Mobilkommunikation Kapitel 9: Transportprotokolle/Mobile TCP

- Motivation
- TCP-Mechanismen
- Klassische Ansätze
  - Indirektes TCP
  - Snooping TCP
  - Mobile TCP
  - PEP generell
- Weitere Optimierungen
  - Fast Retransmit/Recovery
  - Transmission Freezing
  - Selektive Wiederholung
  - Transaktionsorientiertes TCP
- TCP für 2.5G/3G-Systeme

Transportschicht

Beispiel: HTTP (verwendet bei Web Services) nutzt typischerweise TCP

- Zuverlässiger Datentransport zwischen client und server benötigt

TCP

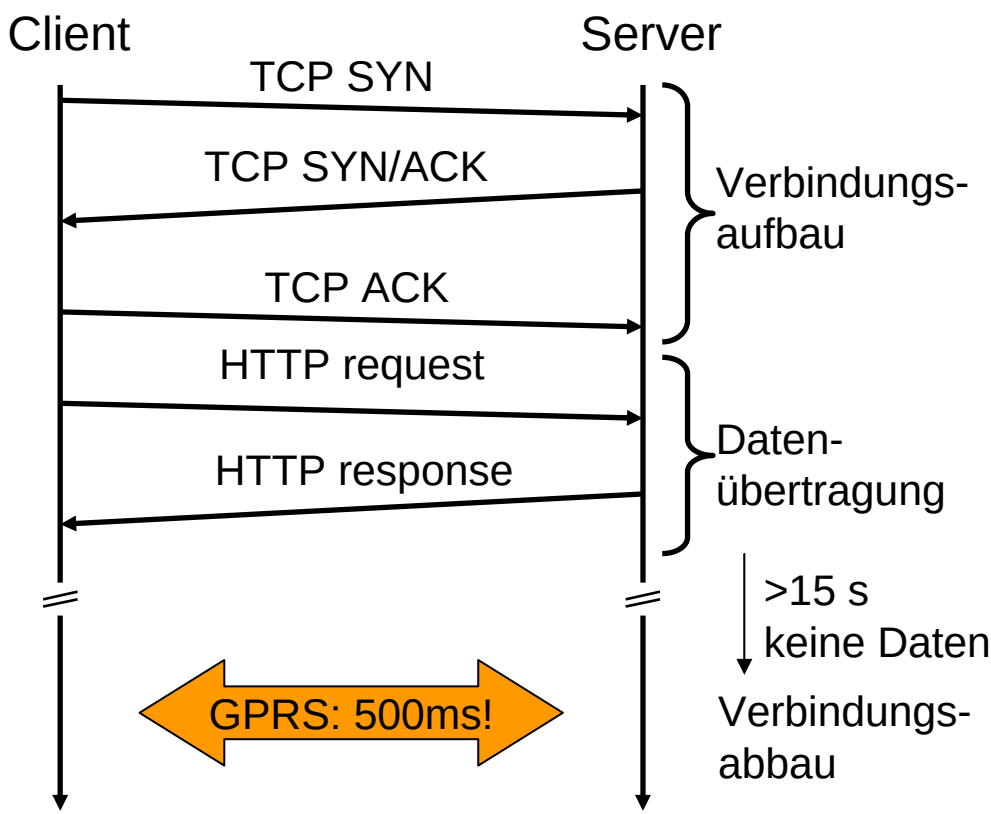
- Stromorientiert, nicht transaktionsorientiert
- Netzwerkfreundlich: bei time-out
  - ➔ Annahme eines Staus
  - ➔ Drosseln der Übertragungsrate

Alt bekanntes Problem – TCP verschätzt sich sehr oft in drahtlosen und mobile Umgebungen

- Paketverlust durch Übertragungsfehler
- Paketverlust durch Netzwechsel

Ergebnis

- Drastische Leistungseinbrüche



Motivation I

Transportprotokolle bisher entworfen für

- Stationäre Endgeräte
- Festnetze

Forschungsschwerpunkte

- Leistungsfähigkeit
- Staukontrolle
- Effiziente Übertragungswiederholung

TCP Staukontrolle

- in Festnetzen entstehen Paketverluste i.allg. durch eine Überlast
- Router müssen Pakete verwerfen sobald ihre Puffer voll sind
- TCP bemerkt Stau nur indirekt anhand von ausbleibenden Quittungen, Übertragungswiederholungen würden nun den Stau nur noch verschlimmern
- Slow-start Algorithmus

Motivation II

TCP Slow-start Algorithmus

- Sender berechnet ein Staufenster für einen Empfänger
- Start mit Fenstergröße gleich 1 Segment
- exponentielles Wachstum des Fensters bis zu einem Schwellwert, dann linear
- bleibt eine Bestätigung aus, so wird der aktuelle Schwellwert halbiert, das Staufenster beginnt wieder mit einem Segment

TCP Fast Retransmit/Fast Recovery

- TCP sendet Bestätigungen nur nach Empfang eines Pakets
- gehen mehrere Bestätigungen für das gleiche Paket ein, so bedeutet dies, das eine Lücke aufgetreten ist, jedoch alle Pakete bis zur Lücke empfangen wurden und weitere Pakete aktuell empfangen werden
- der Paketverlust ist also nicht auf einen Stau zurückzuführen, slow-start wird nicht eingesetzt sondern sofort mit dem aktuellen Fenster weitergesendet

Auswirkung der Mobilität auf TCP-Mechanismen

TCP geht bei Paketverlust von Stau aus

- dies ist meist falsch in drahtlosen Netzen, hier herrschen Paketverluste durch *Übertragungsfehler* vor
- weiterhin kann die *Mobilität* zu Paketverlusten führen, wenn ein mobiler Knoten von einem Zugangspunkt (foreign agent) zu einem anderen geht und Pakete noch zum falschen Zugangspunkt unterwegs sind

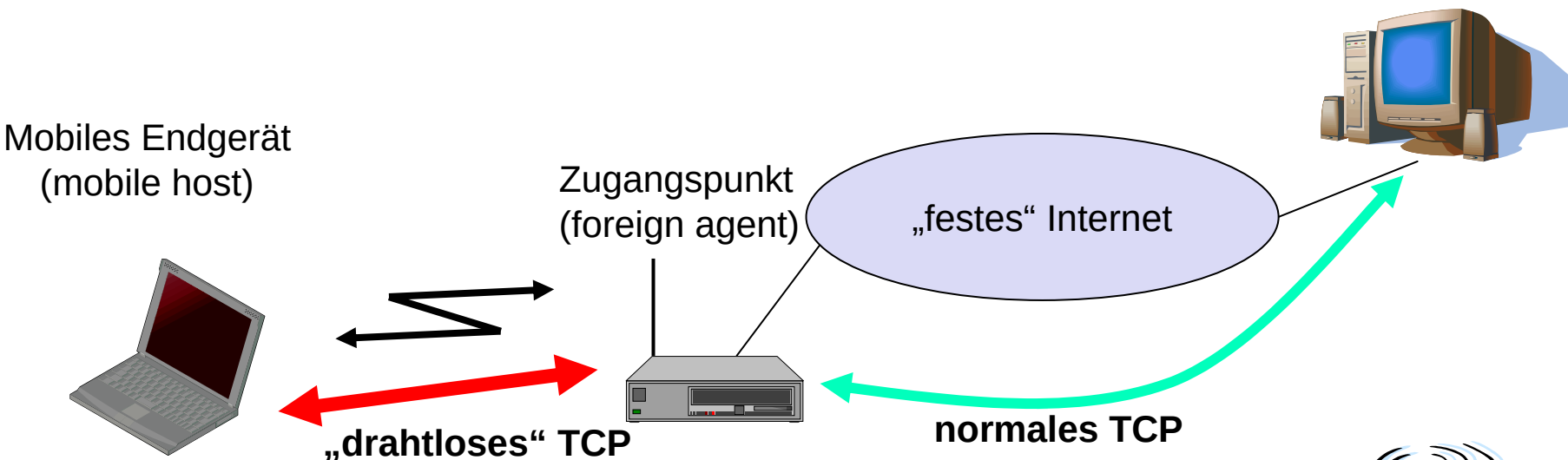
Die Leistung eines unveränderten TCP bricht katastrophal ein!

- TCP kann aber nicht „grundsätzlich“ verändert werden, da Interoperabilität mit Festnetzrechnern notwendig
- TCP-Mechanismen halten im Festnetz das Internet zusammen

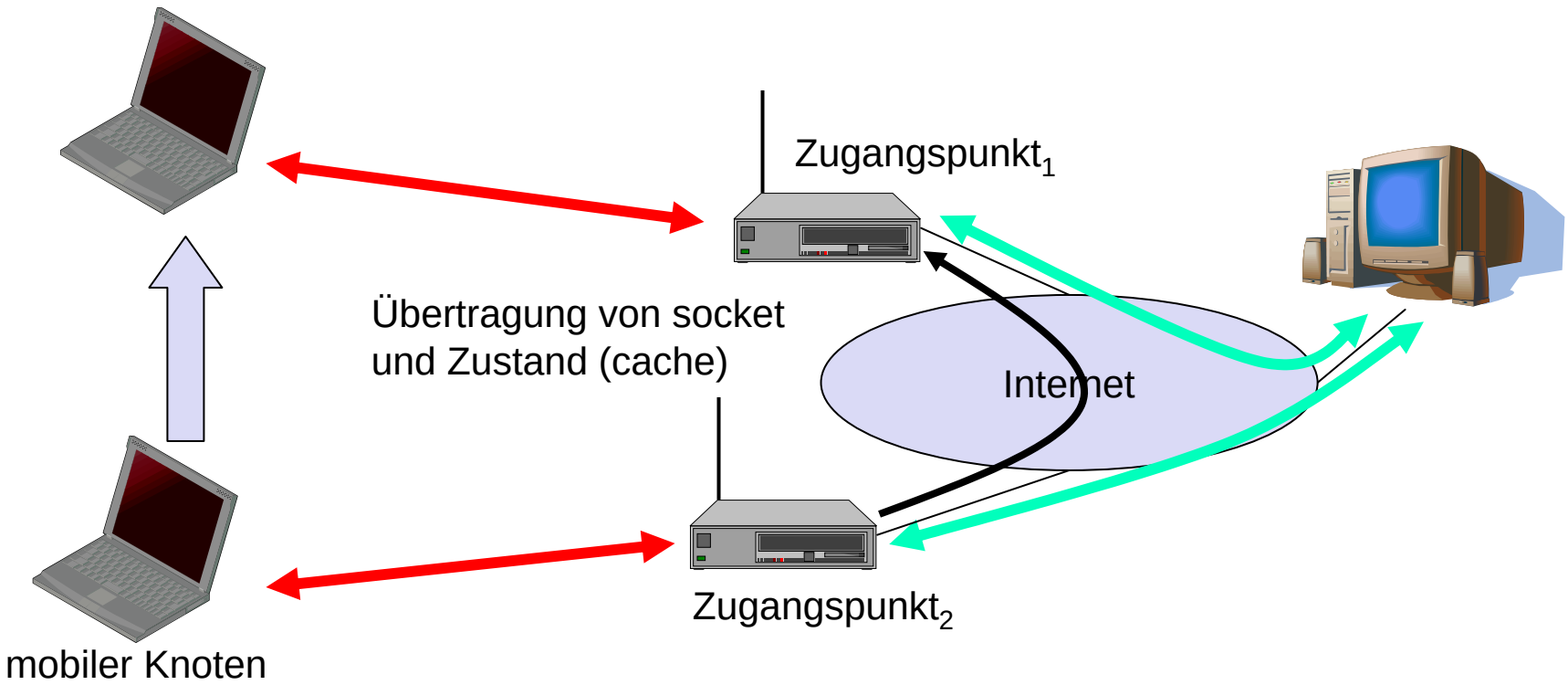
Klassische Ansätze: Indirektes TCP I

Indirektes TCP, auch I-TCP, segmentiert die Verbindung

- keine Änderung am TCP-Protokoll für Rechner im Festnetz, hier ist die installierte Basis zu hoch
- optimiertes TCP-Protokoll für Mobilrechner
- Auftrennung der TCP-Verbindung z.B. am Foreign Agent in 2 TCP-Verbindungen, keine „echte“ Ende-zu-Ende-Semantik mehr
- Rechner im Festnetz bemerken nichts vom mobilen Teil



I-TCP Zustandsübertragung



Indirektes TCP II

Vorteile

- keine Änderungen im Festnetzbereich, alle Optimierungsmaßnahmen helfen hier weiterhin
- Fehler auf der drahtlosen Strecke pflanzen sich nicht ins Festnetz fort
- relativ einfach beherrschbar, da mobile TCP-Varianten nur die kurze Strecke (ein „hop“) zwischen Foreign Agent und Mobilrechner betreffen
- dadurch sehr schnelle Übertragungswiederholung, da Verzögerungszeit auf der Mobilstrecke bekannt ist

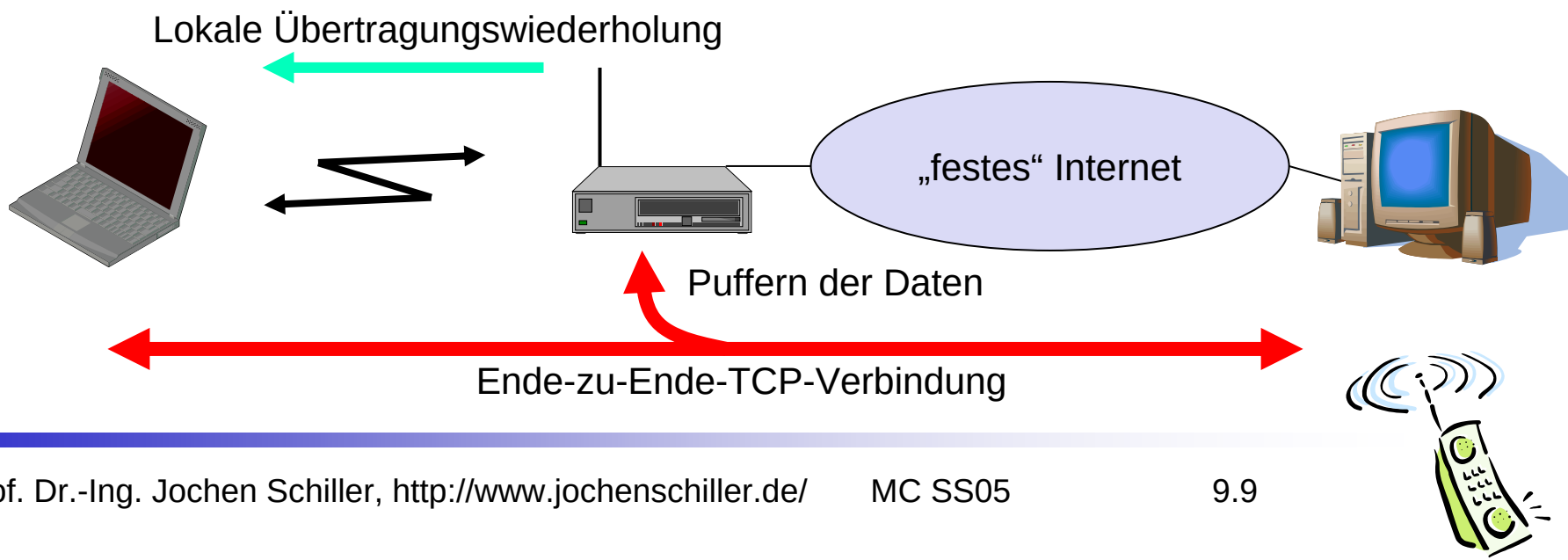
Nachteile

- Verlust der TCP-Semantik, ACK an Sender heißt nun nicht mehr, dass der Empfänger wirklich die Daten erhalten hat - was passiert, wenn der Foreign Agent abstürzt? Konsistenz der Sichten?
- vergrößerte Latenzzeiten durch Pufferung der Daten im Foreign Agent und evtl. Übertragung an den neuen Foreign Agent

Klassische Ansätze: Snooping TCP I

„Transparente“ Erweiterung von TCP im Foreign Agent

- Puffern der zum Mobilrechner gesendeten Daten
- bei Datenverlust auf der Mobilstrecke (beide Richtungen) direkte Übertragungswiederholung zwischen Foreign Agent und Mobilrechner („lokale“ Übertragungswiederholung)
- dazu hört der Foreign Agent den Datenverkehr ab und erkennt Bestätigungen in beide Richtungen (Filtern der ACKs)
- TCP muss nur im Foreign Agent erweitert werden



Snooping TCP II

Datentransfer zum Mobilrechner

- FA puffert die Daten bis zum ACK des MN, erkennt Paketverluste durch duplizierte ACKs oder time-out
- schnelle Übertragungswiederholung, unbemerkt vom Festnetz

Datentransfer vom Mobilrechner

- FA erkennt Paketverluste auf dem Weg vom MN anhand der Sequenznummern, sendet daraufhin NACK zum MN
- MN kann nun sehr schnell erneut übertragen

Integration der MAC-Schicht

- MAC-Schicht hat oft ähnliche Mechanismen wie TCP
- schon in der MAC-Schicht können evtl. Paketduplikate durch Übertragungswiederholungen erkannt und verworfen werden

Probleme

- Snooping TCP isoliert die drahtlose Verbindung nicht so gut
- je nach Verschlüsselungsverfahren ist snooping nutzlos

Klassische Ansätze: Mobile TCP

Spezielle Handhabung längerer und/oder häufiger Unterbrechungen

M-TCP teilt die Verbindung ähnlich wie I-TCP auf

- normales TCP im Festnetz bis zum supervisory host (SH)
- optimiertes TCP zwischen SH und MH

Supervisory host

- keine Pufferung der Daten, keine Übertragungswiederholung
- Überwachung aller Pakete, sobald eine Unterbrechung festgestellt wird:
  - setze Sendefenster auf 0
  - der Sender wechselt dann automatisch in den persistent mode
- der alte oder neue SH öffnet das Fenster wieder

Vorteile

- erhält Semantik, unterstützt Unterbrechungen, keine Zustandsübertragung notwendig bei Wechsel des Zugangspunktes

Nachteile

- Verluste auf der drahtlosen Strecke wirken sich auf das Festnetz aus
- verwendet spezielles TCP auf der drahtlosen Strecke

Fast Retransmit/Fast Recovery

Wechseln des Foreign Agent bedeutet meist Paketverlust

- TCP reagiert mit slow-start obwohl kein Stau vorliegt

Erzwingen des Fast Retransmit

- sobald sich der Mobilrechner bei einem neuen Foreign Agent registriert hat, sendet er bewusst duplizierte Bestätigungspakete aus
- damit erzwingt der Mobilrechner bei den entsprechenden Partnern im Festnetz den Fast Retransmit-Modus
- ebenso wird das TCP auf dem Mobilrechner „gezwungen“ weiterhin schnell zu senden, sobald die neue Registrierung abgeschlossen ist

Vorteil

- einfache Änderungen erzielen große Leistungssteigerung

Nachteil

- weitere Vermischung von IP und TCP, Transparenz des Verfahrens problematisch

Transmission/Timeout Freezing

Mobilrechner können auch relativ lange abgekoppelt sein

- keinerlei Datenaustausch möglich in z.B. Tunnel, Verbindungstrennung bei Überlast
- TCP bricht daraufhin die Verbindung komplett ab

„Einfrieren“ von TCP

- die MAC-Schicht kann oft erkennen, dass ein Verbindungsabbruch bevorsteht
- Signalisierung von TCP über dieses bevorstehende Ereignis
- TCP versucht nun nicht weiter zu senden und nimmt auch keinen Stau an
- erneute Signalisierung bei Wiederaufnahme des Kontakts

Vorteil

- Schema unabhängig von Verschlüsselung, Dateninhalten

Nachteil

- Änderung von TCP auf dem MH, MAC-abhängig

Selektive Übertragungswiederholung

TCP-Quittungen sind normalerweise kumulativ

- ACK n bestätigt korrekten und reihefolgerichtigen Empfang bis n
- treten nun einzelne Lücken im Datenstrom auf, so werden oft unnötigerweise Pakete erneut übertragen

Lösung durch selektive Übertragungswiederholung

- RFC2018 erlaubt Quittung aller empfangenen Pakete, nicht nur der reihefolgetreuen und lückenlosen

Vorteil

- weitaus effizienter
- wird schon häufig im Festnetz genutzt

Nachteil

- etwas komplexere Empfängersoftware, mehr Speicher benötigt
- nicht in allen Implementierungen genutzt

Transaktionsorientiertes TCP

TCP-Phasen

- Verbindungsaufbau, Datenübertragung, Verbindungsabbau
- Aufbau und Abbau nach 3-Wege-Handshake benötigen je 3 Pakete
- selbst für kurze Nachrichten werden so min. 7 Pakete benötigt!

Transaktionsorientiertes TCP

- RFC1644, T-TCP, beschreibt eine TCP-Version, die dies vermeidet
- Verbindungsaufbau-, Daten, und Verbindungsabbaupakete werden zusammengefasst
- dadurch kann mit 2 oder 3 Paketen ausgekommen werden

Vorteil

- Effizienz

Nachteil

- geänderte TCP-Version
- Mobilität nicht mehr transparent

Vergleich der vorgestellten Verfahren

Verfahren	Mechanismus	Vorteile	Nachteile
Indirektes TCP	Auftrennen in zwei TCP-Verbindungen	Isolation der drahtlosen Strecke, einfach	Verlust der TCP-Semantik, erhöhte Latenz
Snooping TCP	Mithören von Daten und Quittungen, lokale Wiederholung	Transparent für Ende-zu-Ende, Integration von MAC	Problematisch bei Verschlüsselung, schlechtere Isolation
M-TCP	Auftrennen in zwei TCP-Verbindungen, Drosseln des Senders über die Sendefenstergröße	Erhalt der Ende-zu-Ende Semantik, kommt mit langen/häufigen Unterbrechungen klar	Schlechte Isolation, höherer Berechnungs-aufwand durch Bandbreitenmanagement
Fast Retransmit/ Fast Recovery	Vermeidung von slow-start nach Verbindungswechsel	Einfach, effizient	Vermischung der Schichten, nicht Transparent
Transmission/ Timeout Freezing	Einfrieren des TCP-Zustands bei Unterbrechung	Unabhängig von Dateninhalten, Verschlüsselung	Änderung von TCP, MAC-abhängig
Selektive Übertra-gungswieder-holung	Wiederholung nur der echt verlorenge-gangenen Daten	Sehr effizient	Etwas komplexere Empfängersoftware, mehr Speicher
Transaktions-orientiertes TCP	Zusammenfassung von Verbindungsauf/-abbau und Datenpaketen	Effizient	Geändertes TCP, nicht mehr transparent

TCP-Verbesserungen I

Ursprüngliche Arbeiten

- Indirect TCP, Snoop TCP, M-TCP, T/TCP, SACK, Transmission/time-out freezing, …

$$BW \leq \frac{0.93 * MSS}{RTT * \sqrt{p}}$$

- max. TCP **B**and**W**idth
- **M**ax. **S**egment **S**ize
- **R**ound **T**rip **T**ime
- loss **p**robability

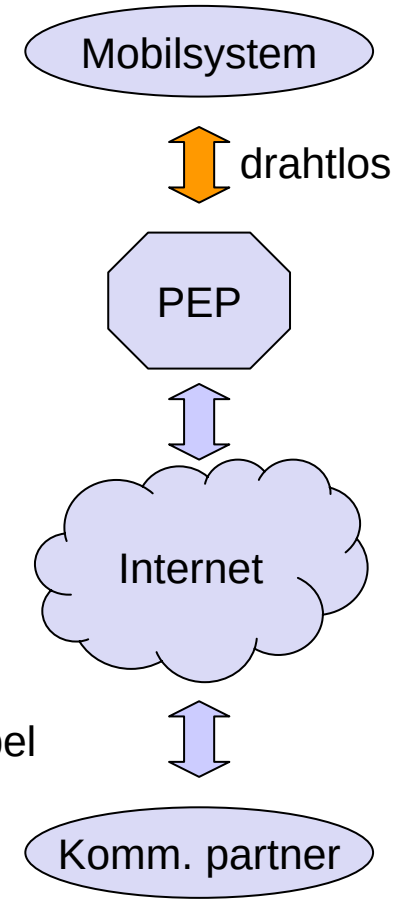
TCP über 2.5/3G Mobilfunknetze

- Optimierung des heutigen TCP
- TCP muss klar kommen mit
  - Datenraten: 64 kbit/s Aufwärtsrichtung, 115-384 kbit/s Abwärtsrichtung; Asymmetrie: 3-6, aber auch bis zu 1000 (Rundfunksysteme), periodische Zuweisung/Freigabe von Kanälen
  - Hohe Verzögerung, hohe Verzögerungsschwankung, Paketverlust
- Verbesserungsvorschläge
  - Große (initiale) Sendefenster Large, große maximale Datentransfereinheiten, selektive Bestätigungen, explizite Staubenachrichtigungen, Zeitstempel, keine Kompression des Protokollkopfes
- Bereits in Verwendung
  - i-mode über FOMA (Freedom of Mobile multimedia Access, NTT DoCoMo)
  - WAP 2.0 (“TCP with wireless profile”)

TCP-Verbesserungen II

Performance enhancing proxies (PEP, RFC 3135)

- Transportschicht
  - Lokale Übertragungswiederholungen und Bestätigungen
- Zusätzlich auf de Anwendungsschicht
  - Inhaltsfilterung, Kompression, Bildskalierung
  - Z.B. Internet/WAP-Gateways
  - Web Service-Gateways?
- Großes Problem: bricht die Ende-zu-Ende-Sematik
  - Verhindert die Nutzung von IP Security
  - Entweder PEP oder Sicherheit!



Weitere offene Gesichtspunkte

- RFC 3150 (slow links)
  - Empfiehlt Kompression von Protokollköpfen, keine Zeitstempel
- RFC 3155 (links with errors)
  - Stellt fest, dass explizite Staubenachrichtigung nicht immer für die gewünschten Zwecke eingesetzt werden kann
- In Kontrast zu den 2.5G/3G-Empfehlungen!