

Kapitel 12 – Mobiles TCP

Vorlesung Mobilkommunikation Wintersemester 2016/17
Prof. Dr. Oliver Waldhorst (HS Karlsruhe), Markus Jung

INSTITUT FÜR TELEMATIK





Mobiles TCP



Mobile Ad Hoc Netze



Mobile IP



WLAN, Bluetooth



GSM, UMTS, LTE



Mobilitätsmanagement



Medienzugriff



Drahtlose Übertragung

- In Festnetzen entstehen Datenverluste i.A. durch Überlast
 - Router verwerfen Dateneinheiten, sobald Puffer voll sind
 - Übertragungswiederholungen würden Stausituation verschlimmern
- TCP wurde für **Festnetze** entwickelt
 - Slow Start zum „Messen“ verfügbarer Ende-zu-Ende-Bandbreite
 - Bei Datenverlust wird Stau angenommen: TCP reduziert Datenrate
 - Fast Retransmit/Fast Recovery bei 4 gleichen Bestätigungen
 - Staukontrollfenster wird auf Hälfte der ausstehenden Daten gesetzt
 - Anschließend direkt Congestion Avoidance (Slow Start entfällt)
 - Slow Start bei Timeout
 - Staukontrollfenster wird auf 1 (oder einen anderen festen Wert) gesetzt



[12.1]

Vorlesung "Telematik"
behandelt TCP im Detail



Mobile TCP

- Stauannahme von TCP im Festnetz i.A. richtig
- Allerdings falsch in drahtlosen mobilen Netzen
 - Drahtlose Netze: Datenverluste meist durch Übertragungsfehler
 - Mobilität: Dateneinheiten zu alter Zustelladresse unterwegs
 - Subnetz-Wechsel impliziert Timeout $\Rightarrow$ Slow Start
 - Sinnvoll (wenn keine Alternative vorhanden), da Ende-zu-Ende-Bandbreite auf neuem Pfad zunächst unbekannt
- Konsequenz für TCP bei Mobilität
 - Slow Start nach jedem Subnetz-Wechsel ist zeitaufwändig
 - Insbes. für Pfade mit hohem Bandbreite-Verzögerungs-Produkt (z.B. Satellitenlinks)
 - Macht sich bei hoher Mobilität stark bemerkbar
 - U.U. langer Timeout, insbesondere wg. Backoff bei „Mehrfach-Timeouts“
 - Übertragungsfehler reduzieren erzielbare Datenrate unnötig
- TCP kann aber nicht „grundsätzlich“ verändert werden
 - Interoperabilität mit Festnetzrechnern notwendig
 - Stau- und Flusskontrolle halten im Festnetz das Internet zusammen

■ Ziel

- Stau- und Flusskontrolle von TCP im Wesentlichen beibehalten
- Zusätzliche Mechanismen
 - bei Übertragungsfehlern in drahtlosen Netzen
 - nach Subnetz-Wechsel bei Mobilität

■ Lösungsansätze

- Erzwungener Fast Retransmit
- Indirektes TCP
- Snooping TCP
- Quick Start für TCP

Erzwungener Fast Retransmit



[12.6]

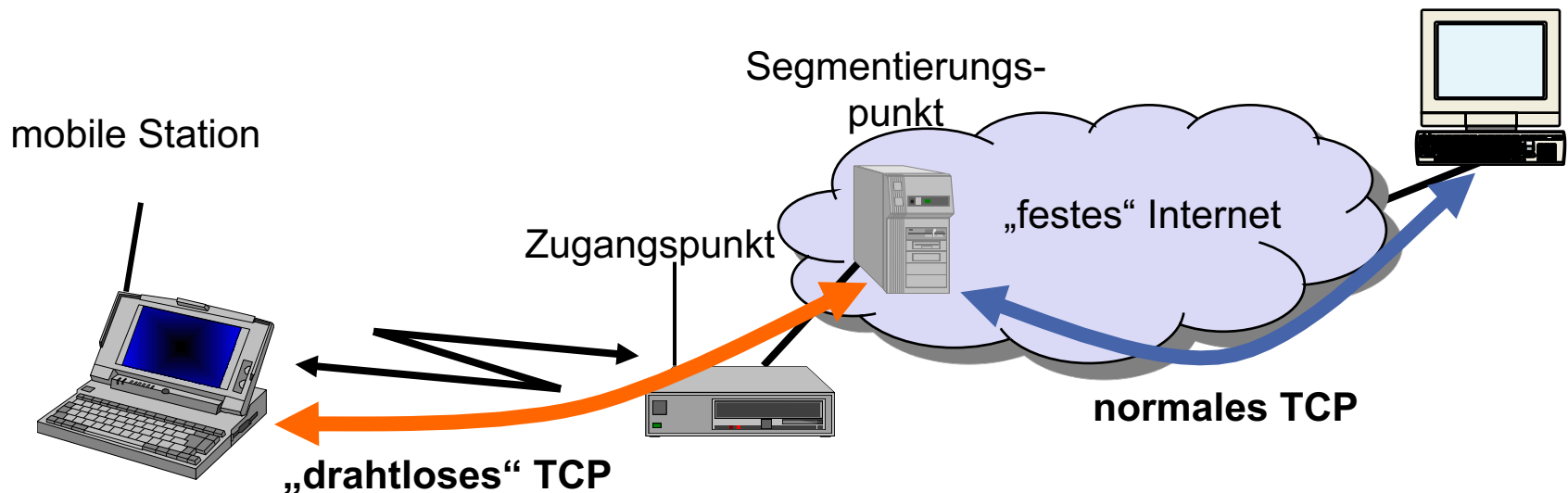
- Problem (am Beispiel Einsatz von Mobile IP)
 - Dateneinheiten werden nach Subnetz-Wechsel zunächst an falsche Zustelladresse gesendet
 - Dateneinheiten gehen verloren
 - Sender erhält keine Bestätigungen
 - Timeout und Slow Start
 - Wiederholung der Dateneinheiten erst nach Timeout
 - Selbst wenn Station erfolgreich im Fremdnetz registriert, Tunnel etabliert, ...
 - Timeout kann lang sein, insbes. wegen Backoff bei "Mehrfach-Timeouts"
- Lösungsmöglichkeit
 - Erzwingen von **Fast Retransmit** durch mobile Station
 - Mobile Station sendet nach Subnetz-Wechsel 4 gleiche Bestätigungen
 - Kommunikationspartner führt Fast Retransmit durch
 - Anschließend **Slow Start** anstatt Fast Recovery
 - Bandbreite auf neuem Datenpfad muss neu bestimmt werden
 - Slow Start automatisch, falls Timeout vor Eingang der Bestätigungen auftrat
 - Ansonsten muss Slow Start "erzwungen" werden

■ Nachteile

- Wiederholte Dateneinheiten legen gesamten Weg durch das Netz zurück
 - Komplette Umlaufzeit für die 4 gleichen Bestätigungen und erste wiederholte Dateneinheit nötig
- Ermittlung der neuen Datenrate über Slow Start langwierig
- Berücksichtigt nur Datenverluste bei Subnetz-Wechsel
 - Durch Übertragungsfehler verursachte Verluste können aber durch reguläres Fast Retransmit/Fast Recovery behoben werden
- Erzwingen des Slow Starts erfordert ggf. Modifikation von TCP am Kommunikationspartner im Festnetz

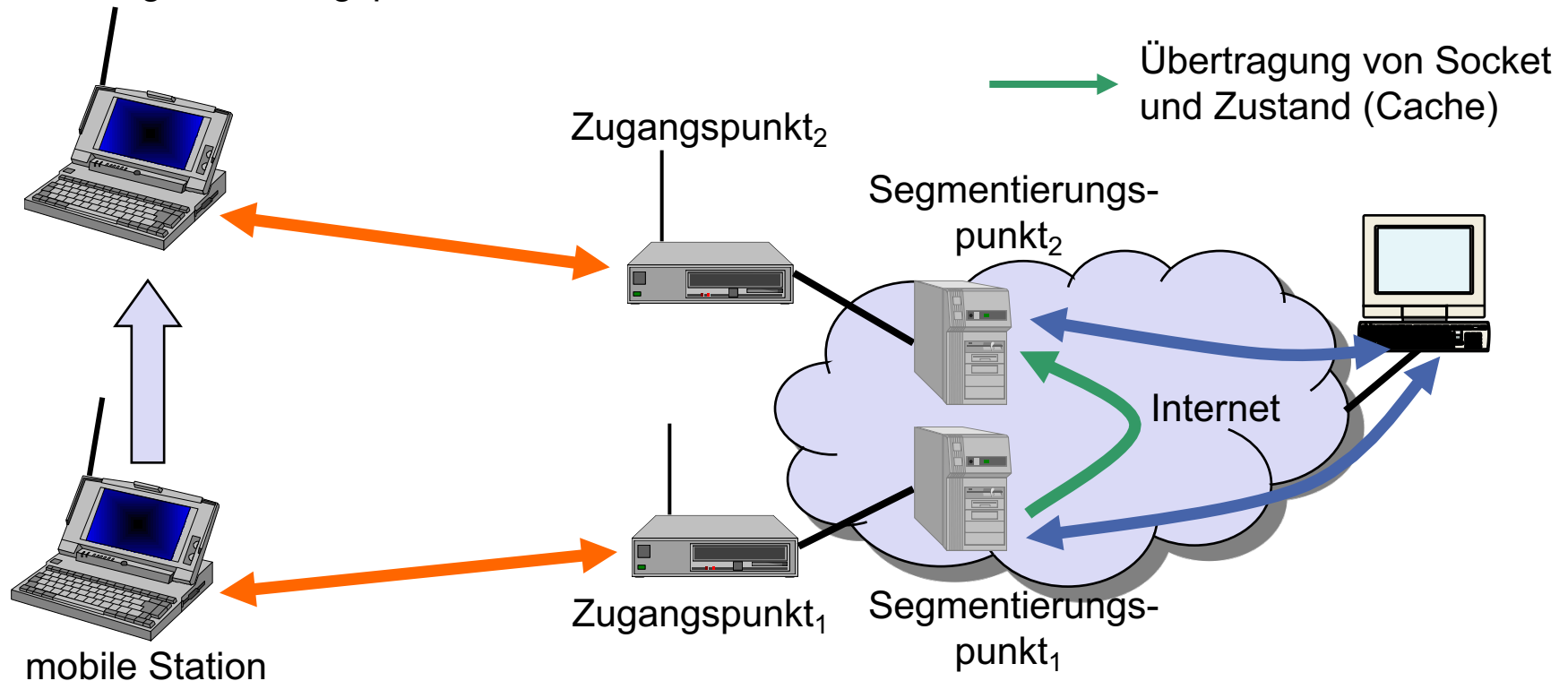
Indirektes TCP

- Aufteilen der TCP-Verbindung in zwei Segmente
 - Aufteilung findet in der Nähe des Übergangs vom Festnetz ins drahtlose Netz statt
 - Zum Beispiel beim Fremdagenten bzw. Zugangsrouter
 - Keine Änderung am TCP-Protokoll für Festnetz-Stationen
 - Installierte Basis ist zu hoch
 - Optimiertes Transportprotokoll zwischen Segmentierungspunkt und mobilem Endgerät („drahtloses“ TCP)
 - Festnetz-Stationen bemerken Subnetz-Wechsel der mobilen Station nicht



■ Optimierung der Mobilstrecke

- Zwischenspeicherung von Dateneinheiten im Segmentierungspunkt
- Schnelle Übertragungswiederholung, da Strecke zwischen Segmentierungspunkt und mobiler Station kurz
- Übertragung des TCP-Zustands bei Subnetz-Wechsel zum neuen Segmentierungspunkt



■ Vorteile

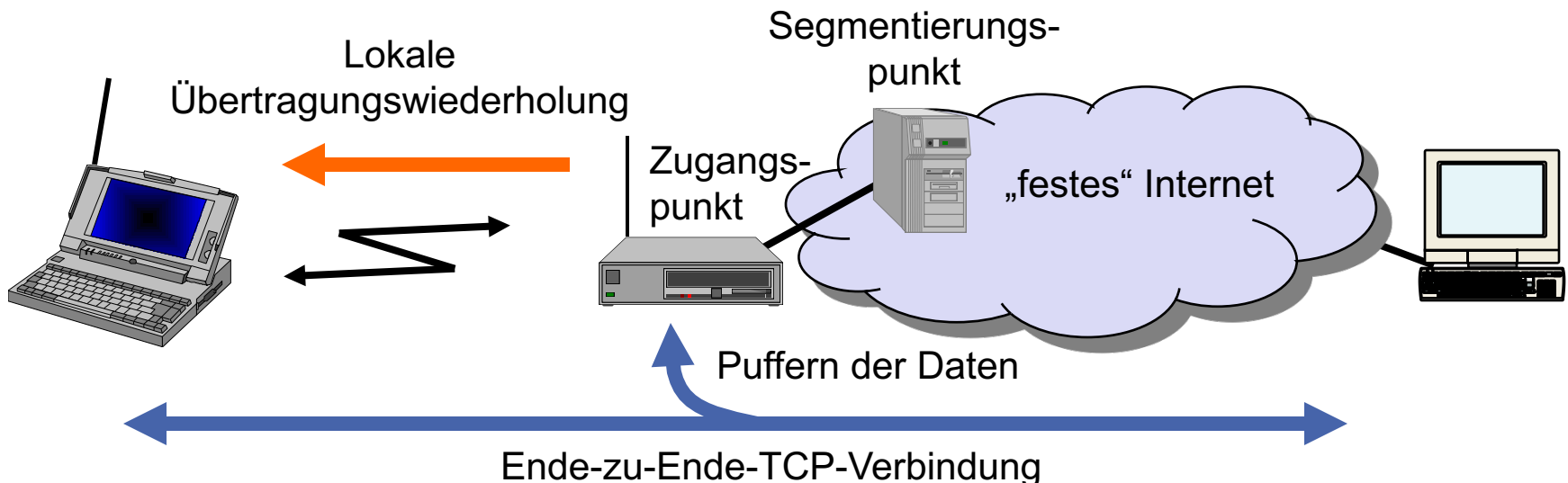
- Keine Änderungen im Festnetzbereich
 - Existierende Mechanismen hier weiterhin effektiv
- Fehler auf der drahtlosen Strecke pflanzen sich nicht ins Festnetz fort

■ Nachteile

- Verlust der Ende-zu-Ende-Semantik
 - Bestätigung an Sender heißt nun nicht mehr, dass Empfänger wirklich die Daten erhalten hat
 - Was passiert bei einem Absturz/Fehlfunktion des Segmentierungspunkts?
 - Konsistenz der Sichten?
- Vergrößerte Latenzzeiten durch Pufferung der Daten im Segmentierungspunkt und evtl. Übertragung an neuen Segmentierungspunkt
- Übertragung von Socket und Zustand (Cache) bei Wechsel des Zugangspunktes notwendig
 - Kurzzeitige Unterbrechung der Datenübertragung über Transportverbindung

Snooping TCP

- „Transparente“ Erweiterung von TCP im Segmentierungspunkt
 - Puffern der zur mobilen Station gesendeten Daten
 - Bei Datenverlust auf der Strecke zwischen Segmentierungspunkt und mobiler Station (beide Richtungen) direkte Übertragungswiederholung durch Segmentierungspunkt („lokale“ Übertragungswiederholung)
 - Dazu hört der Segmentierungspunkt den Datenverkehr ab und erkennt Bestätigungen in beide Richtungen (Filtern der Bestätigungen)
 - TCP bleibt in beiden Endsystemen unverändert



Snooping TCP



[12.2]

[12.3]

- Datentransfer zur mobilen Station
 - Segmentierungspunkt puffert Daten bis zur Bestätigung
 - Erkennt Datenverluste durch duplizierte Bestätigungen oder Timeouts
 - Schnelle Übertragungswiederholung; transparent gegenüber Festnetz

- Datentransfer von mobiler Station
 - Segmentierungspunkt erkennt Datenverluste auf dem Weg von mobiler Station anhand der Sequenznummern, sendet daraufhin negative Bestätigung zur mobilen Station
 - Mobile Station kann nun sehr schnell erneut übertragen

- Probleme
 - Snooping TCP isoliert die drahtlose Verbindung nicht komplett
 - Selbst bei einer lokalen Übertragungswiederholung kann beim Kommunikationspartner ein Timeout auftreten
 - Je nach Verschlüsselungsverfahren ist Snooping nutzlos
 - IPsec verschlüsselt bspw. den Inhalt von IP-Datagrammen und somit auch den TCP-Header. Segmentierungspunkt hat also keinen Zugriff darauf.

Quick Start für TCP

- Problem: Timeout und Slow Start nach jedem Subnetz-Wechsel ist zeitaufwändig
 - Kann bei hoher Mobilität und Datenpfaden mit hohem Bandbreite-Verzögerungs-Produkt Großteil der Kommunikationszeit ausmachen



[12.5]

- Quick Start für TCP hilft, verfügbare Bandbreite *schnell* zu bestimmen
 - Station fragt Router auf Datenpfad nach verfügbarer Bandbreite
 - IP-Option in TCP-Dateneinheit enthält *gewünschte Bandbreite* in Bit/s
 - Zweite IP-Option enthält „Quick Start TTL“
 - Router können Bandbreite in IP-Option reduzieren
 - Quick-Start-fähige Router dekrementieren Quick Start TTL
 - Kommunikationspartner sendet Ergebnis zurück
 - TCP-Option in erster TCP-Dateneinheit
- Quick Start TTL zeigt, ob alle Router Quick-Start-fähig sind
 - Nur dann kann Quick Start angewendet werden
- Station passt Staufenster an verfügbare Bandbreite an
 - Verfügbare Bandbreite * Einschätzung der Umlaufzeit = neues Staufenster
 - Schneller als Bandbreitenermittlung über Slow Start
- Kommunikationspartner führt umgekehrt gleiche Prozedur durch
- Einsatz von Quick Start bei Mobilität zurzeit jedoch noch nicht standardisiert

Quick Start für TCP

■ Vorteile

- Kein zeitaufwändiger Slow Start
- Trotzdem Ausnutzung verfügbarer Bandbreite
- Quick Start auch für andere Transportprotokolle außer TCP geeignet

■ Nachteile

- Beide TCP-Partner müssen Quick-Start-fähig sein
- Alle Router auf dem Pfad müssen Quick-Start-fähig sein
- An ubiquitären Einsatz wird zurzeit nicht gedacht, insbes. nicht in Core-Routern

■ Probleme bei Mobilität

- Nach Subnetz-Wechsel sendet mobile Station i.A. zunächst keine TCP-Dateneinheit
 - Quick-Start-Option muss aber an TCP-Dateneinheit angehängt werden
 - 1 Roundtrip-Zeit erforderlich zum Ermitteln der verfügbaren Bandbreite
 - Mobile Station muss Umlaufzeit auf neuem Pfad kennen, um Staufenster anzupassen
 - Umlaufzeit ändert sich bei Subnetz-Wechsel möglicherweise stark
 - Lösungsansatz: Verwende für Bandbreitenbestimmung benötigte Umlaufzeit
 - Nachteil: Umlaufzeit kann schwanken; ermittelter Wert evt. sehr ungenau
- Noch Forschungsbedarf

Vergleich der vorgestellten Verfahren

Verfahren	Mechanismus	Vorteile	Nachteile
Fast Retransmit/ Fast Recovery	Schnelles Erzwingen einer Übertragungswiederholung nach Verbindungswechsel	Einfach, effizient	Vermischung der Schichten, nicht transparent
Indirektes TCP	Auftrennen in zwei TCP-Verbindungen	Isolation der drahtlosen Strecke, einfach	Verlust der Ende-zu-Ende-Semantik, erhöhte Latenz
Snooping TCP	Mithören von Daten und Quittungen, lokale Wiederholung	Transparent für Ende-zu-Ende	Problematisch bei Verschlüsselung, schlechtere Isolation
Quick Start für TCP	Explizite Auskunft über verfügbare Bandbreite von Routern	Bandbreiten-Bestimmung ohne Slow Start	Beide TCP-Partner und alle Router auf dem Datenpfad müssen Optimierung unterstützen

- TCP wurde nicht für drahtlose Verbindungen entwickelt
 - Annahme Festnetz: Paketverlust = Stau
 - Auf drahtlosen Links gibt es jedoch viele andere Gründe für Paketverluste (Interferenzen, Handover, ...)

- Lösungsansätze
 - Fast Retransmit
 - Indirektes TCP
 - Snooping TCP
 - Quickstart

- 12.1 Worin liegen die Probleme beim Einsatz von TCP über drahtlose Verbindungen?
- 12.2 Welche der betrachteten Verfahren eignen sich besser bei einem Handover, welche bei Paketverlusten durch Interferenzen?
- 12.3 Beschreiben Sie den Einsatz von Fast Retransmit!
- 12.4 Warum kann sollte nach einem Fast Retransmit ein Slow Start erfolgen?
- 12.5 Wie funktioniert indirektes TCP?
- 12.6 Welche Nachteile von indirektem TCP werden durch Snooping TCP gelöst?
- 12.7 Erläutern Sie an einem Beispiel, wie nicht Quickstart-fähige Router erkannt werden!

Referenzen, weiterführende Literatur

- [12.1] Hala Elaarag; Improving TCP Performance over Mobile Networks, ACM Computing Surveys, September 2002
- [12.2] H. Balakrishnan, S. Seshan, R. H. Katz; Improving reliable transport and handoff performance in cellular wireless networks, Wireless Networks, J.C. Baltzer, Band 1, 1995
- [12.3] E. A. Brewer et al.; A Network Architecture for Heterogeneous Mobile Computing, IEEE Personal Communications, 5(5), 1998
- [12.4] A. Fieger, M. Zitterbart; Zuverlässige Transportdienste für Mobile Computing, Informatik – Forschung und Entwicklung, 16(4), 2001
- [12.5] IETF Transport Area Working Group, <http://www.ietf.org/html.charters/tsvwg-charter.html>
- [12.6] S. Schütz et al.: Protocol Enhancements for Intermittently Connected Hosts, ACM Computer Communication Review, Juli 2005

Referenzen, weiterführende Literatur

- [12.13] H. Balakrishnan, S. Seshan, R. H. Katz; Improving reliable transport and handoff performance in cellular wireless networks, Wireless Networks, J.C. Baltzer, Band 1, 1995
- [12.14] E. A. Brewer et al.; A Network Architecture for Heterogeneous Mobile Computing, IEEE Personal Communications, 5(5), 1998
- [12.15] J. Roth; Mobile Computing: Grundlagen, Technik, Konzepte, dpunkt, 2002
- [12.16] A. Fieger, M. Zitterbart; Zuverlässige Transportdienste für Mobile Computing, Informatik – Forschung und Entwicklung, 16(4), 2001
- [12.17] IETF MIP4 Working Group, <http://www.ietf.org/html.charters/mip4-charter.html>
- [12.18] H. Soliman et al.: Hierarchical Mobile IPv6 Mobility Management, RFC 4140, August 2005
- [12.19] Rajeev Koodli: Fast Handovers for Mobile IPv6, RFC 4068, Juli 2005
- [12.20] IETF Transport Area Working Group, <http://www.ietf.org/html.charters/tsvwg-charter.html>
- [12.21] RFC 2608 - Service Location Protocol, Version 2
- [12.22] RFC 2610 - DHCP Options for Service Location Protocol
- [12.23] S. Schütz et al.: Protocol Enhancements for Intermittently Connected Hosts, ACM Computer Communication Review, Juli 2005

Vielen Dank!