Technische Informatik 4

Zusammenfassung des Vorlesungsstoffs

10. September 2008

Inhaltsverzeichnis

1 Einführung

1.1 Klassifikation von Kommunikationsnetzen

Kommunikationsart:

- Unicast (Punkt-zu-Punkt): Ein Sender, ein Empfänger
- Milticast (Punkt-zu-Mehrpunkt): Ein Sender, eine Gruppe von Empfängern
- Broadcast (Rundruf): An alle Teilnehmer des Netzes
- Anycast: An einen (den besten, nähesten gemaess dem Routing-Algorithmus) Teilnehmer

Übertragungsart:

- simplex: unidirektionale Verbindung
- halbduplex: bidirektionale Verbindung beim Umschalten, also nicht gleichzeitig in beide Richtungen
- (voll-)duplex: gleichzeitig in beide Richtungen
- Multiplexverfahren:
  - Frequenzmultiplex (Frequency Division Multiplex, FDM): Geräte verwenden verschiedene Teile des Frequenzspektrums
  - Zeitmultiplex (Time Division Multiplex, TDM): Geräte wechseln sich zeitlich ab
- Vermittlungsart:
  - Leitungsvermittlung:
    - Zwischen Sender und Empfänger wird mittels Signalisierung ein Kanal zur Übertragung aufgebaut (z.B. durch Zeit- oder Frequenzmultiplex)
    - Die zur Verfügung stehende Bitrate muß fest auf die Kanäle aufgeteilt werden
    - Standardverfahren in der Telefonie, bei schwankenden Telefonie Datenaufkommen mit vielen Pausen ineffizient
  - Paketvermittlung
    - Sender schickt Daten in Paketen, die einzeln zum Sender gelangen
    - die Bitrate wird effizienter aufgeteilt
    - kurzfristiges höheres Datenaufkommen kann über Puffer abgefangen werden
    - Dies kann zu Verzögerungen und Pufferüberläufen führen
    - Store and Forward: Das ganze Paket muss beim Router angekommen sein, bevor es auf die nächste Leitung geschickt wird
    - Verzögerungen: Verarbeitung am Router (Prüfung auf Bitfehler, Bestimmung des ausgehenden Links), Warteschlangenverzögerung (lastabhängig), Übertragungsverzögerung, Ausbreitungsverzögerung
    - $$R = BitratedesLinks(bps)$$

      ,

      $$L = Paketlaenge$$

,

$$Zeit, um Bits auf den Link zu senden = L/R$$

,

$$d = Laenge des Mediums$$

,

$$s = Ausbreitungsgeschwindigkeit$$

,

$$Ausbreitungsverzoegerung = d/s$$

Statisches Multiplexen Vergleicht man Paketvermittlung mit den bei der Leitungsvermittlung bekannten Multiplexverfahren, so erscheint diese wie statistisches Multiplexen

Übertragungsmedium

- leitungsgebunden: z.B. verdrillte Kupferdrähte, Glasfaser
- drahtlos: z.B. Funk, Infrarot

Netztopologien Bus (linear), Ring, Stern, Baum, vollständig vermascht

1.2 Protokolle

- wesentliches Strukturierungsprinzip
- legen Nachrichtenformat und Verhalten der Kommunikationspartner fest
- Beispiel: Hypertext Transfer Protocol (HTTP)

Schichtenarchitektur im Internet

- Anwendungsschicht (HTTP, FTP)
- Transportschicht (TCP, UDP)
- Netzwerkschicht (IP, Routing)
- Verbindungsschicht (Rahmen zwischen benachbarten Geräten)
- Physikalische Schicht (binäre Formate, Modulationsverfahren)

2 Anwendungsschicht

2.1 HTTP

- nicht-persistentes HTTP (Version 1.0): Einzelne TCP-Verbindung für jedes Objekt, Server beendet Verbindung sofort nach dem Senden
  1. Client initiiert TCP-Verbindung zu Server an Port 80
  2. Server wartet auf Verbindungen, erhält eine Anfrage vom Client und bestätigt diese
  3. Client sendet HTTP-Anfrage

4. Server sendet Antwort und schließt Verbindung
5. Client erhält Antwort und stellt diese dar
6. Falls eingebettete Objekte wird Schritt 1-5 wiederholt

- persistentes HTTP (Version 1.1): Server lässt Verbindung bestehen, alle Objekte werden über eine Verbindung gesendet
  - ohne Pipelining: nach jedem Objekt Anfrage für nächstes Objekt
  - mit Pipelining: eine Anfrage für alle eingebetteten Objekte

Conditional GET (Übung 1, Folie) Ziel: Objekt nicht neu übertragen, wenn der Client aktuelle Version im Cache hat (If-modified-since:, 304 Not Modified, 200 OK)

Authentifizierung HTTP ist zustandslos, deswegen muss Authentifizierung in jeder Anfrage erfolgen (WWW-Authenticate)

Cookies

- Möglichkeit Zustand des Clients trotz Zustandslosigkeit speichern
- HTTP-Request, Server sendet Cookie in Antwort an Client, Client speichert Cookie und sendet ihn bei folgenden Anfragen mit, Server kann so spezifisch mit den zugehörigen gespeicherten Daten antworten

Dynamische Inhalte Senden: In Rumpf von Anfragenachricht mit Post, häufig auch mit GET über URL (Schlüssel=Wert)

- Server-seitiges Scripting: im HTML ist Code eingebettet, der vom Server interpretiert wird und dabei HTML erzeugt , z.B. PHP
- Client-seitiges Scripting: im HTML ist Code eingebettet, der vom Client interpretiert wird, z.B. JavaScript

2.2 FTP

- Übertragung von Dateien zwischen Hosts
- Eine TCP-Verbindung (Port 21) zur Steuerung (USER, PASS, RETR, LIST, STOR, RETR, etc.), Authentifizierung, Ansehen des Inhalts eines entfernten Verzeichnisses
- Eine TCP-Verbindung (Port 20) zur Übertragung einer Datei, wird bei Befehl zur Dateiübertragung vom Server aufgebaut (Active Mode), wird nach Datenübertragung vom Server wieder beendet
- Active Mode: Client sendet vor Dateiübertragung *PORT N+1* und wartet auf TCP-Verbindung auf Port N+1 (Problem: Firewall)
- Passice Mode: Client sendet vor Dateiübertragung *PASV* an den Server. Server wartet auf eingehende Verbindungen auf beliebigem Port, welchen er dem Client in Antwort mitteilt.

2.3 E-Mail

- 3 Komponenten: Mail User Agent (MUA), Mail-Server, Simple Mail Transfer Protocol (SMTP über TCP, Port 25)
- 3 Phasen der Übertragung: Handshaking, Nachrichtenübertragung, Abschlussphase
- SMPT verwendet persistente Verbindung
- Nachrichtentext muss codiert (z.B. Base-64) werden, da einige Zeichen (CRLF.CRLF) nicht erlaubt sind
- Bei HTTP: Pull (Empfänger fordert von Server an). Jedes Objekt in eigener Antwort. Bei SMTP: Push (Sender sendet unaufgefordert an Server). Meherere Objekte werden in einer Multipart-Nachricht versendet
- Beide, HTTP und SMTP verwenden ASCII-Befehle und Statuscodes
- Zusätzliche Kopfzeilen bestimmen den MIME Content Type: Version, Datentyp (Text (plain, html), Image (jpeg, gif), etc.), Codierung
- POP3 zustandslos, Autorisierungsphase, Transaktionsphase, Port 110
- IMAP behält alle Nachrichten auf dem Server und behält Zustand über Sitzungen hinweg, Port 143

2.4 Netzwerkmanagement

- Simple Network Management Protocol (SNMP) (Anwendungsschicht, Folie 39
- Managing Entity: Rechner im Netz, Prozess = Client
- Managed Device: Gerät im Netz
- Managed Object: HW oder SW im Managed Device, z.B. Routing-Tabelle
- Management Agent: Prozess auf Managed Device (=Server)
- Kommunikation über UDP: GET (Anfrage einer Variable des Managed Objects), SET, GET-NEXT, GET-BULK, TRAP (ausgelöst von Managing Agent), etc.
- Management Information Base (MIB): MIB-Module enthalten Datenstrukturen für die Managed Objects

2.5 Domain Name Service (DNS)

- DNS bildet Domain-Namen auf Werte ab (u.A. IP-Adressen)
- Verteilte Datenbank, besteht aus vielen Name-Servern die über Anwendungsprotokoll kommunizieren
- Domain-Struktur: Hierarchischer Namensraum, Verwaltung von sog. *Zonen*
- Hierarchie: Root Name Server → Top-Level Domain Server (com,de,org,...) → ... → autoritativer Namesever (für Orgranisationen)

Resource Records

- Datensätze der Nameserver (Domainname, Wert, Typ, TTL (Time to Live))
- Typ A: Wert = IP-Adresse (ns.cisco.com, 128.123.234.2, A, TTL)
- Typ NS: Wert = Domainname eines Hosts (DNS-Server), der Namen in der Domain auflösen kann (princeton.edu, cit.princeton.edu, NS, TTL)
- Typ CNAME: Wert = Kanonischer Name eines Hosts, ermöglicht Aliasnamen
- Typ MX: Wert = Domainname eines Hosts, auf dem ein Mail-Server läuft
- NS-Typen sind verweise auf die nächsten Ebene

2.5.1 DNS-Protokoll

- iterativ: Antwort: Anderer Server, der Namen evtl. auflösen kann, NS- und A-Datensatz (Keine Speicherung der Information nötig). Angefragter Server muss selbst wieder andere Server fragen.
- rekursiv: Antwort: Auflösung des Namens, die u.U. von anderen Servern geholt wirt, A-Datensatz (Information muss gespeichert werden). Angefragter Server fragt selbst „weiter “andere Server
- Auch Kombination aus iterativ und rekursiv möglich

2.6 Content Distribution Networks (CDNs)

- Ziel: Vermeidung längerer Wartezeiten beim Laden von Webseiten
- Ähnlich wie Web-Caches, jedoch Spiegel-Server die den Inhalt im vornherein bekommen
- Geografisch verteilt

Verteilung der Anfragen

- Server-basierte HTTP Redirection: Server liefert aufgrund der IP-Adresse des Clients einen anderen geeigneten Server, Gefahr für Überlast des Servers + zusätzliche RTT
- Client-nahe HTTP Redirection: Z.B. durch Web-Proxy, schwierig
- DNS-basierte Redirection: DNS-Server bildet den Domain-Namen des Servers auf die IP-Adresse eines geeigneten Servers ab
- URL-Rewriting: Server liefert Basisseite und schreibt die URLs der eingebetteten Objekte um
- Meist Kombination aus DNS-basierter Redirection und URL-Rewriting
- Beispiel: HTTP-Anfrage, Antwort mit Basisseite die eingebettetes Objekt enthält (z.B. www.cdn.com/video.mpg). Anfrage an DNS-Server für www.cdn.com, der die IP-Adresse eines geeigneten Servers zurückliefert.

2.7 Real-Time Protocol (RTP)

Probleme bei Audio- und Video-Übertragung

- Verzögerung durch Verarbeitung (z.B. Kompression) und Laufzeit
- Veriablilität der Verzögerung
- Verluste verschlechtern Qualität
- Lösungsansatz: Streaming: Wiedergabepuffer um Schwankungen auszugleichen

Streaming

- Pakete werden periodisch gesendet (mit Sequenznummer und Zeitstempel)
- Beim Empfänger prefetching in Wiedergabepuffer
- Zeitstempel ermöglicht kontinuierliche Wiedergabe nach Verzögerung

RTP (Folie 66 ff)

- Verwendung zum Transport von Mulitimediadaten
- Üblicherweise UDP
- Keine Pufferung, Fehlerkontrolle, Zeitsynchronisation auf Endsystemen
- Keine Bestätigung und kein Sitzungsaufbau
- Philosophie: Überlässt möglichst viel der Anwendung
- Ünterstützung von Multicast

2.8 Session Initiation Protocol (SIP) (Folie 67 ff)

- Initialisierung einer Sitzung über IP-Netz, z.B. für Sprache (INVITE → 200 OK → ACK
- Medienströme werden danach unabhängig mittels RTP übertragen
- Suche eines Gesprächspartners über SIP-Server, dann direkte Verbindung zwischen beiden beteiligten Hosts

2.9 Socket-Programmierung

- UDP Verbindungslos
- Sender fügt für jedes Paket Ziel-IP-Adresse und Port hinzu
- Empfänger muss die Quell-IP-Adresse und den Quellport aus dem empfangenen Paket extrahieren
- Problem: Daten können in falscher Reihenfolge empfangen werden oder verloren gehen

2.10 Verteilte Systeme

- Besteht aus mehreren unabhängigen Systemen, die wie ein einzelnes kohärentes System erscheinen
- Oft realisiert durch Middleware, eine verteilte Software-Schicht, die zwischen Anwendungen und Betriebssystem angeordnet ist
- Beispiele für Middleware-Systeme: CORBA, J2EE, .NET, Web Services

Remote Procedure Call (RPC) Folie 80 ff

- Realisierung von verteilten Systemem über Sockets
- Mittels RPC werden Prozeduren auf entfernten Rechnern aufgerufen
- In der Programmiersprache und für den Benutzer sieht das wie ein normaler Prozeduraufruf aus → Transparenz
- RPC wird durch Client-Stub und Server-Stub realisiert
  1. Client-Prozess legt Parameter auf Stack
  2. Client Stub wandelt sie in Nachricht um und sendet sie an Server Stub
  3. Server Stub ruft Prozedur im Server-Prozess auf und sendet Ergebnis in Nachricht zurück
  4. Client Stub schreibt Ergebnis auf Stack oder in andere Speicherbereiche

Verteilte Objekte

- Objekte auf verschiedene Rechnern bieten Dienste an
- Remote Method Invocation (RMI): Wie RPC, für Methoden der Objekte
- Wird durch Vertreter der Objekte (Proxy) realisiert, die die Kommunikation über Primitiven des Betriebssystems ausführen

Anwendungsgebiete von Middleware-Systemen

- Kommunikation (Verdecken der Mechanismen des Netzwerks, Kodierung und Dekodierung gesendeter Daten)
- Prozesse (Threads, etc.)
- Namensdienste (Look-Up-Dienste, Namen für Objekte)
- Uhrensynchronisation, Fehlertoleranz, Sicherheit, etc.

Probleme von Middleware-Systemen: Komplexität, Performance-Overhead, Sicherheitsprobleme

2.10.1 Java-Überblick

- Remote Interface: Beschreibt die Funktionen, die auf dem Server zur Verfügung stehen
- Remote Object: Entferntes Objekt auf dem Server, implementiert das Remote Interface und das Verhalten der Methoden
- Remote Reference: Referenz auf Remote Object, bekommen die Clients von der RMI Registry

Ablauf (Folie 88)

1. Server registriert ein Remote Object bei der RMI Registry unter einem eindeutigen Namen
2. Client fragt mit diesem Namen die RMI Registry und bekommt Remote Reference
3. Client ruft Methode auf dieser Referenz auf
4. Server sendet Rückgabewerte oder Fehlermeldung (z.B. Verbindungsabbruch)

2.10.2 Web Services

Beispiel

- Temperatur Info Service, Flug Info Service, Karten Info Service
- Neuer Web Service (Reise Info Service): Ein Service, der einem mit dem Flugzeug erreichbare Reiseziele, nach Entfernungen geordnet, zusammen mit der aktuellen Temperatur auflistet.

XML Basistechnologien (Folie 94 ff)

- SOAP: Kommunikation zwischen den Services
  - SOAP-Envelope bestehen aus Header und Body
  - Informationen für den Empfänger stehen im Body
  - Header: Digitale Signatur, Routing Informationen, etc.
- UDDI: Dient zur Suche und Registrierung von Services (weltweit eindeutig, vgl. Telefonbuch)
  - White Pages: Namensregister, Auflistung der Anbieter, Kontaktinformationen
  - Yellow Pages: Branchenverzeichnis, verweist auf White Pages, klassifiziert Services
  - Green Pages: Informationen über Geschäftsmodell des Unternehmens, technische Details, etc.
- WSDL: Dient zur Beschreibung der angebotenen Dienste, Informationen für den Entwickler, Nachrichtenformat, Kommunikationsprotokoll, Schnitstellen, etc.

Zusammenhang

1. Web Service Provider beschreibt die Schnitstelle mittels WSDL
2. Web Service Provider registriert seinen Service bei der UDDI Registry
3. Web Service Requestor sucht bei der UDDI Registry nach gewünschtem Service und erhält Informationen wo er das WSDL-Dokument des Service findet
4. Web Service Requestor wertet die Informationen aus (XML) und generiert eine SOAP-Nachricht
5. Web Service Requestor bezieht gewünschte Informationen vom Service Provider

2.11 Peer-to-Peer-Systeme (Folie 106 ff)

- Grundidee: Inhalte nicht nur von zentralem Server, sondern auch von anderen Peers

Napster

- Peer informiert zentralen Server über sine IP und seine Inhalte
- Suche von Inhalten über zentralen Server
- Dateiübertragung direkt zwischen Peers
- Zentraler Server ist Leistungs- und Zuverlässigkeitsproblem

Gnutella

- Anfragender Peer sendet Anfrage an alle Nachbarn
- Wenn diese die Anfrage nicht beantworten können wird die Anfrage an die Nachbarn weitergeleitet (Fluten)
- Wenn Peer Anfrage beantworten kann leitet er dies an den anfragenden Peer zurück (nicht an den ursprünglichen, dieser bleibt anonym)
- Ursprünglicher Peer erhält IP des Peers der den gewünschten Inhalt hat und kontaktiert diesen direkt, Datenaustausch mittels HTTP
- Kein zentraler Server
- Eintritt in Overlay-Netzwerk: Nachricht an veröffentliche Liste von möglichen Peers schicken
- Skalierbarkeiten wegen Fluten problematisch

KaZaA

- Peers bilden Gruppen, einer ist Group leader und kennt alle Inhalte der Peers in seiner Gruppe (Gruppe = Mini-Napster)
- Overlay-Netzwerk zwischen Group Leadern
- Austausch zwischen Group Leadern ähnlich wie bei Gnutella
- Ebenfalls keine zentrale Kontrolle, bessere Skalierbarkeit

Auffinden von Inhalten

- Verteilte Hash-Tabelle
- Finger-Tabelle

Bittorrent

- Schwarm: Peers für gleiche Datei, z.B. Tausende
- Chunks: Teile der zu verteilenden Datei, z.B. 256 KB
- Tracker: Zentraler Server, bei dem sich Peers registrieren
- .torrent Datei mit Meta Daten über zu verteilende Datei und Tracker
- Neuer Peer A tritt Schwarm bei:
  - A registriert sich bei Tracker und erhält IP-Adressen zufälliger anderer Peers (z B 50) des Schwarms
  - A baut TCP-Verbindung zu einigen dieser Peers auf, fragt Liste der Chunks in ihrem Besitz nach und sendet Anfragen für Chunks
  - Rarest First: A fragt die seltensten Chunks der Peers zuerst nach, dadurch gleichmäßige Verteilung
  - A misst Antwortsrate der Peers antwortet an diese in entsprechendem Anteil der Upload-Rate
  - Neue Nachbarn werden zufällig dazugenommen
  - Und mehr: Hashing, Pipelining, Random First Piece, Endgame Model, Choking, Anti-Snubbing, ...
- Hybridarchitektur: Tracker sind Server IP-Adressen von Peers, Peers kommunizieren direkt

3 Transportschicht

- Aufgabe: Kommunikation zwischen Anwendungsprozessen
- TCP (Transmission Control Protocol): Verbindungorientiert, Fehler-, Fluß-, Überlastkontrolle, bietet Abstraktion eines Bytestroms

3.1 UDP

- UDP (User Datagram Protocol): Verbindungslos, keine Kontrollmechanismen, bewahrt nicht Reihenfolge, Verantwortung für Kontrollmechanismen liegt bei Anwendung
- Segment besteht aus Source Port (16 Bit), Dest Port (16 Bit), length (16 Bit) und checksum (16 Bit)

Berechnung der Prüfsumme (Folie 9)

3.2 Fehlerkontrolle

- Rauschen, Pufferüberläufe, Ausfälle von Komponenten versursachen Bitfehler und Paketverluste
- Kann druch Protokoll mit Fehlererkennung, Bestätigung und Sendewiederholung ausgeglichen werden

3.2.1 Stop-and-Wait (Folie 18ff)

- Sender fügt zur Fehlererkennung Prüfsumme oder besser Cyclic Redundancy Check (CRC) zu
- um Duplikate zu vermeiden wird eine Sequenznummer (SQN) benötigt
- Bei großem Bandbreiten-Verzögerungsprodukt: Sender ist die meiste Zeit blockiert → ineffizient

Sender

1. Sende Paket mit aktueller SQN und starte Timer
2. Wenn ein ACK ohne Bitfehler und mit aktueller SQN vor Ablauf des Timeouts zurückkommt, inkrementiere SQN und gehe zu 1.
3. Wenn Timeout abläuft, sende das Paket erneut, starte den Timer erneut und gehe zu 2.

Empfänger

- wenn Paket ohne Bitfehler und mit aktueller SQN ankommt, sende ACK mit aktueller SQN und inkrementiere SQN, sonst sende das letzte ACK erneut

Sequenznummerraum

- Die Repräsentation der Sequenznummern ist endlich: ein Feld mit n Bits ermöglicht $m = 2^n$ Sequenznummern
- Wiederverwendung durch zyklisches Durchlaufen
- Für Stop-and-Wait ist ein Bit zur Darstellung von 2 Sequenznummern ausreichend: 0 und 1 (heißt dann auch Alternating-Bit-Protokoll)

3.2.2 Go-Back-N

Um die Ineffizienz von Stop-and-Wait zu vermeiden, senden Schiebefensterprotokolle mehrere Pakete, bevor die Bestätigung zurückkommt.

- Der Sender darf mehrere Pakete (bis max. Anzahl) vor Erhalt eines ACK senden
- Ein Timer wird beim Senden des ersten Pakets gestartet
- Unbestätigte Pakete werden gepuffert

- Wenn der Timer abläuft werden alle unbestätigten Pakete erneut gesendet
- Der Empfänger schickt kumulative ACKs: ein ACK mit einer SQN bedeutet, daß alle Pakete bis zu der SQN erfolgreich empfangen wurden
- Der Empfänger akzeptiert nur Pakete in der richtigen Reihenfolge und benötigt keinen Puffer
- Kumulative ACKs gleichen Verlust und Verspätung aus

3.2.3 Selective Repeat

- Der Sender darf wieder mehrere Pakete (bis zu einer Maximalzahl) vor Erhalt eines ACKs senden
- Er startet beim Senden *jedes* Pakets einen Timer
- Er puffert die unbestätigten Pakete
- Wenn der Timer für ein Paket abläuft, wird dieses Paket erneut gesendet
- Der Empfänger schickt selektive ACKs: ein ACK mit einer SQN bedeutet nur, daß das Paket mit der SQN erfolgreich empfangen wurde
- Der Empfänger benötigt einen Puffer zum Ausgleich von Lücken beim Empfang
- Weniger Wiederholungen von Sendungen, weil nur wirklich fehlerhafte oder verlorengegangene Pakete erneut gesendet werden

3.2.4 Leistungsanalyse

Stop-and-Wait ohne Fehler

- Datenrate R
- Ausbreitungsverzögerung D
- Zu sendende Daten L
- Kanalpuffergröße a (Anzahl gesendeter Pakete während sich das erste Bit vom Sender zum Empfänger ausbreitet):
$$a = \frac{D}{L/R}$$
- Durchsatz S:
$$S = \frac{1}{1+2a}$$

Schiebefensterprotokolle ohne Fehler

- W Pakete der Länge L
- Durchsatz S:
$$S = \begin{cases} 1 & W \geq 1+2a \\ \frac{W}{1+2a} & W < 1+2a \end{cases}$$

Selective Repeat mit Fehlern

$$S = \begin{cases} 1-p & W >= 1+2a \\ \frac{W(1-p)}{1+2a} & W < 1+2a \end{cases}$$

Go-back-N mit Fehlern

$$S = \begin{cases} \frac{1-p}{1+2ap} & W >= 1+2a \\ \frac{W(1-p)}{(1-p+Wp)(1+2a)} & W < 1+2a \end{cases}$$

3.3 TCP

- Punkt-zu-Punkt: Ein Sender, ein Empfänger
- Reihenfolgebewahrender Bytestrom
- Fensterbasierte Fehlerkontrolle
- Vollduplex: 2 entgegengesetzte Datenströme
- Verbindungsorientiert: Auf- und Abbau einer Verbindung
- Flußkontrolle: Mechanismus, um Überschreitung der Kapazität des Empfängers zu verhindern
- Überlastkontrolle: Mechanismus, um Überlastung des Netzes zu verhindern
- TCP-Verbindung eindeutig gekennzeichnet durch 4-Tupel:
  - Quell-IP-Adresse
  - Ziel-IP-Adresse
  - Quell-Port
  - Ziel-Port
- Sockets sind unterschiedlich, sobald eines der 4 Tupel verschieden ist → über einen Socket können also mehrere Verbindungen lauffen
- Pseudo-Header wie bei UDP
- Prüfsummenbildung wie bei UDP

3.3.1 Fehlerkontrolle

- Mischform von Go-Back-N und Selective Repeat und weiterer Elemente
- Puffer auf Sende- und Empfängerseite
- Kumulative ACKs
- *ein* Timer
- Sequenz- und ACK-Nummern beziehen sich nicht auf Pakete: Sequenznummer = Position des ersten Bytes des Segments im Bytestrom, ACK-Nummer = Position des nächsten erwarteten Bytes im Bytestrom

Fast Retransmit

- Es dauert relativ lange bis ein Paketverlust bemerkt wird und noch länger bei mehreren Paketverlusten
- ACKs mit der gleichen ACK-Nummer heißen doppelte ACKs und sind ein schneller Hinweis auf ein fehlendes Segment
- Bei Fast Retransmit wird bei 3 doppelten ACKs (also 4 ACKs mit der gleichen ACK-Nr.) eine Sendewiederholung des Segments mit der SQN ausgelöst
- TCP ist vollduplex: es werden zwei logische Verbindungen realisiert, eine in jede Richtung
- ACKs reisen Huckepack (Piggybacking): Segmente mit Daten in die eine Richtung werden als ACKs in die andere Richtung benutzt
- Das delayed ACK soll die Anzahl von ACKs reduzieren
- Es gibt eine TCP-Erweiterung Selective Acknowledgements (SACK), bei der zusätzlich im Optionsfeld selektive ACKs gesendet werden
- Sequenznummern: TCP-Erweiterung verwendet Zeitstempel im Options-Feld für weitere Unterscheidung, um Verwechslungen von Segmenten zu vermeiden

3.3.2 Verbindungsaufbau (Folie 130 ff)

- SYN-Segment: Client sendet Segment mit SYN-Flag=1, zufälliger initialer Client-SQN (client_isn), ohne Daten
- SYNACK-Segment: Server sendet Segment mit SYN-Flag=ACK-Flag=1, zufälliger initialer Server-SQN (server_isn), ACK=client_isn+1, ohne Daten; er legt Puffer und Variablen an
- ACK-Segment: Client sendet Segment mit ACK-Flag=1; SQN=client_isn+1, ACK=server_isn+1 und ggfs. Daten; er legt Puffer und Variablen an

3.3.3 Verbindungsabbau

- Jede Seite kann Verbindungsabbau durch Segment mit FIN-Flag=1 veranlassen
- Die andere Seite bestätigt mit ACK-Flag=1
- Beide Seiten müssen ihre Hälfte der Verbindung schließen hat eine Seite geschlossen, sendet sie keine Daten mehr, nimmt aber noch welche an
- Timed Wait: die Seite, die den Verbindungsabbau veranlaßt, wartet zum Schluß noch 2 Segmentlebensdauern, um noch mögliche alte Segmente zu empfangen (und eine neue TCP-Verbindung davor zu schützen)

3.3.4 Schätzung der RTT

- Sender muss Timeout wählen
- Timeout zu klein, unnötige Sendewiederholungen
- Timeout zu groß, späte Erkennung von Fehlern
- Timoeut hängt von der Konfiguration ab und ändert sich dynamisch
- Aus Zeitstempel für Segment und ACK wird RTT berechnet
- Timeout wird aus Durchschnitt und Abweichung mittels mehereren Messungen bestimmt
- Messungen werden bei Sendewiederholung nicht verwendet

3.3.5 Flußkontrolle

- Mechanismus, mit dem der Empfänger den Sender steuern kann, damit er nicht überlastet wird, aber auch so schnell sendet wie möglich
- Üblicherweise durch Benachrichtigung über Fenstergröße
- Der freie Pufferplatz beim Empfänger wird dem Sender mitgeteilt
- Die Anwendung (die Daten in den Puffer des Senders schreibt) blockiert, wenn der Puffer voll ist
- Dadurch reguliert die Empfängeranwendung die Senderanwendung
- Initial wird das AdvertizedWindow (freier Puffer bei Empfänger) möglichst groß eingestellt
- Nach AdvertizedWindow = 0 werden periodisch Sondensegmente gesendet, damit auch wieder ACKs mit einem größerem AdvertizedWindow zurückkommen könnnen
- Silly Window Syndrom: Segmente mit wenig Daten sind ineffizient. Deswegen wartet der Empfänger bis er wieder ein AdvertizedWindow von mindestens MSS (Maximum Segment Size, üblicherweise 536 Bytes) bekanntgeben kann
- Um bei großem Bitraten-Verzögerungs-Produkt den Kanal noch immer gefüllt zu halten wird ein sog. Window-Scale-Factor im ersten Segment gesetzt

3.3.6 Überlastkontrolle

- Sender versucht über zurückkommende ACKs Informationen über mögliche Senderate zu erhalten
- CongestionWindow: Bitrate ergibt sich ungefähr aus $CongestionWindow/RTT$
- Vergrößerung des CongestionWindows: Sender vergrößert Bitrate
- Bei Segmentverlust wird CongestionWindow und damit die Bitrate wieder verkleinert

3 Mechanisment

1. Slow Start: CongestionWindow wird bis zu Segmentverlust exponentiell mit MSS erhöht
2. Dann AIMD: CongestionWindow wird halbiert und dann linear bis zum nächsten Segementverlust erhöht
3. Dann wieder AIMD
4. konservative Reaktion nach Timeout: CongestionWindow wird auf Minimum gesetzt, dann Slow Start bis zur Hälfte des aktuellen CongestionWindows und danach AIMD

3.3.7 Leistungsanalyse

- Zeit zum Kopieren eines Objektes mit TCP hängt ab von Objektgröße (O), Bitrate (R), Ausbreitungsverzögerung (D) und Verzögerungen durch Protokollmechanismen ab
- Insbesondere Slow Start kann sich spürbar auswirken
- Dynamisches Fenster:

$$Verzoegerung = 2RTT + \frac{O}{R} + P[RTT + \frac{S}{R}] - (2^P - 1)\frac{S}{R}$$

4 Netzwerkschicht

- Aufgabe: Kommunikation zwischen Hosts, die über verschiedene Netze verbunden sind
- Weiterleitung (Forwarding): Vermittlungseinheit empfängt Dateneinheiten auf einer Leitung und leitet sie auf einer anderen (der richtigen) weiter
- Wegewahl (Routing): Verfahren, mit denen Vermittlungseinheiten entscheiden, über welchen Weg Dateneinheiten gesendet werden sollen
- Keine Bereitstellung von Dienstmerkmalen wie Fehlerkontrolle, Fluß- Überlastkontrolle, Bewahrung der Reihenfolge, etc.

4.1 IP

IP-Adresse

- Kennzeichnet eine Schnittstelle eines Hosts oder Routers
- 32 Bit, 4 Bytes, Netzwerkteil und Hostteil

Klassenbasierte Adressen

- Class A: 1.0.0.0 bis 127.255.255.255
- Class B: 128.0.0.0 bis 191.255.255.255
- Class C: 192.0.0.0 bis 223.255.255.255
- 000000000000000: This host

- 0000000000 Host: Ein Host im Netzwerk
- 111111111111111: Broadcast im lokalen Netz
- Network11111111: Broadcast in entfernten Netz
- 127 xyz: Loopback

Vorteile

- Selbstidentifizierende Adressen: An den ersten Bits wird erkannt, um welche Klasse es sich handelt
- Weiterleitungstabelle benötigt nur Netzwerkteil der Adresse und kann klein gehalten werden

Nachteile

- Feste Zuordnung von Netzwerken, wenn ein Rechner in ein anderes Netz umzieht muss sein IP-Adresse angepasst werden
- C-Netze erlauben nur wenige Hosts (8 Bit), B-Netze dagegen sehr viele (16 Bit) → Verschwendung, Organisationen bemühen sich um B-Netze und nutzen Adressen nur teilweise

Subnetze

- Hostanteil wird weiter unterteilt in Subnetz (variable Länge) und Host
- Notation für das Netzwerk + Subnetz: IP-Adresse/Länge der Maske (Anzahl der 1en), z.B.: 150.100.12.176/25
- Router brauchen noch immer nur den klassenbasierten Netzwerkteil
- Binäre Adresse mit Subnetzmaske verunden (*logisches UND!*, also beides müssen 1er sein), fertig

CIDR (Classless Inter-Domain Routing)

- Weiterleitung basiert nicht mehr auf klassenbasierten Netzwerkadressen sondern auf einer beliebigen Anzahl von Bits in der Adresse, die durch die Maske gekennzeichnet werden. Z.B. Subnetting
- Die Adresse und die Maske können als Einträge in die Weiterleitungstabelle geschrieben werden → starre Zuordnung aufgelöst, jedoch größere Tabellen

4.1.1 Fragmentierung (Folie 25 ff)

- Wenn Verbindungen unterwegs ein kleinere MTU (Maximum Transmiss Unit) erfordern, wird das Datagram in Fragmente zerlegt und als kleine Datagramme weitergeleitet dies kann sich u.U. mehrmals wiederholen
- IP-Header-Felder: identifier: Kennzeichnung zusammengehöriger Fragmente, flag: Fragment folgt, offset: Position der Daten des Fragments bei offset*8
- Reassemblierung: Erst am Ziel wird wieder zusammengesetzt

4.1.2 ICMP (Internet Control Message Protocol) (Folie 28)

Kontrollnachrichten von Routern an andere Router und Hosts z.B. Benachrichtigung über Fehler (unerreichbare Adresse, maximale Zahl von Hops erreicht, ...). Daten werden in IP-Datagrammen befördert.

4.1.3 NAT (Network Access Translation)

- Umgehung der Adressknappheit
- Netzwerk verwendet intern global ungültige Adressen, z.B. 10.0.0/24
- Netzwerk besitzt eine global gültige IP-Adresse
- Verbindungen zu internen Hosts werden auf Paare abgebildet, die aus dieser Adresse und einem Port bestehen
- nächster Router muqß die Abbildung ausführen, er besitzt Tabelle hierfür und überschreibt Adressen und Ports in den IP-Datagrammen (in beide Richtungen)
- Größe durch Anzahl von Portnummern begrenzt
- Nachteil: Verletzung des Schichtenprinzips (Router beschäftigt sich mit Hosts) und Eingriff in die Ende-zu-Ende-Verbindung
- Vorteil: interne Änderungen ohne externe Auswirkung, bessere Abschirmung

4.1.4 IPv6 (Folie 34 ff)

- IETF-Standardisierungsbemühung initiiert wegen Adressraumproblemen von IPv4, dabei gleich Lösung weiterer Probleme
  - Header mit fester Länge (für schnelles Weiterleiten)
  - keine Fragmentierung (wird einfach verworfen falls ¿ MTU)
  - keine Prüfsumme (Fehlererkennung in höheren Schichten)
  - zusätzliche Optionen außerhalb als nächster Header
  - Autokonfiguration (inzwischen DHCP)
  - Dienstgütemerkmale (inzwischen IntServ, DiffServ)
- Probleme bei der Durchsetzung, inzwischen „kann“ IPv4 vieles auch
- Adressen sind 128 Bits (source address und dest adress in header)

Übergang von IPv4 zu IPv6: Dual Stack IPv6-Knoten haben auch IPv4-Implementierung und können Datagramme als IPv4 versenden, zusätzliche IPv6- Information geht dabei verloren.

Übergang von IPv4 zu IPv6: Tunneling Tunneling: das IPv6-Datagramm wird in ein IPv4-Datagramm eingepackt.

4.2 Aufbau eines Routers

- Aufgaben: Weiterleitung, Ausführung von Routingprotokollen
- Besteht aus Switching Fabric und Routing Processor
- Pufferung (am Eingang), falls Pakete schneller von der Leitung kommen als sie weitergegeben werden können → Paketverlust falls Puffer überläuft
- Effiziente Datenstrukturen für Schnelle Suche des Ziels
- Pufferung (am Ausgang), falls Switching Fabric schneller liefert als Pakete auf die Leitung gegeben werden können → Paketverlust...
- Active Queue Management (am Ausgang): Entscheidung, welches Paket verworfen wird
- Scheduling (am Ausgang): wenn mehrere Pakete gepuffert sind, kann entschieden werden, welches als nächstes gesendet wird

Möglichkeiten für die Switching Fabric

- Speicher: CPU kopiert Paket von Eingangsport in Hauptspeicher, führt Weiterleitungsentscheidung durch und kopiert Paket zum Ausgangsport. 2 mal internen Bus benutzen, Begrenzung der Leistungsfähigkeit. Frühe Router waren einfache Rechner, heute auch noch möglich.
- Bus: Ein Bus verbindet alle Ports, kann nur jeweils für einen Transfer benutzt werden, Wettbewerb. Üblich für kleine bis mittlere Router
- Verbindungsnetzwerk: Bekannt aus der Verbindung von Prozessoren in Parallelrechnern z.B. Crossbar: jeder Port kann direkt mit jedem anderen verbunden werden (quadratischer Schaltungsaufwand) auch mehrstufige Anordnungen

Pufferungs- und Verlusteffekte

- Wenn Switching Fabric schneller als Anzahl Ports x Leitungsgeschwindigkeit
- Normalerweise keine Pufferung bei Eingangsports notwendig
- Wenn mehrere gleichzeitig zu einem Ausgangsport schicken, ist dort Pufferung notwendig
- Head-of-the-Line (HOL) Blocking: Mehrere Eingangports wollen auf gleichen Ausgangsport. Manche müssen warten, die dahinter werden blockiert, obwohl ihr Ausgangsport frei wäre.

4.3 Routing

- Verfahren, mit denen Router entscheiden, über welchen Weg Pakete gesendet werden sollen
- Intradomain: Innerhalb einer Routing-Domäne (= unter einer andimistrativen Instanz), hier können Verfahren verwendet werden, die für sehr große Netze nicht skalieren: Link-State, Distanzvektor
- Interdomain: Zwischen Routing-Domänen, ausgetauschte Routing-Informationen enthalten ganze Pfade, Auswahl durch Regeln, Beispiel: BGP

- Unicast-Routing (Punkt-zu-Punkt): proaktiv, Information über Netztopologie wird ausgetaucht, aktuell gehalten, mit Graph-basierten Verfahren werden Pfade zu allen Zielen bestimmt, bei Sendewunsch werden diese genutzt
- Multicast-Routing (Punkt-zu-Mehrpunkt): proaktiv, Router sollen effizient genutzt werden, Erweiterung von Unicast-Routing
- Ad-Hoc-Routing: Dynamische Netztopologie (Pfade veralten schnell), Erweiterung von proaktivem Verfahren. Auch reaktive Verfahren (erst bei Sendewunsch wird Pfad bestimmt)
- Datenzentrische Verfahren: Adresslos, Daten werden aufgrund ihres Inhalts weitergeleitet, z.B. in Sensornetzen

4.3.1 Link-State-Routing

- Alle Knoten besitzen vollständige Kenntnis der Netztopologie
- Dies wird durch Fluten erreicht
- Jeder Knoten berechnet die kürzesten Pfade zu allen anderen Knoten (Dijkstra-Verfahren)
- Bei Änderungen in der Netztopologie (kann z.B. die Verbindungsschicht erkennen), erfolgt erneutes Fluten und Neuberechnung der kürzesten Wege

Fluten

- Link-State-Advertisements (LSAs) mit
  - Kennung des Knotens, der LSA erzeugt
  - Kosten zu Nachbarn und dessen Kennung
  - Sequenznummer und Lebensdauer
- Jeder Knoten erzeugt LSAs mit den ihm bekannten Informationen über die Verbindungen zu den Nachbarn und sendet sie an alle Nachbarn
- Neue von Nachbarn erhaltene LSAs werden an alle Nachbarn weitergeleitet, aber nicht an den Nachbarn, von dem das LSA kam
- Zur Erzielung von Zuverlässigkeit auch Bestätigungen und Sendewiederholungen zwischen Nachbarn sowie Sequenznummern und Lebensdauer
- Vergleiche *bestätigte Liste und vorläufige Liste*

OSPF (Open Shortest Path First) (Folie 71 ff)

4.3.2 Distanzvektor-Routing (Folie 73 ff)

- Die Suche nach kürzesten Wegen wird verteilt durch alle beteiligten Knoten durchgeführt
- Jeder Knoten teilt seinen Nachbarn mit, mit welchen minimalen Kosten er andere Knoten erreichen kann
- Anfangs können nur die Kosten zu den Nachbarn bekanntgegeben werden, mit jedem Austausch werden längere Pfade bekannt

- Der Algorithmus endet, wenn sich keine Veränderung mehr ergibt (Konvergenz wurde erreicht)
- Der Algorithmus funktioniert auch bei Topologieänderungen und wenn die Informationen in asynchroner Weise ausgetauscht werden (also nicht jeweils gleichzeitig)
- Bei Verkleinerung der Verbindungskosten konvergiert das Verfahren schnell: good news travel fast, siehe nächstes Beispiel
- Bei Vergrößerung der Verbindungskosten können jedoch durch Zyklen in den Pfaden Probleme entstehen: bad news travel slowly, siehe übernächstes Beispiel

Count-to-infinity-Problem

- Veraltete Information in den verteilten Routing-Tabellen enthält zyklischen Pfad
- Die langsame Iteration endet erst, wenn die Kosten des alternativen Pfads erreicht sind
- Abhilfe:
  - Größten Kostenwert beschränken (z.B. 16)
  - Poisoned Reverse: wenn der kürzeste Weg von u nach v über den nächsten Hop w führt, sendet u an w die Kosten von unendlich für die Entfernung von u nach v
- Mit Poisoned Reverse können Zyklen der Länge 2 vermieden werden, nicht jedoch längere Zyklen

RIP (Routing Information Protocol)

- Distanzvektor-Routing
- Router teilen in Advertisements über UDP den Nachbarn mit, welche Netzwerke sie mit welchen Kosten erreichen können
- Advertisements werden periodisch (alle 30 s) und bei Änderungen gesendet
- Kostenmetrik: Anzahl von Hops, maximal 15 (16 für Poisoned Reverse)
- Route-Dämon als Anwendungsprozeß
- Verwendung von Datenstrukturen in der Netzwerkschicht

Vergleich Link-State-Routing und Distanzvektor-Routing (Folie 92) Allgemein gilt: Dynamische Metriken (die von der aktuellen Netzlast abhängen), führen zu instabilem Verhalten und haben sich nicht bewährt

4.3.3 Interdomain-Routing (Folie 93 ff)

- Innerhalb einer Routing-Domäne: Intradomain Routing, Interior Gateway Protocol (IGP)
- Zwischen Routing-Domänen: Interdomain Routing, Exterior Gateway Protocol (EGP) (Border Gateway Protocol (BGP))
- Routingdomänen: Autonome Systeme (AS):

  - Stub AS: Hat nur eine Verbindung zu anderen AS
  - Multihomed AS: Hat mehrere Verbindungen zu anderen AS, befördert aber keinen Durchgangsverkehr
  - Transit AS: Hat mehrere Verbindungen zu anderen AS, befördert auch Durchgangsverkehr
- Ein Router eines AS ist Gateway und führt EGP aus

Pfadbasiertes Routing

- Austasch von ganzen Pfaden
- Keine Betrachtung der Kosten (weil zu groß, unterschiedliche Metriken, etc.)
- Router gibt nur die Pfade bekannt, die andere Router nutzen sollen
- Zyklen können erkannt werden (ein Router zweimal im Pfad)
- Pfade können auch annuliert werden
- Gateways tauschen Advertisements in BGP-Sessions über TCP aus
- Advertisements enthalten Pfade, die Gateways zu Netzwerken kennen
- Ein Pfad besteht aus:
  - Liste von Netzwerken (Adresse+Präfix, CIDR) in einem erreichbaren AS: Z.B.: Alle Subnetze in AS1
  - Sequenz von AS-Nummern zu diesem AS (Hops): AS2, AS1
  - IP-Addresse des sendenden Gateways: Als erster Hop einfuegen
- Beim Weiterleiten eines Pfades fügt ein Router sein AS am Anfang des Pfads an und setzt sich als den nächsten Hop ein
- Dadurch erlaubt er, daß Verkehr an die Liste von Netzwerken über ihn geleitet wird

4.4 MPLS (Multiprotocol label switching)

- Beispiel für virtuelle Leitungsvermittlung, Wurzeln in der ATM-Technologie (virtuelle Leitungen)
- IP Pakete erhalten Label
- Weiterleitung durch die Label Switched Routers (LSRs) entlang des Label- Switched Path (LSP)
- Jeder LSR erkennt Eingangsport und Label, sieht in Tabelle nach, welches das neue Label sein soll, und auf welchem Ausgangsport das Paket weitergeleitet werden soll
- Label Swapping: das alte wird durch das neue Label ersetzt
- Weiterleitung an den Ausgangsport
- LSR muß vorab aufgebaut werden

- Vorteile:
  - Schnelles Weiterleiten (kein Longest Prefix Match)
  - Planung von LSPs für die gezielte Verkehrsleitung
  - Virtual Private Networks (VPNs): Tunnels zwischen Knoten
  - Aggregation von LSPs durch Label Stacking

5 Verbindungsschicht

- Aufgabe: Transfer der Rahmen zwischen Knoten (Hosts, Router) über Verbindungen
- Datagramme der Netwerkschicht in Rahmen ein-/auspacken
- Adressierung: Rahmen enthält physikalische Adresse der Knoten
- Datensicherung
- Medienzugriff (Medium Access, MAC)
- evtl. Flußkontrolle
- Funktionalität meistens in Netzwerkkarte
- Schnittstelle vom Systembus des Knotens und Netzwerk

5.1 Adressierung

Physikalische Adresse (MAC-Adresse)

- 46 Bits, 6 Bytes, 12 Hexadezimalziffern
- In ROM des Adapters eingebrannt (Verwaltung durch IEEE)
- Global eindeutig, keine Strukturierung, nicht die IP-Adresse

Address Resolution Protocol (ARP)

- Jeder Knoten besitzt ARP-Tabelle mit Zuordnung: IP-Adresse, MAC-Adresse, TTL
- Knoten A möchte an B Rahmen schicken, kennt IP-Adresse, aber nicht physikalische Adresse von B
- A sendet ARP-Anfrage als Broadcast (Adresse FF-FF-FF-FF-FF) mit seiner physikalischen Adresse und der IP-Adresse von B
- B erkennt sich als Ziel an IP-Adresse in der ARP-Anfrage und sendet in ARP-Antwort seine physikalische Adresse an die physikalische Adresse von A
- A speichert die Zuordnung der Adressen von B in seiner ARP-Tabelle

Beispiel

- A erzeugt Datagramm mit IP-Quelladresse A und IP-Zieladresse B A findet R in seiner Routingtabelle
- A benutzt ARP um die physikalische Adresse des Adapters von R an LAN1 zu finden
- A erzeugt einen Rahmen mit sich als physikalischer Quelladresse, physikalische Zieladresse ist der Adapter von R an LAN1 (die IP- Zieladresse im eingepackten Datagramm bleibt B!)
- Der Adapter von A sendet den Rahmen auf LAN1
- R's Adapter in LAN1 empfängt den Rahmen und packt das Datagramm aus, liest die IP-Zieladresse B, findet in der Routingtabelle heraus, daß B in LAN2 ist
- R benutzt ARP um die physikalische Adresse von B zu finden
- R erzeugt einen Rahmen mit seinem Adapter in LAN2 als physikalischer Quelladresse und B als physikalischer Zieladresse (die IP-Quelladresse bleibt A!)
- R's Adapter in LAN2 versendet den Rahmen
- B's Adapter empfängt den Rahmen und liefert das Datagramm aus

5.2 Datensicherung (Folie 13 ff)

- Fehlererkennung: Z.B. Prüfdaten um Sendewiederholung zu veranlassen
- Fehlerkorrektur: Kodierung, so dass der Empfänger Fehler erkennen und korrigieren kann

5.2.1 Zyklische Redundanzprüfung (Cyclic Redundancy Check, CRC) TODO

5.3 Medienzugriff

Möglichkeiten für den Mehrfachzugriff auf eine Leitung:

- Zufallszugriffverfahren:
  - Stationen greifen zufällig auf Medium zu, eventuell gleichzeitige Übertragungen (Kollisionen) müssen beachtet werden
  - Urform: ALOHA, abgeleitet: MAC bei Ethernet und WLAN
- Zyklische Zuteilung:
  - Zentralisiert: Polling durch zentralen Knoten
  - Verteilt: Sendeerlaubnis durch rotierendes Bitmuster (Token), z.B. Token Ring, FDDI, USB, Profibus

5.3.1 Feste Kanelaufteilung

- Durch geeignetes Multiplexverfahren, wird das Medium in feste Kanäle für Knotenpaare aufgeteilt
- Bekannt: Frequenz-, Zeit-, Codemultiplex
- Für Sprachkommunikation verbreitet
- Nachteil für Datenkommunikation: Daten werden typischerweise sporadisch versendet, ineffiziente Nutzung des Mediums

Codemultiplex (Code Division Multiplex Access, CDMA)

- Spreiztechnik: Der Sender multipliziert jedes Bit mit einem Chipping-Code
- Er erzeugt dadurch ein Signal mit höherer Frequenz und sendet dieses auf das Medium (er benutzt dazu das volle Spektrum und die gesamte Zeit)
- Die gespreizten Signale überlagern sich auf dem Medium
- Der Empfänger kann hieraus mit dem Chipping-Code das einzelne gesendete Signal extrahieren
- Andere Variante: Frequenzsprungverfahren, der Sender springt während des Sendens eines Bits zwischen verschiedenen Sequenzen, dies erlaubt die Überlagerung vieler Signale auf dem Kanal, durch Kenntnis des Sprungmusters kann das Signal empfangen werden
- Ursprung: Militärtechnik, bekannt aus Mobilfunk

Effizienz fester Kanalaufteilung

- Datenkommunikation ist Bursty (schubartig), feste Kanalaufteilung ist ineffizient
- Mittlere Wartezeit wächst mit Anzahl der Kanäle

5.3.2 Zufallszugriff

- Wenn Knoten Rahmen zum Senden hat, sendet er mit der vollen Bitrate des Mediums
- Wenn Knoten gleichzeitig senden, überlagern sich die Signale auf dem Medium und zerstören sich gegenseitig (normalerweise): es kommt zur Kollision, die durch Sendewiederholung behoben wird
- Grundidee: bei schwacher Last ist dies selten
- Unterschiedliche Verfahren zur Vermeidung und Erkennung von Kollisionen
  - ALOHA, slotted ALOHA
  - Carrier Sense Multiple Access (CSMA)
  - Mit Collision Detection: CSMA/CD (in Ethernet)
  - Mit Collision Avoidance: CSMA/CA (in WLANs)

ALOHA

- Gemeinstamer Funkkanal für alle Knoten
- Wenn die MAC-Schicht eines Knotens von der Netzwerkschicht ein Datagramm erhält, wird der Rahmen sofort gesendet
- Wenn der Empfänger ihn fehlerlos erhält, sendet er eine positive Bestätigung (ACK) zurück
- Wenn nach einem Timeout kein ACK zurückkommt, wartet der Sender eine zufällige Wartezeit (Backoff) und wiederholt dann das Senden
- Das Protokoll ist einfach, verteilt, es gibt keine Absprachen zwischen den Knoten
- Hier ist Produkt aus Bitrate und Verzögerung klein ($a < 1$)
- Ähnlichkeit zu Stop-and-Wait
- Backoff:
  - Nach einer maximalen Zahl M von Kollisionen (z.B. M=10), bricht die MAC-Schicht ab und meldet einen Fehler an die Netzwerkschicht
  - Idee:
    - Backoffzeit an akutelle Last anpassen
    - Niedrige Last: Vermutlich sind nur wenige Knoten an der Kollision beteiligt, Auswahl von K (Wartezeiten) aus wenigen Möglichkeiten reicht
    - Hohe Last: Mehr kollidierende Knoten, Auswahl von K aus mehr Möglichkeiten, größere mittlere Backoffzeit
- Leistungsanalyse: Verschwendung von Kollisionen und Backoffzeiten, selbst bei optimaler Einstellung der Last kann maximal 18% Durchsatz erreicht werden

Slotted ALOHA

- Alle Knoten synchronisieren ihre Slots (z.B. durch zentrales Zeitsignal)
- Sendebeginn nur zu Beginn eines Slots, Kollisionsintervall verkürzt sich auf einen Slot
- Selbst bei optimaler Einstellung der Last kann maximal 37% Durchsatz erreicht werden. Aber besser als ALOHA

Carrier Sense Multiple Access (CSMA)

- Knoten prüfen vor dem Senden, ob Medium belegt (listen before talking)
- Reduziert Kollisionen
- Voraussetzung: Ausbreitungsverzögerung ¡ Rahmensendezeit (sonst sinnlos)
- Kollisionen immer noch möglich: Wenn anderer Knoten startet, bevor sich das Signal auf dem Medium zu ihm ausgebreitet hat
- Wenn die MAC-Schicht eines Knotens von der Netzwerkschicht ein Datagramm erhält, überprüft sie das Medium (listen before talking)

- Wenn es frei ist, wird der Rahmen gesendet, sonst wird gewartet
- Wenn der Empfänger ihn fehlerlos erhält, sendet er eine positive Bestätigung (ACK) zurück
- Wenn nach einem Timeout kein ACK zurückkommt, wartet der Sender eine zufällige Wartezeit (Backoff) und wiederholt dann das Senden

CSMA-Varianten (Folie 43)

- 1-persistent:
  - Wenn das Medium belegt ist, wartet der Knoten bis es frei ist und sendet dann sofort
  - Geringe Wartezeit aber mögliche neue Kollision, wenn mehrere Knoten auf freies Medium warten
- nicht-persistent:
  - Wenn das Medium belegt ist, geht der Knoten in Backoff
  - Weniger Kollisionen aber längere Wartezeit
- p-persisten:
  - Wenn das Medium belegt war und wieder frei ist, sendet der Knoten jeweils mit Wahrscheinlichkeit p oder wartet noch einen Slot mit Wahrscheinlichkeit 1-p
  - Kompromiß

CSMA/CD

- Knoten besitzen HW, um während des Sendens Kollision zu erkennen (listen while talking)
- Nach Kollisionserkennung wird Senden abgebrochen (weniger Verschwendung), ein Jamming-Signal wird gesendet, damit alle Knoten Kollision sicher erkennen
- Keine ACKs
- Kombinierbar mit allen CSMA- Varianten
- Kleines a (Kanalpuffergröße): CSMA/CD am besten
- ALOHA, ALOHA slotted ALOHA unabhängig von a, besser für großes a a

Reservation ALOHA

- Aufteilung in Minislots and Slots
- Wettbewerb der Knoten um Minislots mit Slotted ALOHA
- In einem Minislot wird ein Slot reserviert, der eine Weile für TDMA genutzt wird
- Mischform von Zufallszugriff und fester Kanalaufteilung
- Leistung 80 - 90 %
- Verwendung z.B. in der Mobilkommunikation

5.3.3 Zyklische Zuteilung

Polling

- Sendeerlaubnis wird dem Knoten sukzessive zugewiesen
- durch zentralen Knoten, zufällig ausgewählten Knoten oder durch ein verteiltes Protokoll
- Reihenfolge zyklisch oder anders (z.B. prioritätsgesteuert)
- Zykluszeit: Zeit bis Sendeerlaubnis zurückkommt = für jeden Knoten: Sendezeit für Sendeerlaubnis + Ausbreitungszeiten + Verarbeitungszeiten + Sendezeit für Daten
- Nachteile: Overhead, zentraler Knoten ist *Single-Point-of-Failure*

Token Ring (Folie 57 ff)

- Die Knoten sind ringförmig vernetzt
- Knotenadapter hat Eingang und Ausgang, 2 Modi:
  - Listen Mode: Bits vom Eingang werden mit Pufferung (typisch 1 Bit) weitergereicht, Knoten erhält Kopie
  - Transmit Mode: Bits vom Eingang werden an Knoten geleitet, Bits zum Ausgang kommen vom Knoten
- Ein Bitmuster (Token) zirkuliert auf Ring, 2 Zustände (frei, belegt), z.B.: frei = 0111111*0*, belegt = 0111111*1*
- Wenn Knoten freies Token empfängt und Sendewunsch vorliegt, verändert er das Token durch Umsetzen des Bits in belegt und sendet Token
- Als nächstes sendet der Knoten Daten
- Der Empfänger erhält die Daten
- Nach Ringumlauf entfernt der Sender das belegte Token und die Daten wieder vom Ring und sendet das freie Token weiter

Multi-Token Ring

- Single-Token Ring ineffizient für große a
- Abhilfe: Sender sendet nach Rahmen direkt freies Token, auf dem Ring können sich mehrere Token befinden
- z.B. bei FDDI
- Timed Token Algorithmus zur Kontrolle der Zykluszeit (geht über VL hinaus)
- Leistungsanalyse: maximaler Durchsatz
  - Jeder Knoten sendet Rahmen, wenn er freies Token erhält
  - Overhead ist nur Zykluszeit

5.4 Ethernet

- Rahmenformat: Preamble, Dest. Address (MAC), Source Address (MAC), Type, Data, CRC
- 1-persistentes CSMA/CD, Jam-Signal: 48 Bits
- Verbindungslos: Kein Handshaking erforderlich
- Unzuverlässig: Kein Versenden von Bestätigungen

Ursprüngliche Bus-Topologie (10Base2

- Koaxialkabel ist Bus, Knoten über Transeiver angeschlossen
- Datenrate 10 Mbps

Repeater

- Zur Auffrischung von Signalen
- Operiert auf physikalischer Schicht

Bridge

- Verbindung von Ethernet-Segmenten
- Bei jedem Empfang eines Rahmens an einem Eingangsport wird entschieden, an welchen Ausgangsport der Rahmen weitergeleitet wird und mittels CSMA/CD auf das Medium dieses Segments gegeben
- Operiert auf Verbindungsschicht
- Aufteilung in verschiedene Kollisionsdomänen

Stern-Topologie mit Hub (10BaseT)

- Hub: Repeater mit vielen Ports, keine Pufferung, aber Managementfunktion
- Alle Knoten werden an zentralen Hub angeschlossen, das Signal auf jedem eingehenden Port wird auf jeden ausgehenden Port weitergegeben
- Eine Kollisionsdomäne, CSMA/CD
- Twisted-Pair, RJ-45 (wie bei Telefon)
- R = 10 Mbps
- Entfernung Hub-Knoten max.100 m
- Kaskadierung möglich

Stern-Topologie mit Switch (10BaseT)

- Switch: Bridge mit vielen Ports, Pufferung an jedem Port
- Knoten führen noch CSMA/CD durch, Kollisionen treten aber nicht mehr auf
- Kaskadierung möglich, Kombination mit Hubs möglich

Fast Ethernet

- Sterntopologie, Hubs, Switches
- R = 100 Mbps
- 2 Modi: mit CSMA/CD für Hubs, ohne CSMA/CD für Switches
- Rahmenformat gleich
- Entfernung Hub-Knoten: Twisted Pair: max. 100 m (100BaseT) Glasfaser: max. 2000 m (100BaseFX)
- Kaskadierung, Kombination Switches/Hubs möglich
- Kombination 10BaseT/100BaseT möglich: Switches mit Dual-Speed- Anschlüssen, die sowohl 10BaseT als auch 100BaseT beherrschen

Gigabit Ethernet

- R = 1 Gbps, gleiches Rahmenformat
- Hubs (Buffered Distributers) mit Kollisionen, minimale Rahmengröße 512 Bytes (um Bedingung für Sendezeit und Ausbreitungszeit zu genügen)
- Switches ohne CSMA/CD
- 1000BaseT: Twisted Pair, 100 m
- 1000BaseSX: Multimode-Glasfaser (550 m)
- 1000BaseLX: Singlemode-Glasfaser (5 Km)

10 Gigabit Ethernet

- R = 10 Gbps, gleiches Rahmenformat
- CSMA/CD aufgegeben
- Nur Switches
- Entfernungen Multimode bis 300 m, Singlemode bis 40 Km

Selbstlernend

- Wenn Bridge/Switch Rahmen erhält, muß sie/er Entscheidung treffen, wohin Rahmen weitergeleitet werden soll
- Wenn die physikalische Zieladresse an dem Port ist, von dem der Rahmen kommt, wird er verworfen
- Wenn der Port der physikalischen Adresse unbekannt ist, wird der Rahmen an alle Ports geflutet
- Für einen eingehenden Rahmen wird die Zuordnung von physikalischer Adresse und Portnummer in Tabelle gespeichert
- Soft State, TTL z.B. 60 Minuten
- Mit Bridges/Switches können zyklische Strukturen aufgebaut werden: Dann funktioniert obiger Algorithmus nicht
- Alle Bridges/Switches in einem LAN führen verteilten Algorithmus durch, bei dem im Graph Kanten deaktiviert werden, so daß er einen aufspannenden Baum bildet

5.5 Campusnetzwerke

- Netzwerk für größere Institution (z.B. Uni, Unternehmen), Grundstück, mehrere Gebäude
- Besteht aus verschiedenen LANs mit Hosts, Hubs, Switches, Routern
- Aufteilung in virtuelle LANs (VLANs) möglich

Virtuelle LANs (VLANs)

- Broadcastdomäne in einem geswitchten Netz
- Ziele: Anpassung der logischen Netztopologie an Unternehmensorganisation (z.B. Arbeitsgruppen, Benutzermobilität), Lastoptimierung
- Port-basiert:
  - Endgeräte an bestimmten Ports bilden VLAN
  - Festlegung durch Konfiguration des Switch
  - Rahmen werden von Switch nur innerhalb VLAN weitergeleitet,
  - Inter-VLAN-Verkehr geht über Router
- Tagging:
  - IEEE 802.1q beschreibt VLAN-Tags in Ethernet-Rahmen
  - VLAN-tagged Rahmen werden zwischen Switches ausgetauscht
  - Ermöglicht VLANs über mehrere Switches

Hierarchische Architektur von Campusnetzwerken (Folie 83 ff)

- Hier: Hierarchical Network Design Model (von Cisco)
- Strukturierung in funktionale Schichten (Layers) und Typen von Modulen (Blocks)
- Einsatz von Multilayer-Switching
- Funktionale Schichten:
  - Core Layer: zentrales Backbone zur Verbindung der Gebäudenetze, einfache Erweiterbarkeit ohne grundlegende Strukturveränderungen
  - Distribution Layer: räumliche Verteilung der Netzwerkleistung im Gebäude
  - Access Layer: Netzzugang für Endgeräte
- Modultypen:
  - *Building Block* (Gebäudemodul): Für jedes Gebäude, Access Layer mit Layer-2-Switches als Etagenverteiler, Distribution Layer mit Layer-3-Switch als Gebäudeverteiler
  - *Core Block* (Backbone): normalerweise ein Core Block, leistungsfähige Layer-2- und Layer-3-Switches
  - *Server Block* (Server-Modul): Anbindung zentraler Server, kleinere benötigen nur Distribution Layer, größere auch Access Layer
  - *Building Block Addition* (Gebäudemodulergänzung): Verbindung zu anderen Standorten über Standleitungen, zum Internet, zum Telefonnetz (Public Switched Telephone Network, PSTN)

5.6 Drahtlose LANs

- Schwund der Signalstärke ca. quadratisch mit Entfernung, abhängig von Umgebung (Long-Range Fading)
- Interferenzen durch andere Sender (drahtlose Netze, schnurlose und mobile Telefone, Mikrowellenöfen, Motoren, ...)
- Mehrwegausbreitung: Funkwellen werden reflektiert, phasenverschobene Wellen überlagern sich und schwächen sich kurzfristig ab bzw. löschen sich aus (Short-Range Fading)
- Höhere Fehlerrate, insbesondere als Bursts
- Ursprünglich 1-2 Mbps, Funk (Direct Sequence + Frequency Hopping Spreiztechnik), auch Infrarot
- Weiterentwicklung: Ergänzungen in diversen Arbeitsgruppen, 11b (11 Mbps), 11g (54 Mbps), 11i (Sicherheit), 11e (Quality-of-Service), 11p (Car-to-X), ...
- 2 Betriebsmodi: Infrastrukturnetz (Access Points, Distribution Service), Ad-Hoc-Netz
- Adressierung: 4 physikalische Adressen (Quelle, Ziel, etc.) für Access Points
- Medienzugriff: CSMA/CA (Collision Avoidance): Kollisionserkennung würde 2. Antenne benötigen, die während des Sendens empfängt, schwierig, teuer
- Energiesparen: Schlafphasen, synchronisiertes Aufwachen

Medienzugriff gemäß Basic Access (Folie 97 ff)

- Wenn die MAC-Schicht einer Station von der Netzwerkschicht ein Datagramm erhält, überprüft sie das Medium (listen before talking); wenn es eine Zeitdauer DIFS (Distribution Interframe Space) frei ist, wird der Rahmen gesendet, sonst geht sie in Backoff
- Wenn der Empfänger ihn fehlerlos erhält, wartet er eine Zeitdauer SIFS (Short Interframe Space), dann sendet er eine positive Bestätigung (ACK) zurück
- Wenn nach einem Timeout kein ACK zurückkommt, geht der Sender in Backoff
- Truncated Binary Exponential Backoff: Abhängig von Anzahl der Kollisionen würfelt der Sender eine zufällige Wartezeit, reduziert sie solange das Medium frei ist und geht nach Ablauf zum 1. Schritt zurück
- Variante von nicht-persistentem CSMA/CA (DIFS, SIFS ist neuer Aspekt)

RTS/CTS-Austausch

- Vorheriger Austausch von kurzen Reservierungsnachrichten
- Sender sendet Request-To-Send (RTS) mit Länge des Rahmens, Empfänger (Station bei Ad-Hoc-Netz, AP bei Infrastrukturnetz) antwortet mit Clear-To-Send (CTS), in der Länge auch steht
- Medienzugriff für RTS mit Basic Access
- Nachteil: größerer Overhead
- Vorteile: Wenn keine Kollision auftritt, ist Medium reserviert, wenn Kollision auftritt, dauert diese nicht lange (RTS kurz)
- Hidden-Terminal-Problem teilweise gelöst: die Stationen, die Sender nicht hören erfahren vom Empfänger Reservierung

6 Physikalische Schicht

7 Klausurfragen