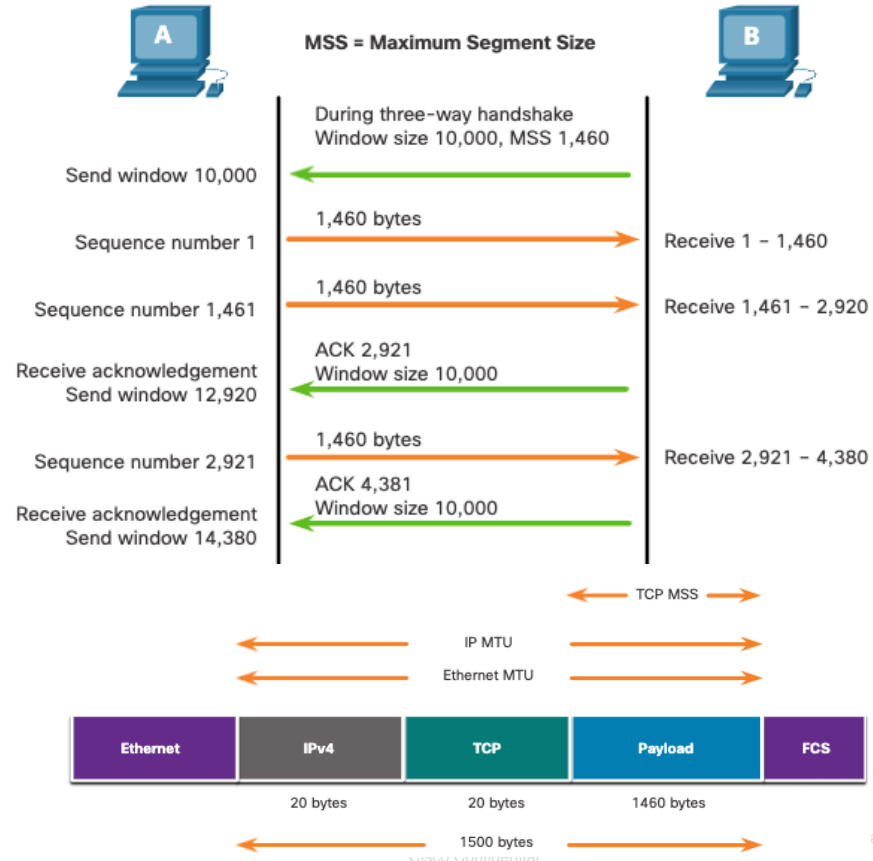


# Zuverlässigkeit und Durchflussregelung

## TCP Flusssteuerung — Maximale Segmentgröße

Maximale Segmentgröße (MSS) ist die maximale Datenmenge, die das Zielgerät empfangen kann.

- Ein gemeinsamer MSS ist 1.460 Bytes bei Verwendung von IPv4.
- Ein Host bestimmt den Wert seines MSS-Feldes, indem er die IP- und TCP-Header von der Ethernet-Maximalübertragungseinheit (MTU) subtrahiert, die standardmäßig 1500 Byte beträgt.
- 1500 minus 60 (20 Bytes für den IPv4-Header und 20 Bytes für den TCP-Header) belässt 1460 Bytes.

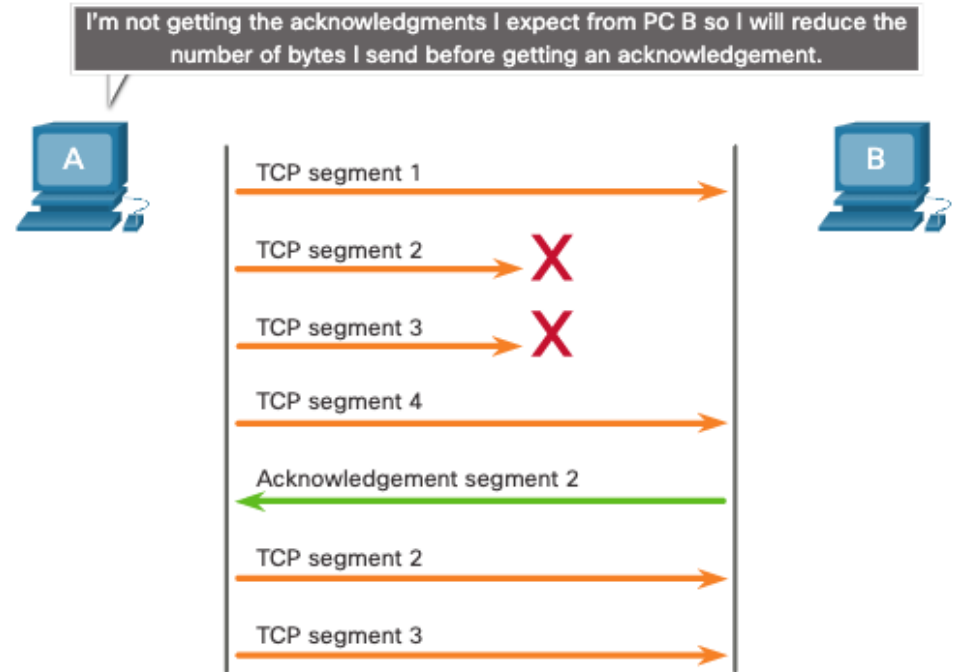


# Zuverlässigkeit und Durchflussregelung

## TCP-Flusssteuerung — Vermeidung von Überlastungen

Wenn in einem Netzwerk Überlastung auftritt, führt dies dazu, dass Pakete vom überbelasteten Router verworfen werden.

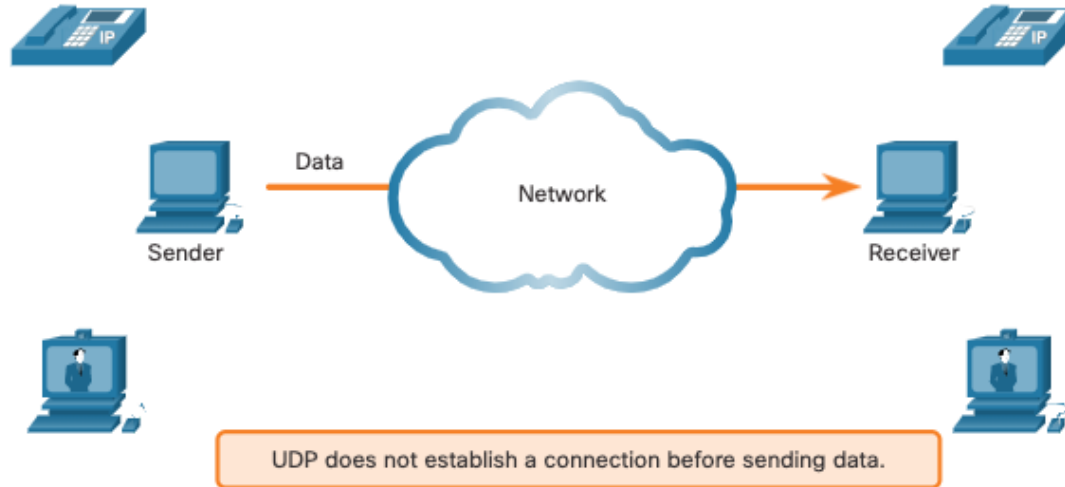
Zur Überlastungsvermeidung und -steuerung setzt TCP verschiedene Mechanismen, Timer und Algorithmen ein.



# 14.7 UDP-Kommunikation

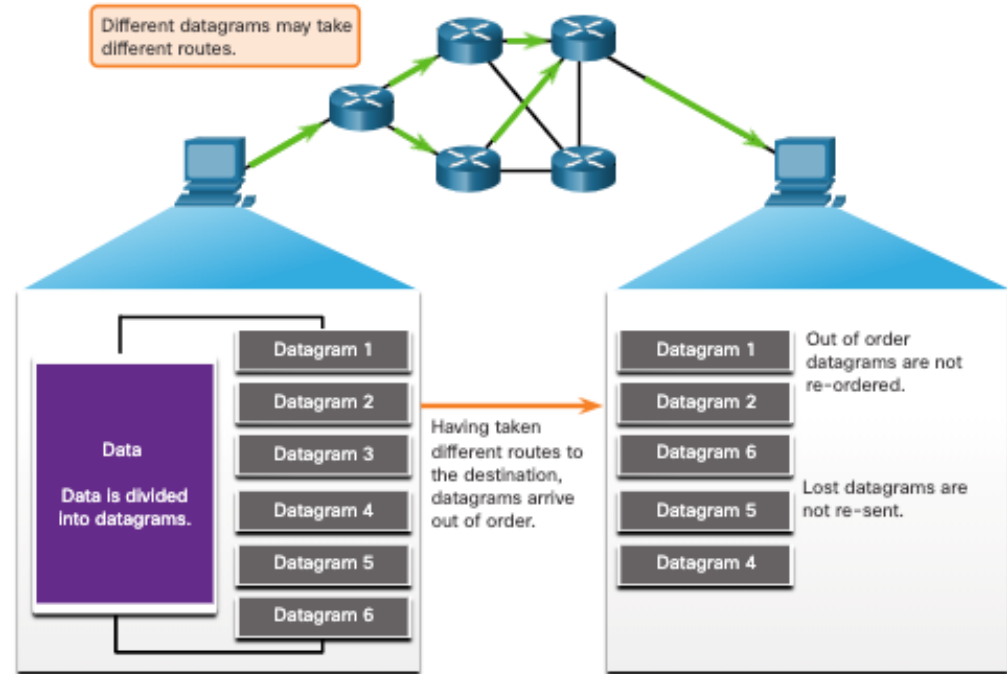
# UDP – Geringer Overhead und Zuverlässigkeit

UDP baut keine Verbindung auf. UDP bietet Datenübertragung mit geringem Overhead, da es einen kleinen Datagramm-Header und keinen Netzwerkmanagementverkehr hat.



# UDP – Zusammensetzung von Datagrammen

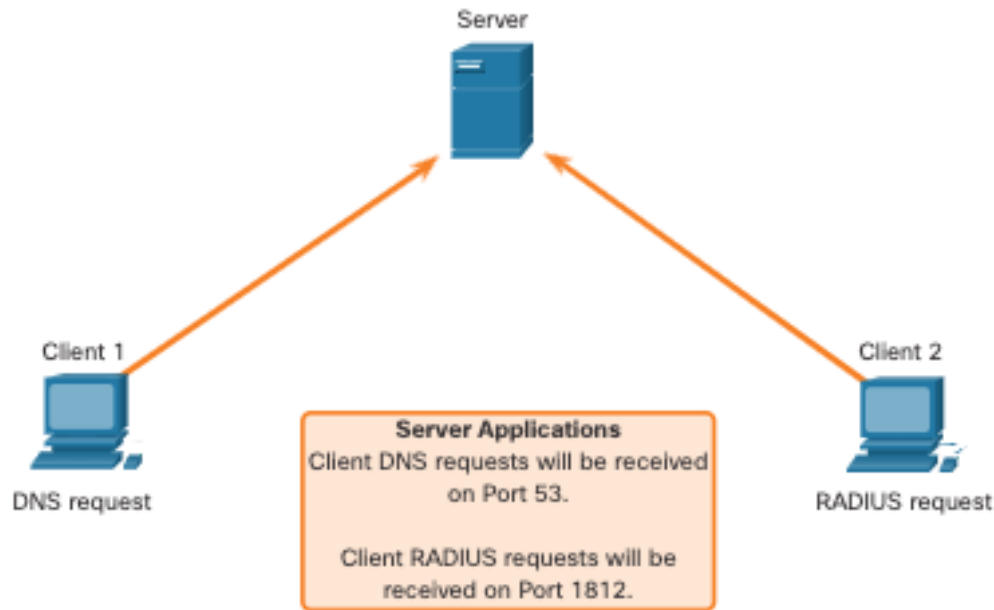
- UDP verfolgt keine Sequenznummern wie TCP.
- UDP hat keine Möglichkeit, die Datagramme in ihre Übertragungsreihenfolge umzuordnen.
- Daher setzt UDP die Daten einfach in der Reihenfolge zusammen, in der sie empfangen wurden, und leitet sie an die Anwendung weiter.



# UDP-Serverprozesse und -anfragen

Wie auch TCP-basierten Anwendungen werden UDP-basierten Serveranwendungen Well-Known- oder registrierte Port-Nummern zugeordnet, wie in der Abbildung gezeigt.

UDP erhält ein Datagramm, das für einen dieser Ports bestimmt ist, leitet es die Anwendungsdaten entsprechend der Port-Nummer an die passende Anwendung weiter.

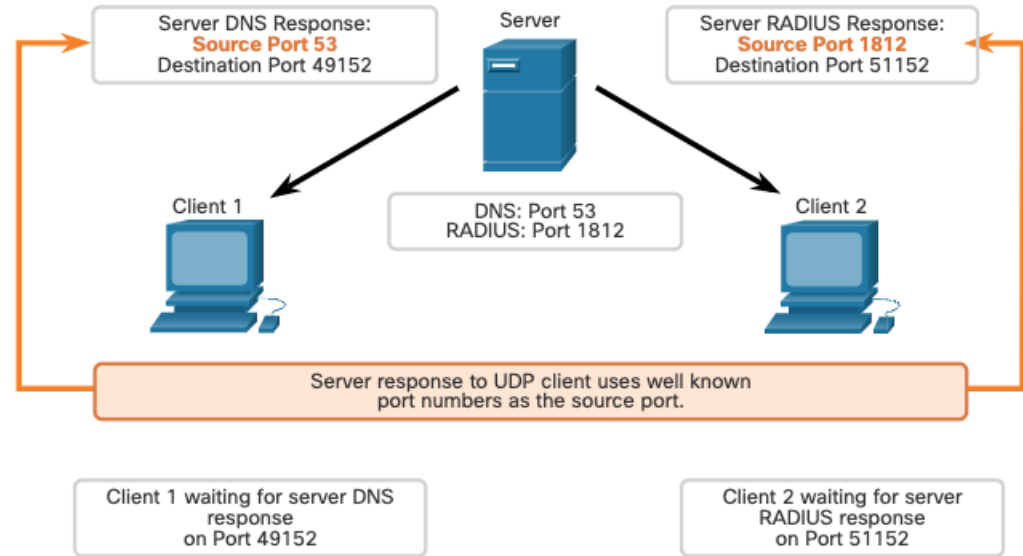




# UDP-Kommunikation

## UDP-Client-Prozesse

- Der UDP-Client-Prozess wählt dynamisch eine Port-Nummer aus dem Port-Nummernbereich aus und verwendet diese als Quellport für die Konversation.
- Der Zielport ist in der Regel eine dem Serverprozess zugeordnete Well-Known- oder registrierte Port-Nummer.
- Wenn ein Client die Quell- und Zielports ausgewählt hat, wird das gleiche Paar von Ports im Header aller Datagramme der Transaktion verwendet.



# 14.8 Modul Übung und Quiz

# Packet Tracer – TCP- und UDP-Kommunikation

In dieser Paket-Tracer-Übung werden Sie Folgendes tun:

- Generieren von Netzwerkverkehr im Simulationsmodus
- Untersuchen Sie die Funktionalität der TCP- und UDP-Protokolle.

# Was habe ich in diesem Modul gelernt?

- Die Transportschicht bildet die Verbindung zwischen der Anwendungsschicht und den unteren Schichten, die für die Netzwerkübertragung verantwortlich sind.
- Die Transportschicht umfasst zwei Protokolle: TCP und UDP.
- TCP erstellt Sitzungen, sorgt für Zuverlässigkeit, liefert die Lieferung in der gleichen Reihenfolge und unterstützt die Flusssteuerung.
- UDP ist ein einfaches Protokoll, das die grundlegenden Transportschichtfunktionen bietet.
- UDP rekonstruiert Daten in der Reihenfolge, in der sie empfangen werden, verlorene Segmente werden nicht erneut gesendet, keine Sitzungseinrichtung und UDP informiert den Absender nicht über die Ressourcenverfügbarkeit.
- Die Protokolle TCP- und UDP-Transportschicht verwenden Portnummern, um mehrere gleichzeitige Gespräche zu verwalten.
- Jeder Anwendungsprozess, der auf einem Server ausgeführt wird, ist so konfiguriert, dass er eine Portnummer verwendet.
- Die Portnummer wird entweder automatisch von einem Systemadministrator zugewiesen oder manuell konfiguriert.
- Damit die ursprüngliche Nachricht vom Empfänger verstanden werden kann, werden die Daten in diesen Segmenten wieder in der ursprünglichen Reihenfolge zusammengesetzt.

## Was habe ich in diesem Modul gelernt? (Forts.)

- Hierzu werden Sequenznummern im Header jedes Pakets zugewiesen.
- Die Flusskontrolle trägt zur Zuverlässigkeit der TCP-Übertragung bei, da die Datenflussrate zwischen Quelle und Ziel für eine bestimmte Sitzung angepasst wird.
- Eine Quelle übermittelt bis zu 1.460 Bytes Daten innerhalb jedes TCP-Segments . Dies ist der typische MSS, den ein Zielgerät empfangen kann.
- Der Prozess, bei dem das Ziel Bestätigungen sendet, wenn es erhaltene Byte verarbeitet, und die kontinuierliche Anpassung des Sendefensters der Quelle wird als Sliding Windows (verschiebbare Fenstergröße) bezeichnet.
- Zur Überlastungsvermeidung und -steuerung setzt TCP verschiedene Mechanismen, Timer und Algorithmen ein.

