

Improving TCP Performance over Wireless Links



Holger Füßler

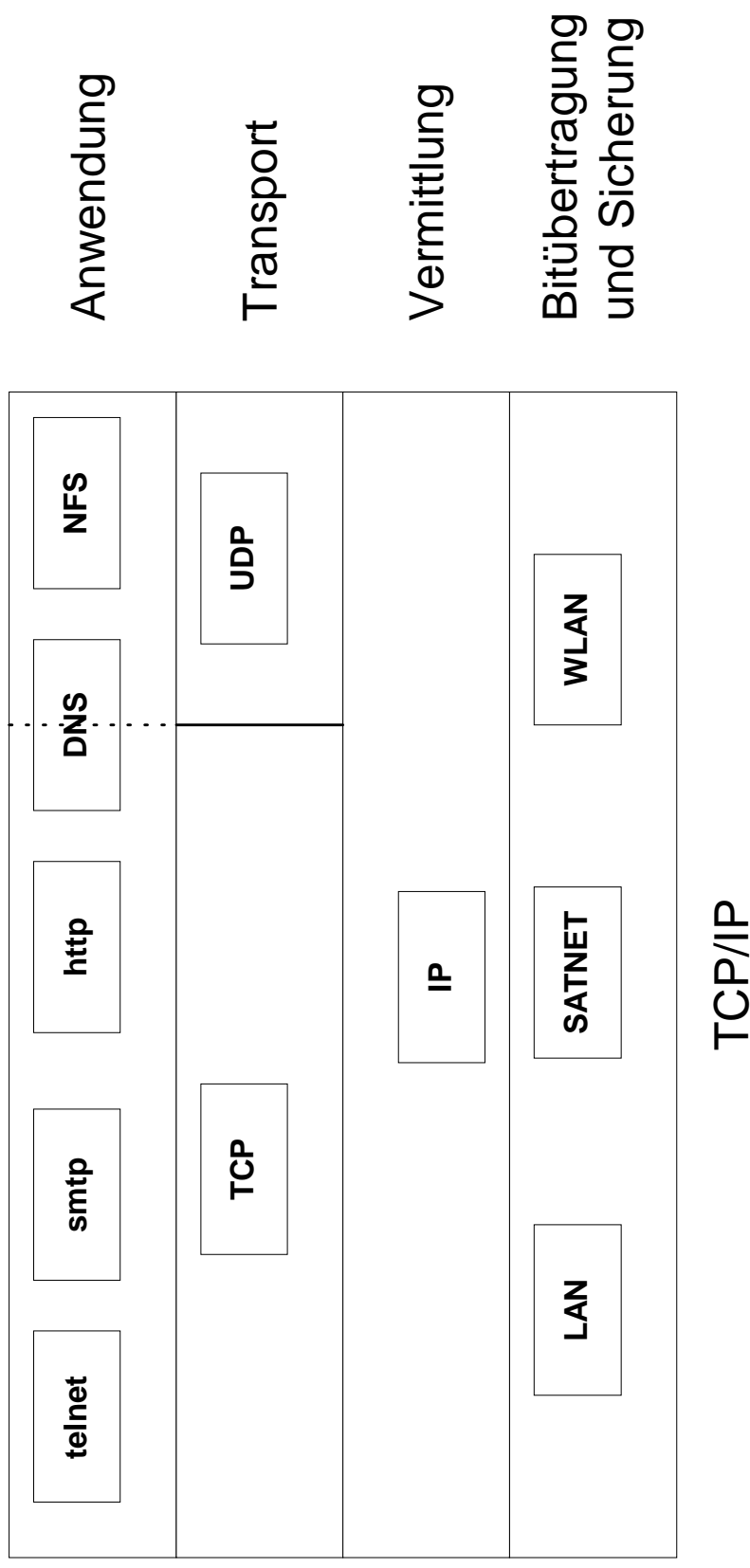
Gliederung

- Status Quo und Motivation
 - momentan verbreitetes TCP
 - besondere Situation in Wireless LANs
 - Motivation für „neue“ Protokolle
- Vorstellung verbesserte Konzepte
 - End-2-End
 - Link-Layer
 - Split-Connection
- Versuch und Versuchsauswertung
 - Versuchsaufbau
 - Performancemessung und -deutung
- Folgerungen

Gliederung 1

- Status Quo und Motivation
 - momentan verbreitetes TCP
 - besondere Situation in Wireless LANs
 - Motivation für „neue“ Protokolle
- Vorstellung verbesserte Konzepte
 - End-2-End
 - Link-Layer
 - Split-Connection
- Versuch und Versuchsauswertung
 - Versuchsaufbau
 - Performancemessung und -deutung
- Folgerungen

TCP im Internet-Protokollstapel



Merkmale von TCP

- Voll duplex-Verbindung (Bytestrom)
- Reihenfolgegarantie (Neuordnung)
- Entfernung doppelter Segmente
- Fehlererkennung und -korrektur

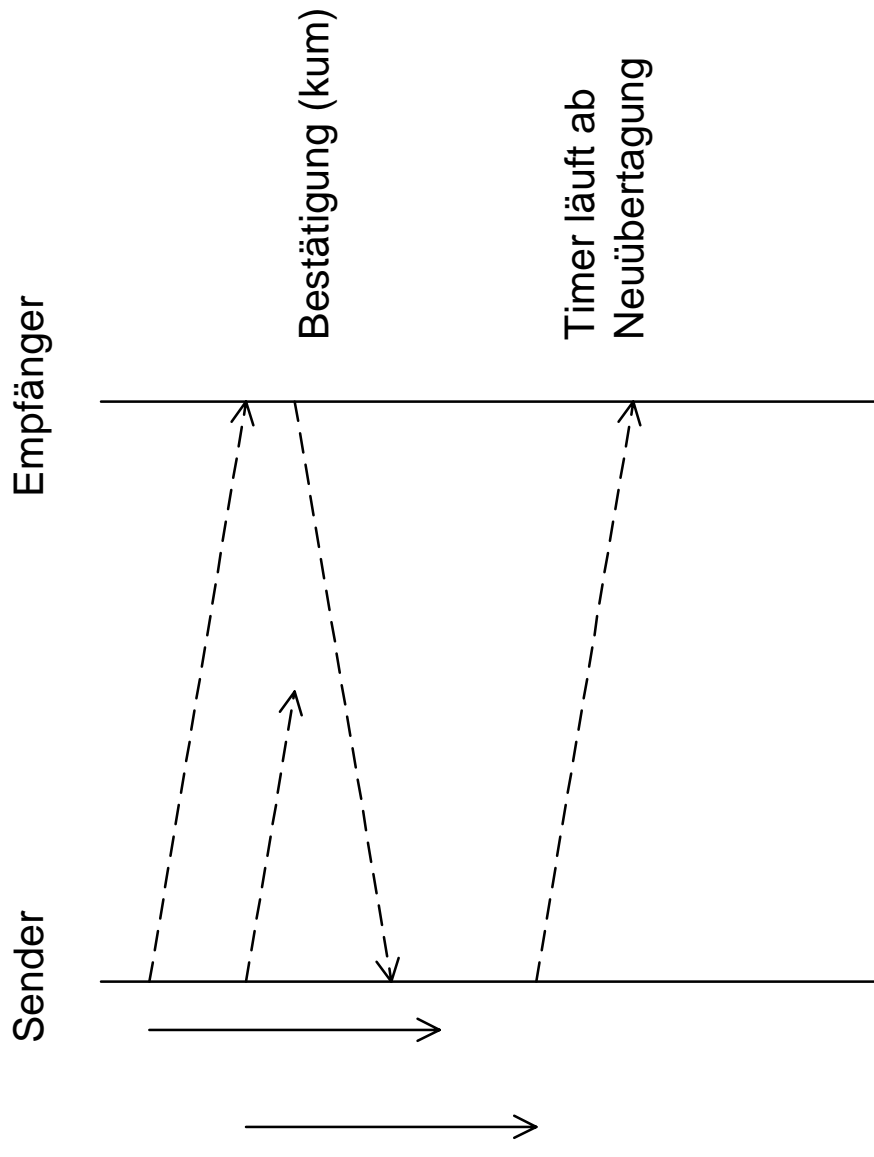
Annahmen beim TCP-Entwurf

- Netzwerke mit geringer Fehlerrate
- Paketverluste hauptsächlich durch Überlast
d.h. Router werfen Pakete, wenn ihr Lastzustand zu gross ist

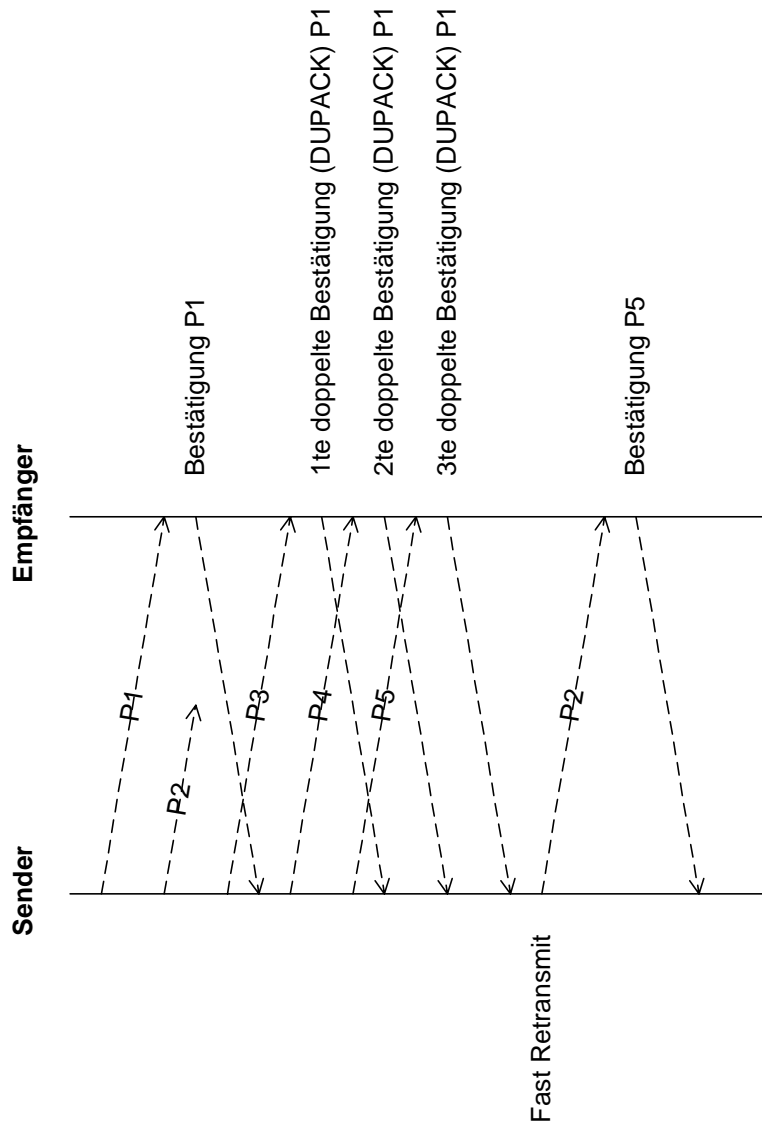
Deshalb:

- Bei Fehlererkennung werden Überlastkontroll und –
vermeidungsmechanismen aktiviert

Fehlererkennung bei TCP



Variante der Fehlerbehebung



nur 1 Paket wird neu gesendet

Fenster- und Timerverwaltung

- **Receiver Window (rwnd)**
- **Congestion Window (cwnd)**

Maximales Sendefenster entspricht dem Minimum der Beiden

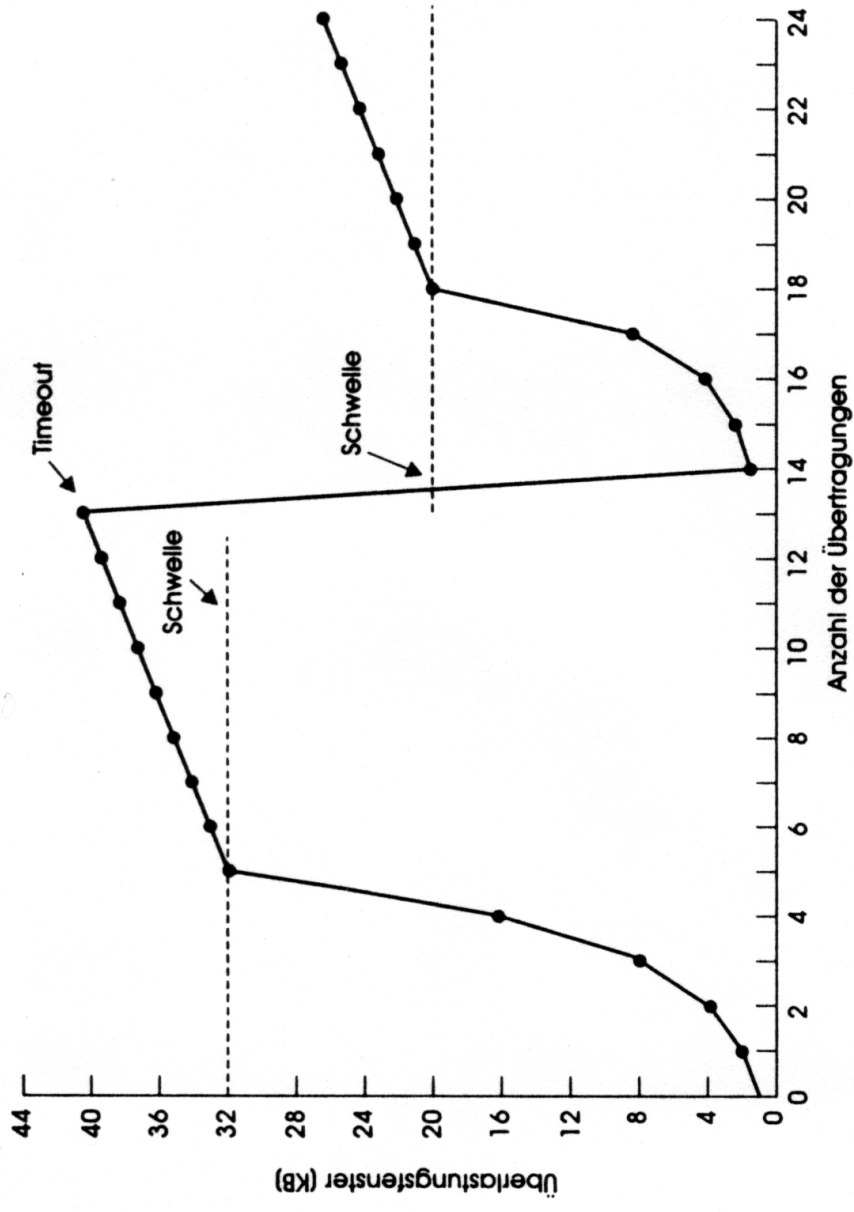
- **Retransmission Timer (rtst)**

Berechnet sich aus fortgeschriebener (geglätteter) mittlerer round-trip-time und Ihrer Standardabweichung

Lastkontrollmechanismen

- **Slow Start**
Bandbreite wird an Lastgrenze herangeführt
- **Congestion Avoidance**
Bei Fehler wird das Lastfenster (drastisch) verkleinert
UND der rttst vergrößert
- **Fast Retransmit**
Beim dritten DUPACK wird das fehlende Paket nochmal gesendet
- **Fast Recovery**
Nach Fast Retransmit wird nicht wieder bei 1 angefangen

Typische Größe des Lastfensters



Situation in drahtlosen Netzen

- **Hohe mittlere Fehlerrate**
Unnötiger Aufruf von Lastkontrollmechanismen
- **Grosse Anzahl von Burst-Fehlern**
Möglicherweise mehrere Pakete innerhalb eines Sendefensters betroffen, deshalb funktioniert Fast Retransmission nicht
- **Periodisch auftretende Verbindungsverzögerungen durch Hand-Offs**

Dies führt zu einer insgesamt zu großen Berechnung des $rtst$, der dann die Fehlererkennung verzögert

Folgen für den TCP Einsatz

- Dramatische Reduktion des Durchsatzes
(bis zu 50%)
- Hohe Verzögerungen bei interaktiven Anwendungen

Deshalb Anpassung der Protokolle an die neue Situation

Gliederung 2

- Status Quo und Motivation
 - momentan verbreitetes TCP
 - besondere Situation in Wireless LANs
 - Motivation für „neue“ Protokolle
- Vorstellung verbesserte Konzepte
 - End-2-End
 - Link-Layer
 - Split-Connection
- Versuch und Versuchsauswertung
 - Versuchsaufbau
 - Performancemessung und -deutung
- Folgerungen

Lösungsansätze

Grundsätzlich lassen sich zwei Ansätze identifizieren:

- Das Verstecken aller nicht-überlastbedingten Fehler vor den höheren Schichten
 - Link erscheint als Verbindung höherer Qualität, aber mit niedrigerer Bandbreite
 - Lokale Lösung eines lokalen Problems
- Der Sender weiss, dass es auch nicht-überlastbedingte Fehler gibt und vermeidet bei solchen Fehlern das Aufrufen von Lastkontrollmechanismen

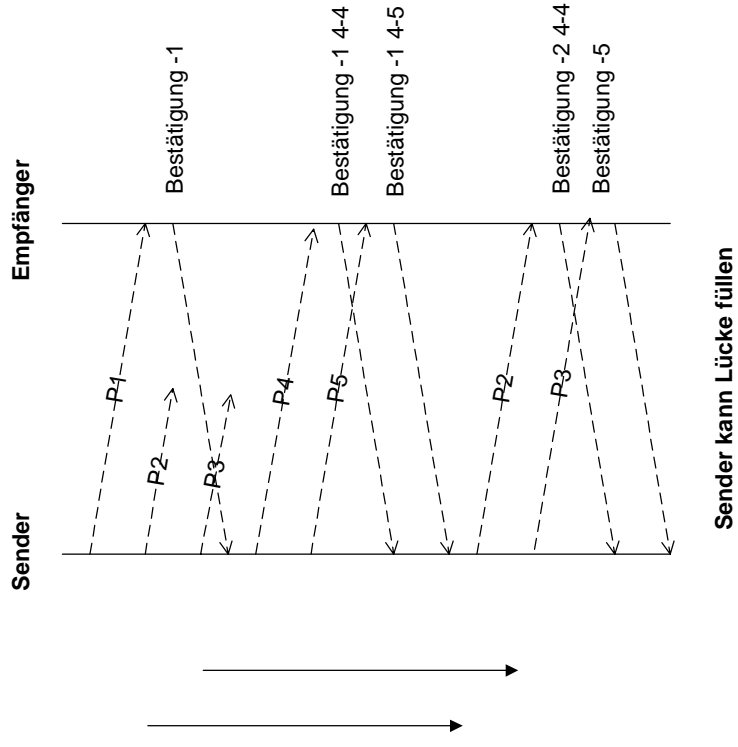
Einteilung in drei Gruppen

- **End-2-End-Ansätze**
 - SACK (selective acknowledgment)
 - ELN (explicit loss notification)
- **Split-Connection-Ansätze**
 - Auftrennen der Verbindung in zwei an der Basisstation
 - Cachen der Pakete
 - Beide Verbindungen homogen bzgl. Parameter
- **Link-Layer-Ansätze**
 - Local Retransmission
 - Evtl. Forward Error Correction

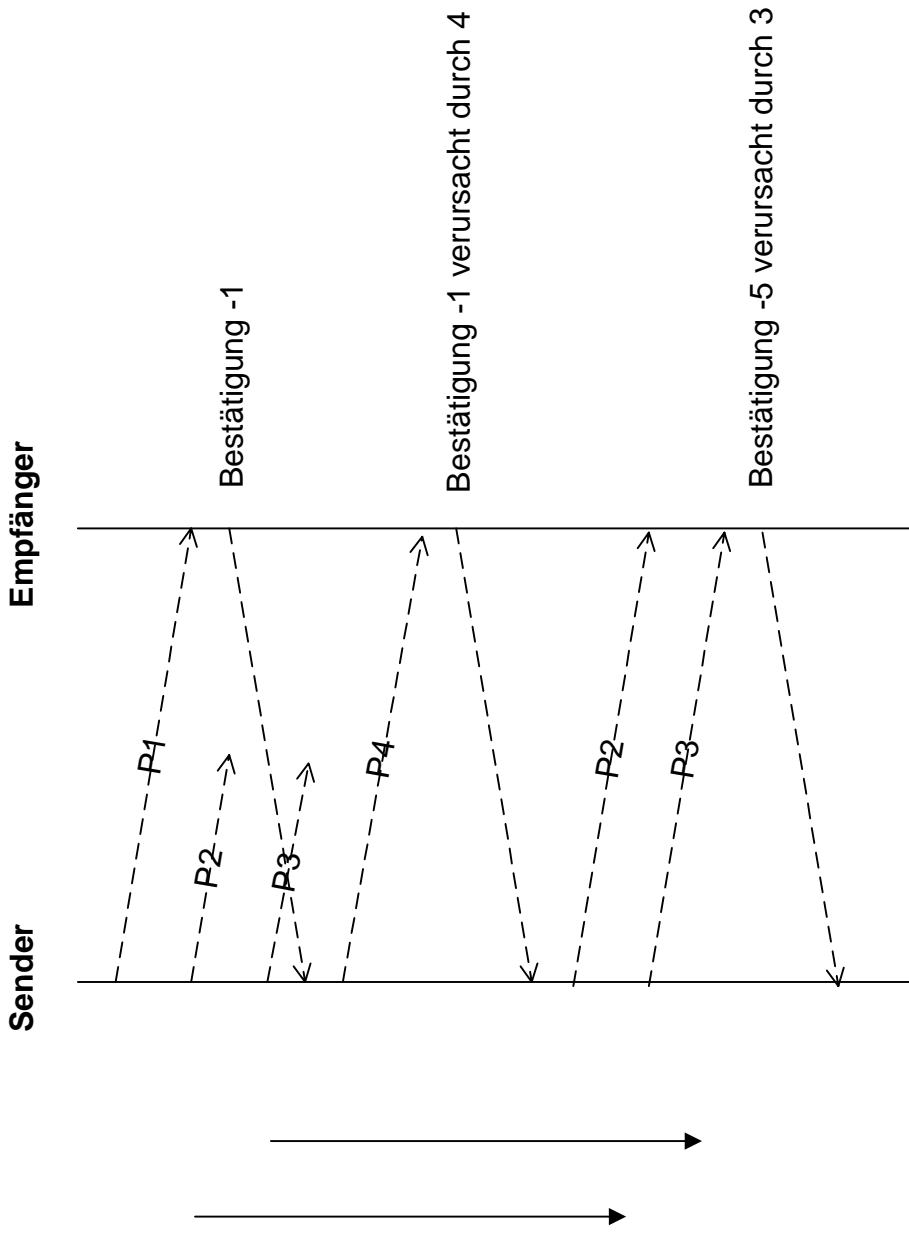
Selective Acknowledgements

- Im Gegensatz zu cumulative acknowledgements können hier auch nichtzusammenhängende Datenbereiche bestätigt werden
- Der Sender muss beim retransmit nur die fehlenden Pakete übertragen
- Es gibt 2 relevante Vorschläge:
 - SACK-Vorschlag der IETF
 - 3 nicht-zusammenhängende Blöcke können spezifiziert werden
 - das Senderverhalten wird nicht spezifiziert
 - SMART
 - das ACK enthält das Paket, dass für seine Generierung verantwortlich war
 - der Sender kann daraus eine Bitmaske über erhaltene Pakete generieren
 - tritt eine Lücke auf, wird sofort ein retransmit der betr. Pakete eingeleitet

Beispiel SACKS



Beispiel SMART

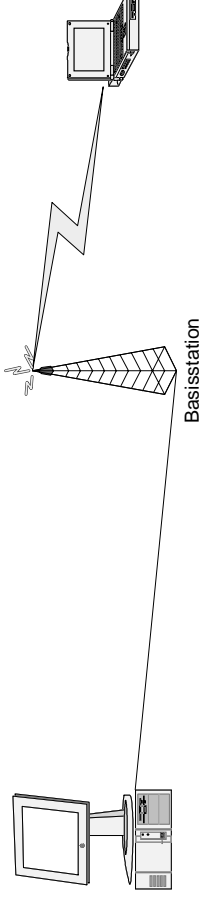


Sender kann Lücke füllen

Explicit Loss Notification

- hier merkt sich der Stack, wenn ein nicht-überlastbedingter Fehler aufgetreten ist und fügt zukünftigen DUPACKs ein ELN hinzu
- der Sender kann dann retransmittieren, ohne Lastkontrolle zu aktivieren

Split-Connection Ansatz



- auf drahtloser Seite wahlweise
 - normales TCP
 - TCP mit SACK / SMART
- keine Überlagerung von wireless-Effekten und internet-Effekten
- trotzdem beide vorhanden

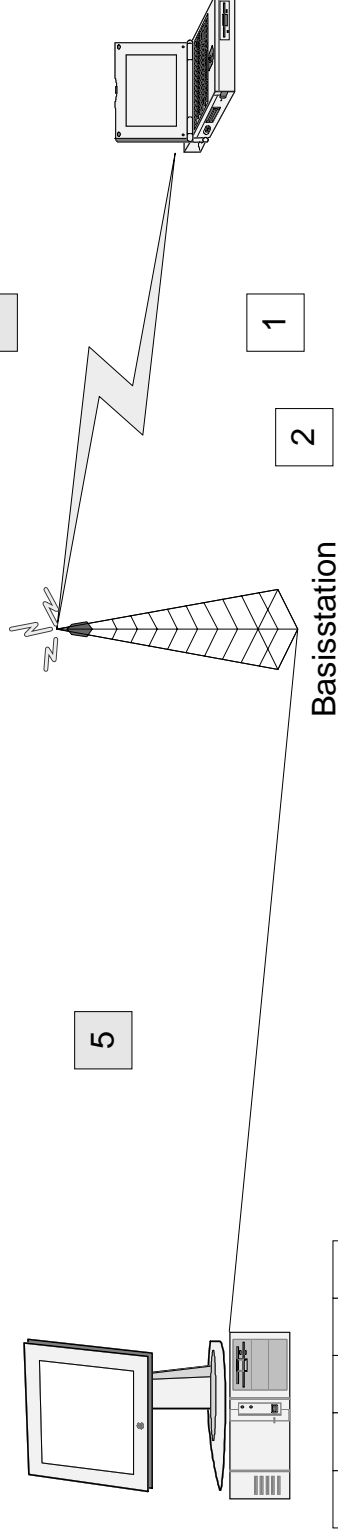
Link-Layer-Ansatz

- keine de facto-Standards
- Forward Error Correction
 - durch Redundanz in der Übertragung wird Fehlerverhalten verbessert
- Automatic Repeat Request (ARQ)
 - fehlerhafte Pakete werden automatisch erneut übertragen
- **selbständige Übertragungswiederholung und Unterdrückung der rückläufigen DUPACKs**

Beispiel Snoop Protocol

DUPACKs von 3, 4 und 5

Annahme: Paket 1 und 2 gehen verloren

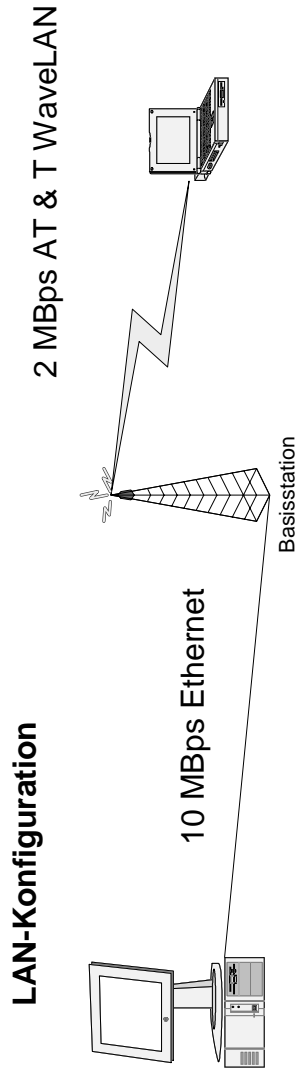


Retransmission von 1 und 2

Gliederung 3

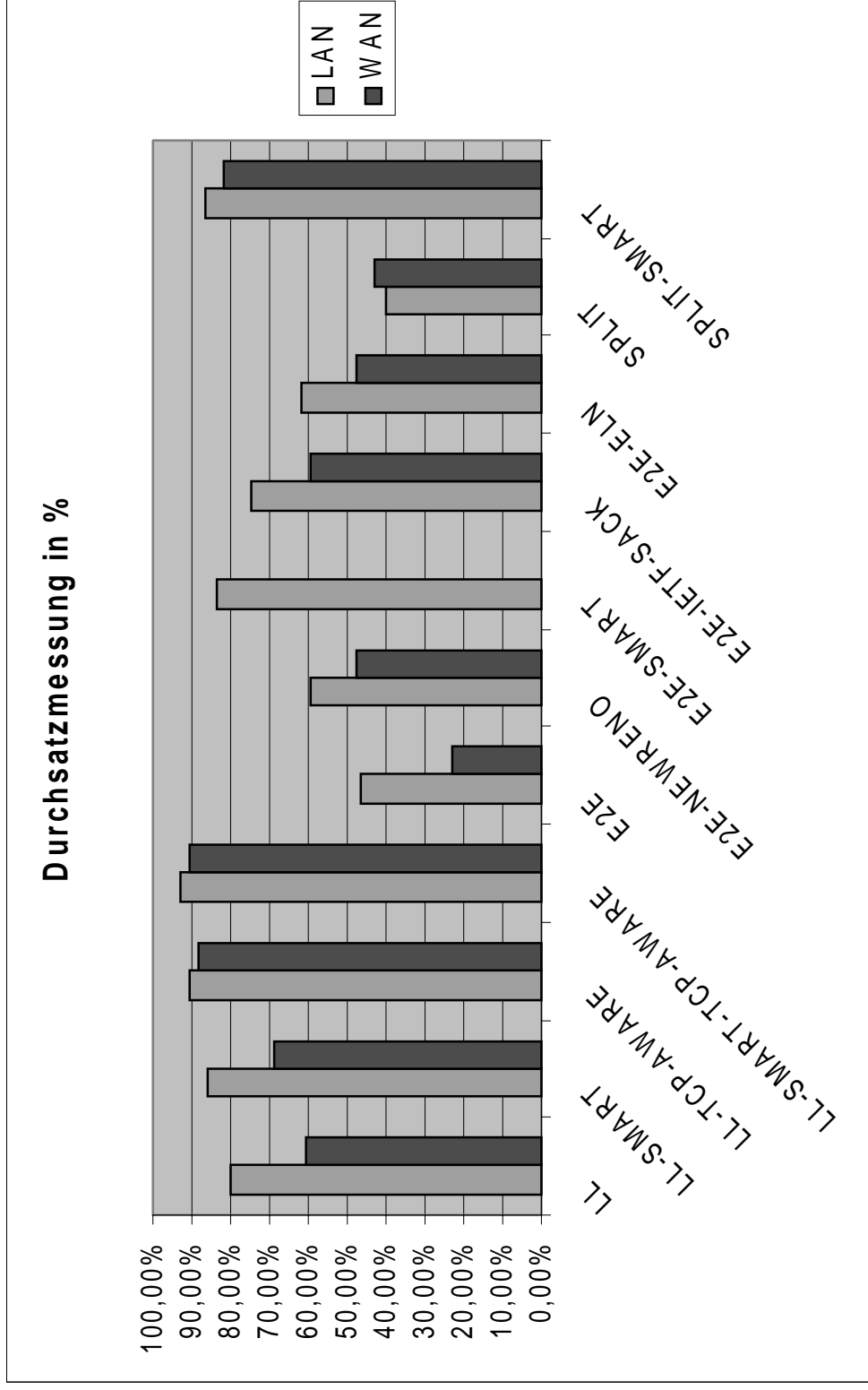
- Status Quo und Motivation
 - momentan verbreitetes TCP
 - besondere Situation in Wireless LANs
 - Motivation für „neue“ Protokolle
- Vorstellung verbesserte Konzepte
 - End-2-End
 - Link-Layer
 - Split-Connection
- Versuch und Versuchsauswertung
 - Versuchsaufbau
 - Performancemessung und -deutung
- Folgerungen

Versuchsanordnung



- Absenz von Überlast und „richtigen“ Übertragungsfehlern
- Fehlern werden exponentialverteilt erzeugt
- Filetransfer von 8MB zum Laptop
- WAN-Konfiguration:
 - statt Ethernet 16 Internet-Hops
 - nachts für minimale Überlast

Durchsatzmessung



Gliederung 4

- Status Quo und Motivation
 - momentan verbreitetes TCP
 - besondere Situation in Wireless LANs
 - Motivation für „neue“ Protokolle
- Vorstellung verbesserte Konzepte
 - End-2-End
 - Link-Layer
 - Split-Connection
- Versuch und Versuchsauswertung
 - Versuchsaufbau
 - Performancemessung und -deutung
- Folgerungen

ausgewählte Folgerungen

- durch ein möglichst intelligentes link-layer-Protokoll lässt sich die beste Performance erzielen
- bei den Ende-zu-Ende-Protokollen ist das von der IETF vorgeschlagene Protokoll das Beste (SMART im LAN-Bereich)
- ELNs stabilisieren die Größe des cgnw und erhöhen seinen Durchschnitt
- SPLIT ist E2E nur im WAN-Bereich überlegen
- SPLIT-SMART performt hervorragend

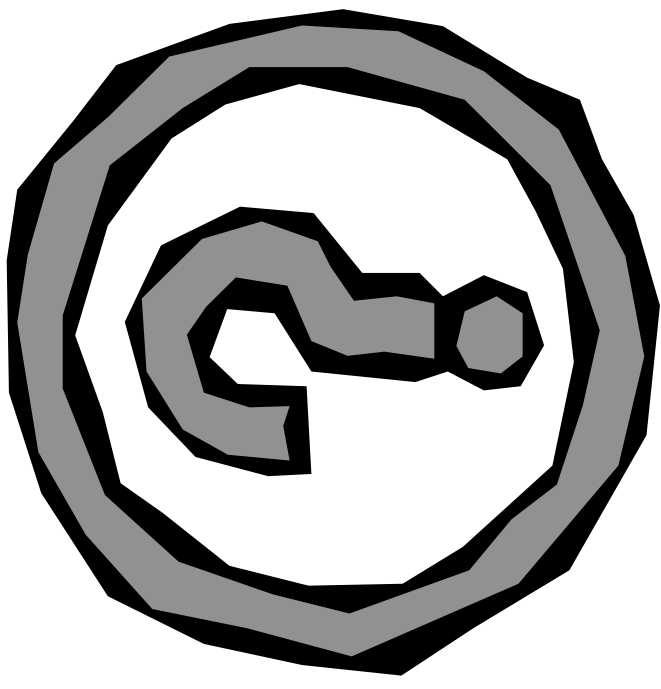
Fazit

- den neuen Transportmedien muss auch in den verwendeten Protokollen Rechnung getragen werden
- es gibt bereits entscheidende Verbesserungen, die noch Einzug in die Systeme halten müssen

Performance bei UDP

- sollte kein Problem sein, da UDP per Definition unzuverlässig
- in der Praxis verlässt man sich leider doch darauf
- Protokolle wie NFS, die eigenen reliability layer haben, müssen sich selbst etwas einfallen lassen

Fragen



Vielen Dank für Ihre
Aufmerksamkeit

