Client-Server-Paradigma Server stellt Dienst bereit, der von Client angefordert wird. Rollen können dabei wechseln **verteilte Anwendung** aus meherern kleinen Anwendungen eine große z.B: Datenbank, Applikations-Server´ **P2P** dezentral→keine Steuereinheit; Hybridarchitektur→zentrale Architektur für Initialisierung

Anforderung an untere Schicht Datenverlust schlecht, bei z.B. http; Ok bei Multimedia **Bitrate** Multimedia braucht mindestbitrate, rest meist nicht **Verzögerungszeit** schlecht bei Echtzeit-Multimedia und interaktive Spiele, Sicherheitskritische Dinge

HTTP Ablauf URI (Host, Pfad) in Webbrowser, dieser anfrage an Webserver **Anfragenachricht** METHODE URI Version \r\n; 2-n. Zeile: Headerzeilen \r\n; \r \n; Rumpf; **Methoden** GET, HEAD, POST, PUT, DELETE **Antwortnachricht** Version StatusCode Phrase; Header...; Leerzeile; Rumpf **Headerzeilen** Server Set-cookie: hallo=abc; ciau=cba; Client cookie: hallo=abc;ciau=cba; **Statuscode** 200 OK; 400 Bad Request; 404 Not Found **Nicht persistent** Jedes Objekt einzelne Verbindung (entweder parallele Verbindungen, oder sequenzielle) **persistent** alle Objekte über eine Verbindung ohne piplining immer Anfrage für nächstes, mit piplining alle auf einmal angefragt **Web Caching** verhindert lange Wartezeiten durch Zwischenspeicherung; Kann über Http-Anfrage mit If-modified-since: [date] abgefragt werden. Wenn passt Antwort mit 304 Not Modified, sonst 200 OK mit Daten **Http/1.1 vs Http1.2** Bei 1 Piplining Reihnfolge der übertragenen Daten wird eingehalten; Bei 2 Datei in Frames unterteilt kleine Datei kann kurz zwischen großer übertragen werden

FTP zur übertragung von Dateien zw. Hosts TCP an Port 21 (Steuerleitung für Kommandos) Port 20 Dateiübertragung **Mode** Active: Server baut verbindung für Datenübertragung auf, Client sendet PORT N+1 ; Passiv: Client baut Verbindung zu Server auf Client sendet PASV;

Smtp TCP Port 25; 7-bit ASCII-Text auch Anhang multimedia mail extension (MIME) **Ablauf** S 220 [S-host]; C HELO [C-host]; S 250 Hello [C-host]; C MAIL FROM: <[Absender]>; S 250 [Absender]... Sender ok; C RCPT TO: <[Empänger]>; 250 [Empfänger] ... Recipient ok; C DATA; S 354 Enter mail, end with "." on a line by itself; C [Nachricht mit letzter Zeile nur ein Punkt]; S 250 Message accepted for delivery; C Quit; S 221 [S-host] closing connection;

Email Sicherheit Email Verschlüsselung Text mit K_s symmetrischen Schlüssel verschlüsseln; K_s mit öffentlichen Schlüssel von Empfänger **EMail Datenintegrität** Hashwert h der Email erzeugen; Siginieren mit privaten Schlüssel des Absenders; keine Verschlüsselung der E-Mail **Sichere Email** PGP, S/MIME Einfach beides oben machen; zuerst Datenintegrität

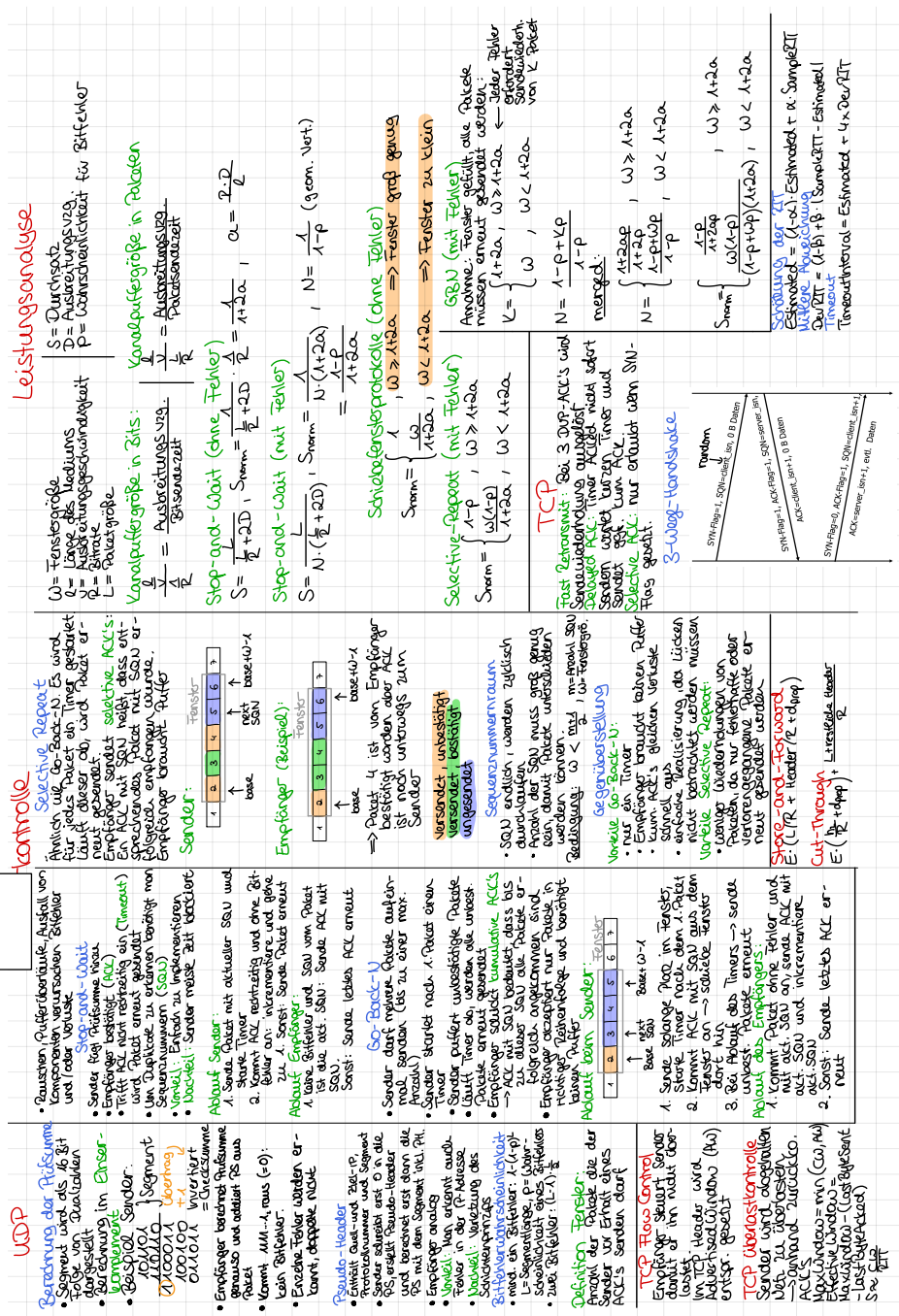
IMAP 143 erlaubt auch Verzeichnisse; Mails bleiben bei Server **POP3** TCP 110 Ablauf: S: +OK POP3...; C user bob; S +OK; pass 1234; S +OK; Dann befehle list (Liste der Mails) retr <nr> ...; retr <nr> (lädet email), dele nr; am ende C quit; S +OK

Netzwerkmanagement Überwachung und Verwaltung eines Netzwerks = komplexes HW/SW-Gebilde **FCAPS-Modell** Fault Management (Monitoring, Dokumentation) Configuration Management (Übersicht über Geräte und Configs) Accounting Management (Verwendung von Netzressourcen, Zugangs von Benutzern und Geräten) Performance Management (Auslastung, Durchsatz, Antwortzeiten) Security Management (Kontrolle des Zugangs, Filterregeln für Firewalls, Intrusion Detection) **Simple Network Management Protocol (SNMP)** UDP Kann obige Daten sammeln. Managing Entity Prozess auf zentralen Steuerrechner; Managed Device (Gerät im Netz); Managed Object (HW oder SW im Managed Device); Management Agent (Prozess auf Managed Device, kann lokale Aktionen ausführen); Schlussendlich Manager entscheidet, was zu tun **SNMP Messages** Get-Request M(anager) an A(gent) um Daten abzufragen; Get-Next-Request M→A nächsten Datensatz, Zugriff auf sequentielle Datensätze; Get-Bulk-Request M→A, für mehrere Datensätze auf einmal; Set-Request M→A init oder ändert den Wert eines Datensatzes; Response A→M Antwort auf Get und Set-Nachrichten; Trap A→M unaufgeforderte Nachricht über Fehlersituation

DNS Hostname→IP-Adresse; Root Name Server → Top-level Domain (.com, .de); **Datensätze** (Domainname, Wert, Typ, TimeTo-Live); Typ=A(AAA)→Wert=Ipv4(Ipv6)Typ = NS→Wert=Domain von Host, der Domain kennt; Typ=CNAME→Wert=Alias für Domain; Typ=MX→Wert=domain auf dem der zugehörige Mailserver läuft **AnfrageAntowortprotokoll** Kopf: Art d. Anfrage, Zurodnung Anfrage/Antwort Rumpf: Questions, Answers, Authority (Antwort des DNS-Server, der IP schickte) **Anfragearten** iterativ: direkt auflösen, oder anderen DNS-Schicken (kein zw.speichern) rekursiv: auflösen u.A mit Datensätzen von anderen DNS-Servern

Content Distribution Networks CDN Spiegelserver (geographisch verteilt) statt einem großen. Ähnlich zu Cache, Inhalt wird aber je nach beliebtheit verbreitet (z.B. YouTube) **Verteiltung der Anfragen** Server-basierte HTTP Redirection (server leitet weiter) URL-Rewriting (Eingebettete Urls umschreiben) DNS-basierte Redirection; meist mit DNS und URL-Rewriting in echt.

P2P Inhalte werden auch von Clienten mitverteilt→Overlaynetz **Unstrukturiert** Zentralisiert: Zentrales Verzeichnis; Dateien über P2P-Netz; Bei Eintritt eines Peers informiert Zentralen Server mit seiner IP und Inhalte; Rein P2P: Anfragen durch fluten zum suchen nach Dateien (max. Hopcount), braucht veröffentlichte Liste möglicher Peers; Hybrid: Peers machen Gruppen mit Groupleader **Strukturiert** Jeder Peer p hat Datenelement k mit Schlussl S_k bei sich




```
TCP-Server:
ServerSocket s = new ServerSocket(PORT);
while(true) {
        Socket connect = s.accept();
        BufferedReader r = new BufferedReader(new
        InputStreamReader(connect.getInputStream(), "UTF-8"));
        BufferedWriter w = new BufferedWriter(new
        OutputStreamWriter(connect.getOutputStream(), "UTF-8"))
        connect.close();
}
```

```
UDP-Client:
byte[] rcv = new byte[1024];
  byte[] snd = "Du schaffst das.".getBytes("UTF-8");
  DatagramSocket s = new DatagramSocket();
  InetAddress a = InetAddress.getByName(name);
  DatagramPacket rp = new DatagramPacket(rcv, rcv.length);
  s.receive(rp);
  DatagramPacket sp = new DatagramPacket(snd, snd.length, a, p);
  s.send(sp);
  String l = new String(rp.getData(), 0, rp.getLength(), "UTF-8");
  int rcvInt = ByteBuffer.wrap(rp.getData(), 0, rp.getLength()).getInt();
  s.close();
```

```
String.parseInt() //in Integer
String.substring(x) // String ab x
String.startsWith("PORT")
```

```
UDP-Server:
  DatagramSocket ss = new DatagramSocket(port);
  byte[] rcv = new byte[1024];
  byte[] snd = "Ja, das schaffe ich.".getBytes("UTF-8");
  byte[] sndInt = ByteBuffer.allocate(4).putInt(1337).array();

  while (true) {
        DatagramPacket rp = new DatagramPacket(rcv, rcv.length);
         ss.receive(rp); // Auslesen zu String wie bei udpClient
        InetAddress ip = rp.getAddress(); int p = rp.getPort();
        DatagramPacket sp =
        new DatagramPacket(snd, snd.length, ip, p);
        ss.send(sp);} ss.close()
```

```
TCP-Client:
Socket clientSocket = new Socket("hostname", PORT);
BufferedWriter w = new BufferedWriter(
new OutputStreamWriter(clientSocket.getOutputStream(),
"UTF-8"));
BufferedReader r =
new BufferedReader(new
InputStreamReader(clientSocket.getInputStream(), "UTF-8"));
clientSocket.close();
```

Physikalische Adresse (MAC-Adresse, LAN-Adresse): 48 Bit (6 Byte oder 12 Hex-Ziffern eingebrannt in den ROM des Adapters; Global eindeutig und ohne Strukturierung

IP-Adresse -> Physikalische-Adresse über ARP (Adress Resolution Protocol): Jeder Knoten besitzt eine ARP-Tabelle, in der alle Teilnehmer mit IP-Adresse, MAC-Adresse und TTL (Time To Live / Gültigkeitsdauer des Eintrages) zugeordnet sind; z.B.: Knoten A will Rahmen an B senden, sendet also ARP-Anfrage als Broadcast (Adresse FF-FF-FF-FF-FF-FF) mit seiner Physikalischen Adresse und der IP-Adresse von B -> B erkennt sich als Ziel an der 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 -> Autonomität (*"Soft-State"*)

Bitfehler: Ursache z.B. thermisches Rauschen oder elektromagnetische-/ radioaktive Einstrahlung; im Bereich 10^{-3} (Funk) bis zu 10^{-12} (Glasfaser); treten oft in Schüben (Bursts) auf

Fehlererkennung: an die Nutzdaten werden Prüfdaten hinzugefügt um Fehler beim Sender zu erkennen und Sendewiederholung zu veranlassen z.B. Paritätsprüfung, Zyklische Redundanzprüfung

Fehlerkorrektur: Nutzdaten redundant Kodiert; Empfänger kann Fehler erkennen und beheben. z.B. mit Block- und Faltungs-Codes. Redundanz größer als für Fehlererkennung; einsatz typisch in stärker gestörten Kanälen und bei hoher Latenzanforderungen; Hybrider ansatz möglich

Zyklische Redundanzprüfung Eigenschaften bei der Fehlererkennung: Alle Einzelbitfehler, falls Koeffizienten von x^i und x^0 gleich 1; Alle Doppelbitfehler, falls G unzerlegbaren Faktor mit min. 3 Termen enthält; Jede ungerade Bifehlerzahl, falls G den Faktor (x+1) enthält; Fehlerbursts die kürzer als r sind

Möglichkeiten für den Mehfachzugriff: 3 Verschiedene Verfahren:

Feste Kanalaufteilung: Frequenzmultiplex (Frequency Division Multiplex Access, FMDA): Geräte verwenden verschiedene Teile des Frequenzspektrums; **Zeitmutliplex (Time Division Multiple Acces, TDMA)**: Geräten werden Zeitschlitze zugewiesen; **CDMA (Code Division Multiplex Access):** Spreiztechnik; jedes Bit der Sender wird mit Chipping-Code multipliziert; die gespreizten Signale überlagern sich auf dem Medium; Empfänger extrahiert mithilfe des Chipping-Codes das einzelne Signal daraus; ***Alternativ***: Frequensprungverfahren: Sender sendet Bits auf jeweils verschiedenen Frequenzen, was Überlagerung der Signale ermöglicht; Anhand des Sprungmusters kann das Signal beim Empfänger korrekt empfangen werden; **FAZIT:** Feste Kanalaufteilung ist nicht sehr Effizient, da kommunikation schubartig ist und die mittlere Verzögerung im Puffer Groß ist

Zufälliger Medienzugriff: Knoten sendet Rahmen soweit vorhanden mit voller Bitrate des Mediums; Gleichzeitiges Senden führt zu Kollision und gegenseitige Zerstörung der Pakete; -> Sendewiederholung

ALOHA: alle Knoten an gemeinsamen Funkkanal; Erhält die MAC-Schicht ein Datagram, so wird der Rahmen sofort gesendet; falls das Paket fehlerfrei ankommt - > ACK; wenn nach einem Timeout kein ACK zurückkommt, wartet der Sender eine zufällige Zeit (Backoff) und wiederholt dann das Senden; Ähnlich wie Stop-and-Wait; geeignet für zB. Satelliten; Maximaler Durchsatz 18%! S_{max}= 1/2e

Slotted ALOHA: alle Knoten synchronisieren ihre Slots z.B. durch zentrales Zeitsignal; Sendebeginn nur zu Beginn eines Slots, somit verkürzt sich das Kollisionsintervall auf einen Slot; doppelter Durchsatz wird erreicht (Maximaler Durchsatz 38%!) S_{max}= 1/e

CSMA (Carrier Sense Multiple Access): Knoten prüfen vor dem Senden ob Medium belegt; reduziert Kollisionen (immer noch möglich falls ein anderer Knoten startet, bevor sich das Signal auf dem Medium zu ihm ausgebreitet hat); CSMA/CA S_{max}= 1/1+6.4a

1-persistent: 1.) falls Medium frei, sendet Knoten sofort; 2.) falls Medium belegt, wartet Knoten bis es frei ist und geht dann zu 1.; 3.),falls nach dem Senden kein ACK kommt, geht der Knoten in Backoff, danach zu 1.

Geringere Wartezeit; aber sichere Kollision, falls mehrere Knoten auf freies Medium warten

Nicht-persistent: 1.) Wenn das Medium frei ist, sendet der Knoten sofort; 2.) Wenn das Medium belegt ist oder nach dem Senden kein ACK kommt, geht der Knoten in Backoff, danach zu 1.

-Weniger Kollisionen, aber längere Wartezeit

P-persistent: 1.) Wenn das Medium frei ist, sendet der Knoten jeweils mit Wahrscheinlichkeit p oder wartet noch einen Slot mit Wahrscheinlichkeit 1-p; 2.) Wenn das Medium belegt ist, wartet der Knoten bis es frei ist; 3.) Wenn nach dem Senden kein ACK kommt, geht der Knoten in Backoff, danach zu 1.

Kompromiss; Binary Expansion Backoff = random(0, ...2^m-1) * t, wobei m bei jeder Kollision inkrementiert wird

CSMA/CD: Knoten besitzen HW, um während des Sendens Kollision zu erkennen; nach Erkennung einer Kollision wird das Senden abgebrochen und eine Jamming-Signal gesendet damit alle Knoten die Kollision sicher Erkennen; keine ACKs; **Kollisionsfreiheit:** L/R > 2D; S_{max}= 1/1+4.4a

Zyklische Zuteilung: Polling: Sendeerlaubnis wird den Knoten nacheinander erteilt (eingeteilt durch zB Zentralen Knoten oder Protokoll); *Zykluszeit*: Zeit bis Sendeerlaubnis zum Knoten zurückkommt; - Overhead, zentraler Knoten ist "Single-Point-of-Failure"; **Token Ring:** Knoten sind ringförmig vernetzt; Listen Mode und Transmit Mode

Ethernet: Rahmenformat: Präambel (8 Byte): 7 Bytes bestehend aus 1010101010 zur Synchronisation der Taktfrequenz; 1 Byte 10101011 Symbolisiert den Beginn der Zieladresse; Quell und Zieladresse: 6 Byte; Type: 2 Bytes nummer für Protokoll TCP..; Nutzdaten: 46-1500 Byte; CRC: 4 Bytes; Gesamtgröße minimal 64 Byte ohne Präambel; **Medienzugriff** über 1-persistent CSMA/CD; binärer exponentieller Backoff; Kein Handshaking erforderlich (verbindungslos); kein versenden von Bestätigungen (unzuverlässig); minimale Sendezeit muss größer gleich zweifacher Ausbreitungszeit im Medium sein!

Topologien: Bus: S_{max} = 1/(1+4.4a); Bestehend aus Repeater (Auffrischung von Signalen, operiert auf physikalischer Schicht; transparent für Medienzugriffsschicht; **Sterntopologie mit Hub:** alle an einem Zentralen Hub, verbunden mit Twisted-Pair; Bitrate 10 Mbps;CSMA/CD; **Sterntopologie mit Switch:** Store and Forward; voll-duplex; Kaskadierung/ Heterogenität von Bitraten/ Kombination mit Hubs möglich; Kollisionen treten nicht mehr auf obwohl Knoten CSMA/CD durchführen; möglich

Fast Ethernet: Sterntopologie, Hubs, Switches

Selbstlernen: Ein Switch entscheidet, wohin er eingehende Frames weiterleitet: Er verwirft Frames mit der physischen Zieladresse am Eingangsport, flutet unbekannte Adressen an alle Ports, speichert die Zuordnung von Adresse und Port in einer Tabelle und verwaltet dies als Soft State mit beispielsweise einer TTL von 60 Minuten.

Spanning Tree Protocol: Mit Switches können zyklische Strukturen im Netzwerk entstehen, was das Selbstlernen behindern kann; Alle Switches in einem LAN führen einen verteilten Algorithmus durch, um einen aufspannenden Baum (Spanning Tree) zu erstellen (STP, IEEE 802.1D); Die Idee besteht darin, den Root Switch anhand seiner MAC-Adresse zu bestimmen, wobei jeder Switch den kürzesten Pfad zum Root Switch ermittelt und nur auf den Ports entlang dieses Pfads weiterleitet; Das Rapid Spanning Tree Protocol (RSTP, IEEE 801.1D-2004) ermöglicht eine schnellere Konvergenz im Vergleich zu STP (von 30 Sekunden auf unter 1 Sekunde).

Virtuelle LANs: Virtuelle LANs (VLANs) dienen dazu, die Broadcast-Domäne zu segmentieren, z.B. für ARP und DHCP; VLANs ermöglichen die Aufteilung in scheinbar verschiedene LANs mit unterschiedlichen Subnetzen; Hauptziele sind die Aufteilung, Lastoptimierung und Anpassung der logischen Netztopologie an die Unternehmensstruktur (z.B. Arbeitsgruppen, Benutzermobilität); Switch-Konfiguration erfolgt über Management-Software, basierend auf Ports, MAC-Adressen oder Protokollinformation; Port-basierte Konfiguration: Endgeräte an bestimmten Ports gehören zu einem VLAN; Rahmen werden nur innerhalb des VLANs von Switch weitergeleitet; Inter-VLAN-Verkehr erfordert einen Router; Tagging (Markierung): Der IEEE 802.1Q-Standard beschreibt VLAN-Tags in Ethernet-Frames; Rahmen mit VLAN-Tags werden zwischen Switches ausgetauscht, was VLANs über mehrere Switches ermöglicht.

Drahtlose LANs: erhöhte Fehlerrate im Vergleich zur Übertragung über Kabel (insbesondere Bursts) durch zB. Dämpfung, Interferenzen und Mehrwegausbreitung

Hidden-Terminal Problem: A, B hören sich; C, B hören sich; A, C hören sich nicht -> bekommen mögliche Kollisionen bei B nicht mit

Architektur eines Infrastrukturnetzes: **Station (STA):** System mit Zugriffsfunktion auf das drahtlose Medium und Funk-Kontakt zum Access Point; **Zugangspunkt, Access Point (AP):** Station, die sowohl in das verbindende Netz als auch das Funk-Lan integriert ist; **Basic Service Set (BSS):** Gruppe von Stationen mit gleicher Funkfrequenz ; **Portal:** Übergang in ein anderes Netz ; **Distribution System (DS):** Verbindung der APs über schicht 2

Architekur eines Ad-Hoc-Netzes: Station (STA): System mit Zugriffsfunktion auf das drahtlose Medium; **Independent Basic Service Set:** Gruppe von Stationen, die dasselbe Funkfrequenz Nutzen; zufällige MAC-Adresse als BSSID

MAC- Rahmenformat: Rahmensteuerung: 2 Bit für Type;.. 4 physikalische Adressen: Sender, Empfänger usw.; Sequenznummern für ARQ; Daten, CRC

Basic Access Medienzugriff: ähnlich zu CSMA/CA erweitert um DIFS (Distribution Interframe Space) und SIFS (Short Interframe Space) -> Zeit die gewartet wird um jeweils den Rahmen oder das ACK zu senden

Backoff-Mechanismus: 1.) CW = CWmin; 2.) Nach jedem Sendeversuch CW := (CW+1)*2-1; 2.) Bis CWmax erreicht wird, danach bleibt CW konstant; 3.) Wartezeit: gleichverteilt {0, ..., CW} Slot Times; 4.) Unterbrechung des Dekrementierens wenn Medium belegt

RTS/CTS-Austausch: Vorheriger austausch von kurzen Reservierungsnachrichten; Sender sendet Request-To-Send mit Länge des Rahmens, Empfänger, (Station bei Ad-Hoc-Netz, AP bei Infrastrukurnetz); Antwort mit Clear-To-Send in der die Länge steht; Medienzugriff für RTS mit Basic Access

-größerer Overhead

+ Kollisionen werden durch reservierung vermieden, und die Kollisionen die Auftreten sind nur kurz (da RTS nicht lang)

+ Hidden-Terminal-Problem teilweise gelöst: Stationen, welche den Sender nicht hören erfahren vom Empfänger Reservierung

Wired Equivalent Privacy (WEP): ähnliche Sicherheit wie bei leitungsgebundener Kommunikation durch Verschlüsselung; Umsetzung mit symmetrischen 40- oder 104-Bit Schlüssel in Station und Zugangspunkt; für jedes Paket wird 24 Initialisierungsvektor (IV) erzeugt und unverschlüsselt im Rahmen gesendet; für Daten wird 4-Byte Integrity Check Value (ICV) mittels CRC berechnet; XOR Verknüpfung der Daten mit ICV; sicherheitsprobleme bestehen weiterhin (obsolet)

WPA2: Gegenseitige Authentifizierung via Pre-Shared Key oder Authentifizierungsserver; Sicherheitsprotokoll mit Counter Mode und Cipher Block Chaining Message Authentication Protocol. Sicherheitslücke entdeckt in WPA3 seit 2018.

Leistungsanalyse: Wahrscheinlichkeit dass ein Knoten in einen beliebigen Slot ohne Kollision Sendet: $N * p(1-p)^{2(N-1)}$

Netzwerkschicht:

->Forwarding: Vermittlungseinheit empfängt Dateneinheiten auf einer Schnittstelle (Interface) und leitet sie auf einer anderen weiter

->Routing: Verfahren, mit denen Vermittlungseinheiten entscheiden, über welchen Weg Dateneinheiten gesendet werden sollen

IP-Adressen:

->Klassenbasierte Adressierung:

IPv4 Adresse (32Bit) bestehend aus Netzwerk- und Hostteil.

+ selbstidentifizierende Adressen: an ersten Bits wird erkannt um welche Klasse es sich handelt

- feste Zuordnung von Netzwerken. Wenn Rechner in anderes Netzwerk zieht, muss seine IP-Adresse angepasst werden.

->Klassenlose Adressierung:

Hostanteil wird unterteilt in Subnetz. Subnetzadresse: IP-Adresse & Maske in Binär, dann AND und wieder dezimal. .0 und .255 immer belegt

Fragmentierung:

Datagramm wird in Fragmente zerlegt und als kleinere Datagramme weitergeleitet, wenn Verbindungen unterwegs eine kleinere Maximum Transmission Unit (MTU) erfordern. IP-Header aus identifier, flag, offset.

ICMP (Internet Control Message Protocol):

Kontrollnachrichten von Routern an andere Router/Hosts (z.B. Benachrichtigung über Fehler). Wird mit IP-Datagrammen befördert. Nicht zuverlässig, da i.d.R nur für Netzwerkdiagnose und -steuerung

DHCP (Dynamic Host Configuration Protocol):

Client-Server Protokoll zum automatischen Bezug einer IP-Adresse (und auch Router, DNS-Server). DHCP-Server muss im Subnetz sein. 4 Schritte (DHCP server discovery, DHCP server offers, DHCP request, DHCP ACK)

NAT:

Umgehen von Adressknappheit (IPv4), indem intern global ungültige Adressen (z.B. 10.0.0.0/24) und nur eine global gültige IP-Adresse verwendet werden. Verbindungen zu internen Hosts werden auf Paare abgebildet, die aus dieser Adresse und einem Port bestehen. NAT-Router führt Abbildungen mit Tabelle durch (Änderungen im Header: überschreibt Adresse, Port, neue TCP Prüfsumme und Adressumsetzung). Größe durch Anzahl von Portnummern begrenzt.

- Verletzung des Schichtenprinzips (Ports sind für Dienste zw. Transport- und Anwendungsschicht gedacht)
- Eingriff in die Ende-zu-Ende-Verbindung

Datagrammbasierte Paketvermittlung:

Jedes Datagramm trägt globale Adresse, die von Routern zur Weiterleitung verwendet wird. Keine Bereitstellung von Dienstmerkmalen wie Fehlerkontrolle, Fluss-/Überlastkontrolle, Informationssicherheit, Bewahrung von Reihenfolge

->z.B. IP

Virtuelle Leitungsvermittlung:

Jede Dateneinheit erhält lokale Kennung, die beim Weiterleiten von jeder Vermittlungseinheit verändert wird. Ggf. Bereitstellung von Dienstgütermerkmalen

IPv6:

Insgesamt 128Bit unterteilt in Gruppen von 16Bit, mit jeweils 4 Hexadezimalzahlen

->Header haben feste Länge, schnelles weiterleiten

->keine Fragmentierung

->Informationssicherheit

->keine Prüfsumme

Routeraufbau:

->Eingangsport: Pufferung, wenn Pakete schneller von der Leitung kommen, als sie weitergegeben werden können. Paketverlust wenn Puffer überläuft. Verteiltes Weiterleiten (Port besitzt Kopie der Weiterleitungstabelle). Effiziente Datenstrukturen für schnelle Suche und inhaltsadressierbarer Speicher (CAM).

->Switching Fabric, Möglichkeiten:

Speicher: CPU kopiert Paket von Eingangsport in Hauptspeicher, führt Weiterleitungsentscheidung durch und kopiert Paket zum Ausgangsport.

Bus: Bus verbindet alle Ports, Eingangsport versieht Paket mit Markierung und sendet sie über den Bus per Broadcast an alle Ausgangsports. Kann nur jeweils für einen Transfer benutzt werden

Verbindungsnetzwerk: Jeder Port kann direkt mit jedem anderen verbunden werden, ermöglicht nebenläufige Weiterleitung

->Ausgangsport: Pufferung, wenn Switching Fabric schneller liefert als Pakete auf die Leitung gegeben werden. Active Queue Management, proaktive Entscheidung, wenn Pakete verworfen werden. Scheduling, wenn mehrere Pakete gepuffert sind, kann entschieden werden, welches als nächstes gesendet wird

IP-Adresse: 192.168.19.4/20

IP-Adresse in Binär:

1100 0000 1010 1000 0001|0011 0000 0100

->20 Bits vorne abzählen -> Subnetzgrenze

Subnetzmaske (1en bis zur Subnetzgrenze):

1111 1111 1111 1111 1111|0000 0000 0000

Subnetzadresse (Subnetzmaske & IP-Adresse verunden):

1100 0000 1010 1000 0001|0000 0000 0000

MPLS:

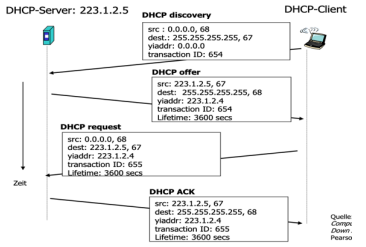
Einsatz bei ISP(Providern). Schnelleres Weiterleiten mit Labeln statt IP-Adressen über Label Switched Path's (LSP)

Übergang von IPv4 zu IPv6:

->Dual Stack: Endsystem mit IPv6 und IPv4 Implementierung, abhängig vom Ziel wird IPv4 oder IPv6 angewendet. Problematisch wenn Router nur mit IPv4 weiterleitet.

->Tunneling: IPv6 Datagramm wird in ein IPv4 Datagramm eingebettet

-> NAT: Datagramm mit privater IPv4 Adresse in IPv6-Paket einbetten und bei NAT-Übersetzung in ein IPv4-Packet mit öffentlicher Adresse wandeln



Routing:

->Intradomain (Interior Gateway Protokolle): Innerhalb einer Routing-Domäne. Hierfür können Verfahren verwendet werden, die nicht für große Netzwerke skalieren. Link-State (Dijkstra) und Distanzvektor (Bellman-Ford-Verfahren)

-> Interdomain (Exterior Gateway Protokolle): Zwischen-Routing Domänen, ausgetauschte Routing-Informationen enthalten ganze Pfade (z.B. Border Gateway Protokoll, BGP)

-> Unicast-Routing (Punkt-zu-Punkt): proaktiv: Information über Netztopologie wird ausgetauscht, aktuell gehalten, mit Graph-basierten

Verfahren werden Pfade zu allen Zielen bestimmt, bei Sendewunsch werden diese genutzt

->Multicast (Punkt-zu-Mehrpunkt): ebenfalls proaktiv

->Ad-Hoc-Routing: dynamische Netztopographie, auch reaktive Verfahren: erst bei Sendewunsch wird Pfad bestimmt

->Datenzentrische Verfahren: adresslos, Nachrichten werden aufgrund ihres Inhalts weitergeleitet

Link-State Routing:

Alle Knoten besitzen vollständige Kenntnis der Netzwerktopologie (durch Fluten).

->zentrales Verfahren

->bei n Knoten Komplexität des Dijkstra Verfahrens: O (n^2)

->effiziente Implementierungen schaffen O(nlogn)

->beschränkt Skalierbarkeit

->Nachrichtenaustausch: O(ne) bei e Kanten

->Robustheit: nur weitergegebene Topologie kann fehlerhaft sein

Distanzvektor-Routing:

Jeder Knoten kennt nur Kosten zu direkten Nachbarn und die von ihm erreichbaren Ziele

->verteilter Algorithmus

->Konvergenzprobleme bei Zyklen

->beschränkte Skalierbarkeit

->Robustheit: Router können fehlerhafte Pfade weitergeben

->Fehlfunktion eines Routers wirkt sich auf andere aus

Mögliches Problem: Count-to-Infinity Problem: veraltete Information in den verteilten Routing Tabellen enthält zyklischen Pfad. Lösung: Poisoned Reverse: Nachbarn auf dem kürzesten Pfad zu einem Ziel wird ∞ als Kosten für dieses Ziel mitgeteilt. Jedoch nur bis Zyklen der Länge 2. Oder: größten Kostenwert beschränken

IPG/EGP:

->IGP in der Regel für kleinere Netzwerke bzw. innerhalb einer Routingdomäne

->IGPs haben begrenzte Topologieinformationen, berücksichtigen nicht immer die besten Pfade in Bezug auf Latenz u. Bandbreite

->EGPs komplexer

->EGP für große Netzwerke ausgelegt

IPsec:

Sicherheit auf Netzwerkschicht, bietet Authentizität des Senders, Datenintegrität der Nachrichten, Vertraulichkeit des Inhalts und von Protokollinformationen. Ermöglicht virtuelle private Netze (VPNs). Erfordert die Einrichtung von Security Associations