Zusammenfassung: Internet

Themenbereich: Mobile IP

- Erreichbarkeit in fremdem Subnetz über gleiche IP (spezieller Router erforderlich)
  - Zustände: **Home Link** (Host im eigenen Subnetz), **Foreign Link** (fremdes Subnetz)
  - Geräte:
    - **Mobile Node**: Host der Mobile IP nutzt
    - **Home Agent**: Router im Heimnetz des Nutzers. Kennt Standort der Mobile Node, er leitet Pakete an Mobile Node weiter.
    - **Foreign Agent**: Router im fremden Subnetz. Kennt den Home Agent und teilt ihm die „care-of"-Adresse der Mobile Node mit (**Tunnel** zwischen Home Agent und Foreign Agent)
  - Methode: **Kapselung** der IP-Pakete in IP-Paketen
  - **Collocated CoA**: IP des Mobile Node im fremden Netz. Wird verwendet wenn kein Foreign Agent Verfügbar. (Tunnel zwischen Home Agent und Mobile Node)
  - **Triangle Routing**: Durch Mobile IP auftretendes Problem. Physikalisch nahe Hosts schicken Pakete u.U. über unnötig lange Wege, da Home Agent weit entfernt.

Themenbereich: IPv6

- Starke Erweiterung des Adressraums im Gegensatz zu IPv4. Notwendig wegen stark steigender Zahl an Hosts durch mobile Endgeräte.
- Wichtige Header-Felder:
  - **Priority**: für QoS, Congestion Control, Packet Dropping
  - **Flow Label**: Pakete mit dem gleichen Label gehören zu einem Flow
  - **Payload Length**: Menge der Nutzdaten; falls 0 Extension Header: Jumbo Hader
  - **Next Header**: Gibt Typ des nachfolgenden Headers an
  - **Hop Limit**: vergleichbar mit TTL des IPv4 Headers
  - **SA, DA**: Quell- und Zieladresse jeweils nun 128 Bit lang
  - **Hop-By-Hop Header**: Optionen, welche von jedem Router ausgewertet werden müssen: z.B. Jumbo Header
  - **Routing Header**: Legt fest welche Route das Paket durch das Netz nehmen soll
  - **Fragment Header**: für Fragmentierung
  - **Authentication Header**: Authentisierung
  - **Encapsulating Security Payload**: Verschlüsselung und Authentitätsprüfung der Nutzdaten.
  - **Destination Options Header**: Optionen, die nur vom Empfänger ausgewertet werden.
- **Mobile IPv6**:
  - Unterschiede zu Mobile IPv4
    - **Knoten** können die aktuelle CoA kennen und **senden dann direkt**; ansonsten über **Home Agent** → Vermeidung des **Triangle Routings**

- **Binding Update und Acknowledgement**: Dieser Mechanismus dient der **Bekanntmachung der aktuellen IP-Adresse des Mobilknotens für einen Empfänger** oder den **Home Agent.**
  - **Update**: Informiert dem Empfänger über die CoA
  - **Acknowledgement**: Bestätigt das Binding
  - **Request**: Anforderung eines Updates; lässt sich an Paket anhängen **(Piggybacking)**
- Wann ist Binding sinnvoll?
  - Wenn der **Empfänger Mobile IPv6 versteht**
  - Wenn **Authentisierung** möglich ist
  - **Datenaustausch** beabsichtigt ist
  - Wenn ein Algorithmus das **Binding sinnvoll steuert**

Themenbereich: TCP-Protokoll

- Stärke und gleichzeitig Schwäche von TCP ist seine **Verbindungssicherheit**
  - Stärke äußert sich durch das sichere Ankommen der Pakete beim Adressaten
  - Paket Senden → Paket Bestätigen, sonst Paket nochmal senden
  - **Flussteuerung über Fenstergröße** (Erlaubte Sendebytes ohne Bestätigung)
    - 2 Phasen: **Slow Start** (exponentiell), **Congestion Avoidance** (linear)
    - Congestion Avoidance beginnt ab einem **Schwellwert**
    - **Paketverlust**: Erneuter Slow Start → niedrigerer Schwellwert → Fenstergröße steigt langsamer → Datenrate steigt langsamer
    - Annahme: **Paketverluste entstehen durch Überlast** einer oder mehrerer Netzkomponenten.
    - Problem: Drahtlose Übertragung erzeugt häufig **Bitfehler ganz ohne Überlastung** → Flussteuerungsalgorithmus wirkt bremsend.
    - Lösung: Angepasste TCP-Verfahren

Themenbereich: DNS – Domain Name Service

- Dient der Zuordnung von schwer Merkbaren **IP-Adressen** zu leicht Merkbaren **Hostnamen**
- **Baumstruktur** der Server-Weiterleitung:
  - ROOT → TOP LEVEL DOMAINS → SUBDOMAINS → ...
  - www.hs-weingarten.de = <host>.<domain>.<top-level-domain>
- Realisiert durch **verteilte Datenbank**. Die Hoheit über die Subdomains wird **delegiert**
  - ROOT kennt DNS-Server für Domain „de" und schickt deshalb die Anfragen an diesen Server weiter.
- Die **„de"-Domain**: Verwaltet von der **Organisation DENIC**
  - BelWÜ ist Mitglied der **Genossenschaft**
  - „hs-weingarten" ist über BelWÜ registriert
    - DNS-Server der Hochschule zuständig für alle Subdomains von „hs-weingarten.de"
- **Umkehrung der Abfrage** über in-addr.arpa: IP-Adresse → Hostname; **Verwaltung durch InterNIC**

- **Zonen:** Hoheitsgebiet eines DNS-Servers, **keine Überlappung**, = nicht delegierter Teil einer Domain, DNS-Server **antwortet auf Anfragen aus eigenen DNS-Records**, enthält mindestens 2 DNS-Server
- **Namensauflösung:**
  - **Rekursive Abfrage**: Stete Weiterleitung der Anfrage → **Antwort bei Erfolg**
  - **Iterative Abfrage**: Server Antwortet mit **Verweisen auf mögliche DNS-Alternativen**, wenn er den Namen selbst nicht kennt und nicht im Cache hat. Client fragt selbst weiter.
  - **Caching**: Zugriff auf bekannte Domain-Namen verkürzt. **Begrenzte Speicherdauer**
- Ressource Records:
  - **Name**: Domainname (URL)
  - **TTL**: Zeitangabe wie lange die Zuordnung im Cache bleiben soll (1d oder 86400)
  - **Klasse**: Normalerweise „IN" für Internet
  - **Type**: Art des Records
    - **A**: IP-Adresse [64Bit – Adresse]
    - **AAAA**: IPv6-Adresse [128Bit – Adresse]
    - **CNAME**: Alias für Host [Domainname]
    - **HINFO**: Systemangaben über Host
    - **MX**: Mailserver der Domain [16 Bit Präferenzwert + Hostname]
    - **PTR**: Zuordnung IP-Adresse → Hostname [Domainname]
    - **NS**: Verweis auf weiteren DNS-Server (Name Server)

Themenbereich: SMTP - Simple Mail Transfer Protocol

- Verbindungsablauf
  - Client: Aufbau einer TCP Verbindung zum Server
  - Server: Statuscode + Begrüßung
  - Client: Begrüßung EHLO (alt HALO) + Angabe des Absenders
  - Server: Statuscode + Unterstützte SMPT-Erweiterungen
  - Client: Nennt den Absender (Email-Adresse)
  - Server: Statuscode
  - Client: Nennt Adressaten (mehrere möglich)
    - RCTP TO: <Email-Adresse>
  - Server: Statuscode
  - Client: Ankündigung Datenübertragung
  - Server: Statuscode
  - Client: Zeilenweise Übertragung der Email; Letzte Zeile enthält nur „."
  - Server: Empfangsbestätigung: 250 message accepted
  - Client: Beendet Sitzung mit QUIT
  - Server: Statuscode (Bestätigt den Verbindungsabbau)
- Wichtige Kommandos:
  - **EHLO**: Extended HALO; Client-Rechnername als Parameter (nicht immer zwingend)
  - **MAIL FROM**: Parameter = Absenderadresse
  - **RCTP TO**: Parameter = Empfängeradresse
  - **DATA**: Einleitung der Nachrichtenübertragung; max. Zeilenlänge 998; Zeilenende mit CRLF (Carriage Return Line Feed)

- **RSET:** Abbruch der aktuellen Transaktion
- **VRFY:** Server soll Empfänger Bestätigen; **normal deaktiviert** um Spammern keine Datengrundlage zu bieten.
- **HELP:** Server Soll Hilfe an Client senden
- **QUIT:** Verbindung abbauen

- Der Ablauf muss immer gleich sein: **Client fragt → Server antwortet**. Vor der Antwort darf der Client nicht erneut fragen.
- Ereignisklassen: 1. Stelle des Statuscodes beschreibt die Klasse die beiden anderen beschreiben den Fall näher
  - **1XX:** Vorläufig positive Antwort
  - **2XX:** Endgültig positive Antwort
  - **3XX:** Positives Zwischenergebnis
  - **4XX:** Temporärer Fehler
    - Kommando nicht ausgeführt → nochmal versuchen
  - **5XX:** Fataler Fehler
    - Kommando nicht ausführbar → bleiben lassen
- Auffinden des Mail-Servers (local_part@domain.topleveldomain)
  - **local_part**: Username auf dem Mailserver
  - Auflösen des **Hostname/Domainname mit DNS**
    - **MX** Records nach ihren **Prioritäten** wählen
    - Wenn kein MX Record vorhanden, Suche nach Hostnamen
- Sicherheitskonzepte
  - Mails können **nur intern verschickt** werden, Mails werden **nur von intern akzeptiert**
  - **Verschlüsselung** der normal im Klartext übertragenen Daten
  - **Nutzerauthentifizierung**, Challenge Response (ESMTP)
  - **Echtheitsprüfung:** Zuordnung von Mail zu Absender (ESMTP)
  - **Integrität**: Keine unbemerkte Veränderung der Nachricht (ESMTP)
- MIME – Multipurpose Internet Mail Extension
  - **SMTP unterstützt nur 7-Bit US-ASCII** → Keine Umlaute, Bilder, chinesische Schriftzeichen Übertragbar.
  - **MIME –Headerfelder**
    - **Version:** z.B.: „1.0“
    - **Content**-Type:
      - **Top-Level**: Text, Image, Audio, Video, Multipart (Attachments), Message (Weiterleitung)
      - **Second-Level**: Nähere Beschreibung wie z.B. die Codierung der Daten
      - Beispiele: text/plain; charset=us-ascii; image/jpeg; video/mpeg
      - **Typ Multipart:**
        - **Mixed**: Unabhängige Nachrichtenteile, sollen in Reihe dargestellt werden
        - **Parallel**: Unabhängige Nachrichtenteile, sollen parallel dargestellt werden

        - **Alternative**: verschiedene Versionen der gleichen Nachricht (verschiedene Sprachen); nur eine Alternative soll dargestellt werden
        - **Digest**: Jeder Teil ist eine Eigene Mail
    - **Typ Message**
        - **Rfc822**: Inhalt ist vollständig Mail, wichtig für Weiterleitung
        - **Partial**: Mail-Fragment, wichtig wegen begrenzter Mail-Größe, Parameter: **ID** = beschreibt zusammengehörige Teile, **Number** = Sequenznummer des Teils, **Total** = Gesamtzahl der Teile
        - **External Body**: Nachricht enthält Zeiger auf Datei; Parameter beschreibt wie auf die Datei zugegriffen werden kann; Datei wird nach Empfang der Nachricht heruntergeladen
- **Content-Transfer-Encoding**
    - **Mögliche Codierungen**:
        - **7-Bit**: SMTP-konform
        - **Quoted-Printable**: Auch für nicht us-ascii-Zeichen im Text verwendbar Codierung durch „=HH" wobei HH eine zweistellige Hex-Zahl ist, ggf. Zeilenumbrüche zur Einhaltung der maximalen Zeilenlänge
        - **Base64**: Codierung von Binärdaten
- Mail-Zugriff
    - **POP**: Post Office Protocol
        - Zugriff in **3 Phasen** gegliedert:
            - **Autorisierung**: Username und Passwort senden (bedenklich wenn unverschlüsselt)
            - **Client fordert**: Auflistung, Übertragung, Löschung der Mails; Phase wird mit QUIT beendet
            - **UPDATE**: Aktualisierung der Mailbox auf dem Server; wird **nur bei QUIT** ausgeführt
        - Befehle:
            - **USER**: Benutzername als Argument
            - **PASS**: Passwort als Argument
            - **STAT**: Liefert die Gesamtgröße der Mails
            - **LIST**: Liefert ohne Argument die Liste aller Mails und deren Größen, Argument kann eine spezifische Mail sein, dann wird nur deren Größe gemeldet
            - **RETR**: Argument ist eine Mail, diese Mail wird übertragen
            - **DELE**: Argument ist eine Mail, diese Mail wird gelöscht
            - **NOOP**: Nichts tun
            - **RSET**: Delete rückgängig machen
            - **QUIT**: Transaktion beenden, Schließen der TCP-Verbindung
    - **IMAP**: Internet Message Access Protocol
        - **Mails** bleiben i.d.R. **auf dem Server**
        - Viele Verwaltungsfunktionen
            - Mehrere Mailboxen

- Mehrere Ordner mit versch. Zugriffsrechten
- Benachrichtigungsdienste

- Mail-Attribute
  - Unique Identifier (UID)
  - Message Sequence Number
  - **Flags**: Seen, Answered, Delete, Draft, Recent
  - Datum und Zeit des Empfangs
  - Größe der Nachricht
- Befehle für jeden Zustand
  - **CAPABILITY**: Vom Server unterstützte Fkt. Abfragen
  - **NOOP**: Nichts tun
  - **LOGOUT**: Abmelden
- Befehle wenn nicht authentifiziert
  - **STARTTLS**: Transportschichtverschlüsselung aktivieren
  - **AUTHENTICATE**: Argument ist das verlangte Authentifizierungsverfahren, Authentifizierung mit diesem Verfahren
  - **LOGIN**: Argument sind Benutzername und Passwort im Klartext
- Befehle wenn authentifiziert
  - **SELECT**: Auswahl einer Mailbox mit Lese-und Schreibrechten
  - **EXAMINE**: Auswahl einer Mailbox mit Leserechten
  - **CREATE**: Erzeugung einer Mailbox, Name der neuen Mailbox als Argument
  - **DELETE**: Löschen einer Mailbox, Name der zu löschenden Mailbox als Argument
  - **RENAME**: Mailbox umbenennen
  - **SUBSCRIBE**: Ordner Abonnieren
  - **UNSUBSCRIBE**:Ordner Abbestellen
  - **LIST**: Auflisten aller verfügbaren Mailboxen
- Befehle wenn Mailbox ausgewählt
  - **CLOSE**: Mailbox schließen
  - **EXPUNGE**: Alle als /Deleted gekennzeichneten Mails löschen
  - **SEARCH**: Suche nach Mails
  - **FETCH**: Liefert Daten einer Mail oder ganze Mail
  - **STORE**: Ändern von Attributen einer Mail
  - **COPY**: Argumente: Ausgewählte Mails, Zielordner; kopiert Mails in angegebenen Ordner
- Protokollvergleich

	POP	IMAP
Speicherung	Benutzer PC	Server
Bearbeitung	Offline	Online

Mehrere Boxen	Nein	Ja
Teilweises Laden	Nein	Ja
Komplexität	Gering	Hoch
Verwendung mehrerer Clients	Problematisch	Voll unterstützt

- Spam:
  - **Unerwünschte Werbemails**
  - Versendet wird über **SMTP Relay Server**: Mail **mit vielen Adressaten**
  - Absenderadresse i.d.R. gefälscht
  - Es werden fremde Rechner zum versenden verwendet, da diese Rechner die Verteilung übernehmen und damit die Kosten dafür tragen
  - **Spam-Anteil der Mails zeitweise bei 90%**
- Gegenmaßnahmen:
  - Server darf **nur Mails des eigenen Netzes** annehmen
  - **Rückverfolgung** von Mails
  - Email Adresse nicht bekannt werden lassen (**nicht online stellen, oder nur manipuliert angeben**)
  - **Graylisting**: Emailempfang verzögern
  - **Blacklisting**: öffentliche Liste von Mail-Versendern, Spamdatenbanken → Abfrage durch Mailserver
  - **Spamfilter**: Heuristische Verfahren (Suche nach Begriffen) → Fehlzuordnungen, Ohne Zustimmung des Nutzers rechtlich problematisch
  - Spam-Versender ergreifen wiederum Gegenmaßnahmen
- Rechtsfragen bei Spam
  - Persönlichkeitsrecht, **Unterlassung, Schadenersatz**
  - Unerwünschte Mails sind **Wettbewerbswidrig**
  - Unterlassungsanspruch, **Ordnungsgeld, Ordnungshaft**
  - **Viren und Trojaner** werden oft per Mail verbreitet
    - Geschieht dies mit Absicht ist man wegen **Sabotage** strafbar

Themenbereich: HTTP – Hypertext Transfer Protocol

- **TCP-Verbindungen werden aufrecht erhalten**, da auf eine Anfrage meist eine weitere folgt.
- Requests:
  - **OPTIONS**: Optionen einer Netzwerkressource oder eines Servers abfragen
  - **GET**: Liefert die Informationen der nachfolgenden URI
    - Bedingungen: If-Modified-Since, If-Unmodified-Since, If-Match, If-NoMatch, If-Range
    - **Teile** einer Ressource lassen sich **über das Range-Feld** abfragen
  - **HEAD**: Wie GET, jedoch nur Abfrage des Headers; **für Testzwecke**
  - **POST**: Verschicken einer Nachricht an den Server
  - **PUT**: Hochladen einer Internetseite; wie Post aber Speicherung unter angegebener URI
  - **DELETE**: Löschen einer Datei
  - **TRACE**: Zurückschicken der Nachricht an den Absender (ECHO); **für Fehlersuche**

    - **CONNECT**: Noch keine Verwendung vorgesehen
- Von diesen Befehlen **müssen nur GET und HEAD beherrscht werden** können.
- Aufbau von HTTP-Nachrichten
    - **Startline**: Ist Request oder Status, gefolgt von CRLF
    - **Header**: General-, Request-, Response-, Entity-Header
    - General-Header-Felder
        - **Cache Control**: Caching Direktiven
        - **Connection**: Verbindung soll nach der Antwort geschlossen warden
        - **Datum**
        - **Pragma**: Für HTTP1.0 Kompatibilität, z.B. no-cache, veraltet
        - **Trailer**: Es gibt im Anhang weitere Header-Felder
        - **Upgrade**: Fordert Server auf das Protokoll zu ändern → HTTP2.0
        - **Via**: Protokolliert Gateways und Proxies
    - Request-Header-Beispiele
        - **Accept Language**: gibt bevorzugte Sprache an
        - **Accept**: Legt Medientypen fest, die der Versender der Nachricht akzeptiert
        - **Accept Charset**: Legt Schriftsatz fest, den der Versender akzeptiert
        - **Accept Encoding**: Legt die Mediencodierung fest, die der Versender akzeptiert
        - **Content-Type**: Medientyp des Inhalts
        - **Host**: Host der Abgerufenen Ressource
        - **User Agent**: Infos über die Client-Software (Browser)
    - Response-Header-Beispiele
        - **From**: Email-Adresse des für den Inhalt verantwortlichen
        - **Age**: Alter des Inhalts
        - **Allow**: Sender gibt die Methoden an die unterstützt werden
        - **Accept-Ranges**: Sender unterstützt den Abruf von Teilen des Inhalts
        - **Retry-After**: Angabe wann Ressource wieder Verfügbar ist
        - **Server**: Serverinformationen
- Aufbau von HTTP-Requests
    - Methode: OPTIONS, GET... gefolgt von Space
    - Request URI gefolgt von Space
    - HTTP-Version gefolgt von CRLF
- Aufbau von HTTP-Responses
    - HTTP-Version gefolgt von Space
    - Status-Code gefolgt von Space
        - Status Code sehr ähnlich zu SMTP Status: 5 Klassen, 1.Ziffer = Klasse, 3Ziffern pro Code, der Client muss die erste Ziffer auswerten.
    - Reason-Phrase gefolgt von CRLF
        - Erläuterungen bei Fehlermeldungen, muss durch den Client nicht angezeigt werden.
- Caching:
    - Warum: Reduzierung der **Netzlast**, weniger **vollständige Responses**, Reduzierung der **Antwortzeiten**
        - Problem: Wann ist der **Inhalt des Caches veraltet** → **Expiration, Validation**

- **Methoden**: **Client-Seitig**, Serverseitig
  - Speichern von abgerufenen Daten im Cache → Erneuter Aufruf → Aus dem Cache laden:
  - Häufig Cache Hierarchien: Auf dem Client → Im LAN →Beim ISP
  - **Serverseite**: Server-Replication, Content Delivery Networks
- Expiration
  - **Ablaufdatum** wird explizit angegeben: Expires-Header-Feld
  - **Heuristische Ermittlung** über die last modified time
  - **Cache-directive**
- Validation
  - Inhalt des **Cache kann validiert werden → Bedingte Anfrage** starten, wenn **Bedingung erfüllt**, dann vom Content-Server **neu laden**. Ansonsten nur Status-Antwort „Not Modified"
  - **Feststellung der Gültigkeit** über den „**last-modified**"-Wert, oder ETag (Entity Tag)
- Nachrichten denen Expiration oder Validation – Informationen fehlen werden nicht im Cache gespeichert.
- Cache Control Header enthält Anweisungen zur Cache-Verwaltung
  - **No-cache**: muss validiert werden wenn im Cache gespeichert
  - **No-store**: darf nicht gespeichert werden
  - **Public**: darf gespeichert werden
  - **Private**: nur im privaten Cache speichern
  - Expiration spezifische Cache direktiven
    - **Max-age**: Gültigkeit in Sekunden
    - **Min-fresh**: Anforderung von Informationen mit Mindestgültigkeit
    - **Max-stale**: Anforderung von Informationen, deren Gültigkeit nicht länger als Max-stale abgelaufen ist
  - Validation spezifische Cache direktiven
    - **If-Match**: Überprüft die Aktualität von Entity-Tags
    - **If-None**-Match: s. If-Match
    - **If-Modified-Since**
    - **If-Range** Sende Teile einer Ressource, wenn Ressource nicht verändert, sonst sende alles, kann **nur mit Range Header** auftreten
- Authentifizierung:
  - **Basic**: Anmeldung mit Username und Passwort (Base64-Übertragung im Klartext, nur mit TLS-Verschlüsselung ratsam)
  - **Digest Authentication Scheme**
    - Server sendet **zufällige Bitfolge** im Authenticate-Header
    - Client berechnet aus **Passwort und Zufallszahl** einen Hashcode
    - Server prüft **Authentizität durch Hash-Code**

Themenbereich: HTML – Hypertext Markup Language

Sollte man sich in praktischer Arbeit zu Gemüte führen

Themenbereich: Drahtlos ins Web

- Auftretende Probleme:
  - **Geringe Bandbreite**
  - Displays mit **kleiner Auflösung**
  - **Wenig Speicher** im Endgerät
  - **Geringe CPU-Leistung**
- Protokolle
  - **WDP – Wireless Datagramm Protokoll**: Einheitliches Protokoll das Unabhängigkeit vom Trägerdienst herstellt → Quell- und Zielportnummern, WCMP = Pendant zu ICMP
  - **WTLS – Transportschichtverschlüsselung** für Drahtlose Endgeräte. Keine hohe Sicherheit
  - **WTP – Wireless Transaction Protocol**
    - **Kein Verbindungsaufbau**
    - Zuverlässigkeit durch: Entfernen von Duplikaten, Sendewiederholungen, Bestätigungen
    - PDUs: **Invoke, Result, Acknowledgement**
    - **Klasse 0**: Sende nur Invoke (unzuverlässig)
    - **Klasse 1**: Sende Invoke und warte auf Acknowledgement (sicherer Request)
    - **Klasse 2**: Sende Invoke, warte auf Acknowledgement, warte auf Result, Bestätige Empfang des Result (sicherer Transfer von Request und Response)
  - **WSP/B: Wireless Session Protocol/Browsing**
    - Verbindungsaufbau und Abbau; Parken und Wiederaufnahme
    - Aushandeln der Fähigkeiten
    - Codierung/Kompression der Inhalte
    - HTTP1.1 Funktionalität mit Binärcodierung (kleinere Datenmengen)
    - Austausch von Sitzungsinformationen insb. HTTP-Header, welche für die Sitzung gültig bleiben
    - Push und Pull möglich
    - Asynchrone abfragen
    - Nutzung der verschiedenen Transaktionsklassen des WTP
- **WML – Wireless Markup Language**
  - Optimiert für: niedrige Bandbreite, kleine Displays, Eingeschränkte Eingabemöglichkeiten
  - **Dokument:**
    - Definiert durch **mehrere Karten (Stapel)** → aber nur **1 URL** → wird als **Einheit** geladen
  - WML-Skript ist die WAP-Variante des Java Skript
- WAP 2.0
  - Unterstützt auch die gängigen Internet-Protokolle, da Mobile Datenrate immer größer wird: TCP/IP, TLS, HTTP → Ergänzungen (Wireless Profile)
  - **Trägernetze**: GSM/GPRS und SMS für Push-Dienste, Neu: UMTS
  - **Transportdienste**: TCP (mit kompatiblen Anpassungen) oder WDP
  - **Transferdienst**: HTTP oder WTP und WSP
  - Push und Pull möglich, Multimedia-Messaging, Browser-Plugins,

- Gegenüberstellung des alten und neuen WAP

WSP	HTTP
WTP	TLS
WTLS	TCP
WDP	IP
Trägerdienst (GSM/GPRS/UMTS)	

Themenbereich: FTP – File Transfer Protocol

- Protokoll zur Dateiübertragung
  - Client-Server-Protokoll: Bidirektionale Datenübertragung
  - **Authentifizierung** erforderlich
    - **Username und Passwort**
    - **Anonymous/ftp** als Passwort sollte dann die Email-Adresse hinterlassen werden
  - Out of Band Signalisierung: **2 TCP-Verbindungen:** Daten und Befehle sind getrennt
- Software
  - **Server:** Protocol Interpreter (liest und schreibt auf der Befehlsverbindung); data transfer process greift auf das Dateisystem zu und übermittelt Daten über die 2. Verbindung
  - **Client:** Wird über Bedienoberfläche gesteuert → Befehle an der Protocol Interpreter → Initiierung von Datentransfers → Rückmeldung auf der Bedienoberfläche
- Verbindungsablauf:
  - **Open:** starten einer Verbindung
  - **User:** Identifizierter Nutzer oder Anonymous
  - **Pass:** wird i.d.R. automatisch abgefragt
  - **Spezifizierung eines Transfermodus:** ASCII (evtl. Ersetzung von Steuerzeichen) oder Binary
- Kommandos:
  - Starten eines File-Transfers: **Get/MGet, Put/MPut**
  - 2 Modi
    - **Aktiv:** Client öffnet Port → Sagt Server den Port → Server öffnet Verbindung
    - **Passiv:** Client sendet PASV an Server → Server Öffnet Port und sendet Portnummer → Client öffnet TCP-Verbindung
  - CLOSE und QUIT schließen die Verbindung, bei CLOSE wird der Client nicht beendet
  - Status-Messages sind an das HTTP-Protokoll angelehnt

Themenbereich: Multimedia Im Internet – Transportschichtprotokolle

- Echtzeiterfordernisse, große Verzögerungen unzulässig
  - Kein TCP wegen: Sendewiederholungen, Congestion Control
  - Kein UDP wegen: schlechte Datensicherung, verschleierte Paketverluste, Paketvertauschung
  - →Man braucht ein Echtzeitprotokoll
- RTP – Real-Time Transport Protocol
  - Zeitstempel: für Taktrückgewinnung und Synchronisation

- Erkennung von: Paketverlusten, Paketreihenfolge, Verdopplungen,
- Keine Sendewiederholungen
- Kennzeichnung der Datenkodierung (Art der Nutzdaten)
- Identifikation der Versender
- Verwendet für: Sprach- und Videodaten im Internet
- RTCP: Pendant zu ICMP
- Eingebettet in UDP-Pakete

- RTCP – Sendereports
  - Rückkopplung über Verbindungsqualität → Steuerung adaptiver Codecs →Auswahl anderer Codecs
    - Paketverlustinformationen
    - Jitter
  - Identifikation des Senders durch Canonical Name (user@host)
  - Anpassung der Senderate an Anzahl der Teilnehmer
  - Übermittlung von Zusatzdaten
  - RTCP-Verbindung ist Bidirektional
  - RTP-Ports sind immer geradzahlig → RTCP-Port ist RTP-Port + 1
- Anforderungen an Echtzeitverbindungen
  - Delay: Durch Codierung/Decodierung, Signallaufzeit, Paketbildung, Warteschlangen
    - Warteschlangenverzögerung ist variabel → Jitter
    - Jitter-Ausgleich Td = Tmax – Tmin
    - Tolerierbare Verzögerungen: 150ms (kaum hörbar) – 400ms (mit Verlusten)
    - Konsequenzen: Echo, Rede-Gegenrede-Störungen, Verbindungsabbruch
  - Paketverluste: Durch Bitfehler (insb. Mobilfunk), Überlauf (insb. IP)
    - Verschleierung im Empfänger möglich (concealment)
    - Verschlechterung der Qualität
    - 3,5% Verlust sind erträglich
- QoS im Internet
  - Realisierung: Integrated Service (IntServ)
    - Reservierung von Ressourcen auf Netzkomponenten
    - Protokoll RSVP
    - Unterschiedliche Bearbeitungsklassen für IP-Pakete
      - Kennzeichnung durch DSCP von 6 Bit (6 höherwertigen Bits des ToS-Feldes [IPv4], Traffic Class-Feldes [IPv6]
      - Expedited Forwarding [101110]
      - Assured Forwarding → 12 Klasen
      - Best Effort [000000]
      - Network Control Traffic
  - Differentiated Services (DiffServ)
    - Vereinbarungen zwischen Nutzer und Netzbetreiber
    - Host setzt die DSCP-Bits
    - Edge-Router wertet diese Bits aus
      - Kontrolle: Hat Benutzer die Rechte für diese Bitkombination
      - Ggf. Korrektur der Bits
      - Verkehrsmessung, Shaping, Policing

- RSVP – Ressource Reservation Protocol
  - Protokoll zur Reservierung von Netzressourcen → nur unidirektional → Festlegung des Weges der Pakete durch das Netz
    - Erzeugung einer virtuellen Verbindung → Punkt- zu Mehrpunkt möglich.
  - Token-Bucket-Modell:
    - Dient der Begrenzung des Datendurchsatzes
    - Behälter hat Maximalgröße x und anfänglichen Füllstand B
    - Wird mit steter Rate R gefüllt
    - Soll ein Paket mit Größe a versendet werden muss der Füllstand größer als a sein
    - Nach dem Versenden von a wird B um a dekrementiert
  - Bewertung:
    - Garantierte Bandbreite, nur wenn das ganze Netz RSVP unterstützt
    - Enormer Overhead für kurze Verbindungen
    - Anwendungen müssen über Qualitätsanforderungen informiert sein
    - Große Anzahl von Flows in einem Router kaum handhabbar
    - Gleichartige Verkehre können zusammengefasst werden
- MPLS – Multi Protocol Label Switching
  - Funktionsprinzip: Autonomes System → Alle Router im Netz kennen sich durch Routing Protokoll (OSPF, RIP) → Labels (zwischen Schicht 2 und Schicht 3) identifizieren Zielrouter → Kein IP-Routing mehr → Switching aufgrund der Labels
    - Reduktion des Routing-Overhead führt zu verbesserter Netzauslastung
    - Ingress-Router (E-LSR) vergibt Label für Core-LSR (Push Operation)
    - Core-LSR vergibt Label für Egress-Router (E-LSR) (Swap Operation)
    - Egress-Router routet Paket mit IP-Routing weiter und entfernt das Label (Pop Operation)
    - MPLS-Schicht 2 Rahmen : Label, S (bottom of stack), TTL; Label ist der wichtigste Eintrag, TTL wie bei IP
- Audio-Streaming
  - Datentransport über UDP/RTP
  - Steuerung über RTSP – Real-Time Streaming Protocol → Anweisungen:
    - DESCRIBE: Abfrage von Medienparametern
    - SETUP: Logischen Kanal zwischen Server und Client aufbauen
    - PLAY: Start der Übertragung
    - RECORD: Start der Übertragung zum Server
    - PAUSE: Unterbrechung der Datenübertragung
    - TEARDOWN: Kanal abbauen
- Standard H.323
  - Gateway:
    - Wandlung von Sprachdaten: Kompression, Paketierung → RTP, Echokompensation, Erzeugung des Comfort Noise, Ausgleich des

Jitter am Ausgangsgateway, Taktregenerierung, Dekompression → PCM-Codierung.

- Anpassung der Signalisierung

- Gatekeeper:
  -