
Einfluss der Window Scale Option auf die Fairness in TCP/IP-Netzen

Torsten Müller,
TU-Dresden, Institut für Nachrichtentechnik,
Professur Telekommunikation Prof. Dr.-Ing. R. Lehnert



Übersicht

- Einleitung
- TCP Window Scale Option
- Untersuchungsszenario
- Ergebnisse FIFO + großer Puffer
- Ansätze zur Problemlösung
- Ergebnisse
 - ⇒ Pufferbegrenzung Drop-Tail
 - ⇒ WFQ
 - ⇒ RED
- Zusammenfassung



TCP Window Scale Option

- TCP: sliding window Flußkontrolle
- max. Fenster 64 kByte (begrenzt WND-Feld 16 bit)
- Problem bei hohem Bandbreite-Verzögerungs-Produkt:
 - ⇒ begrenzter Durchsatz
- größere Fenster durch Window Scale Option möglich
(RFC 1323: TCP Extensions for High Performance)

$$D_{\max} = \frac{WND_{\max}}{RTT} = \frac{64kByte}{RTT}$$

⇒ Beispiel: s=40000km, RTT=133ms, D=3,8 Mbit/s

⇒ bei geforderten 155 Mbit/s -> WND=2.5 MByte

(RFC 1323: TCP Extensions for High Performance)

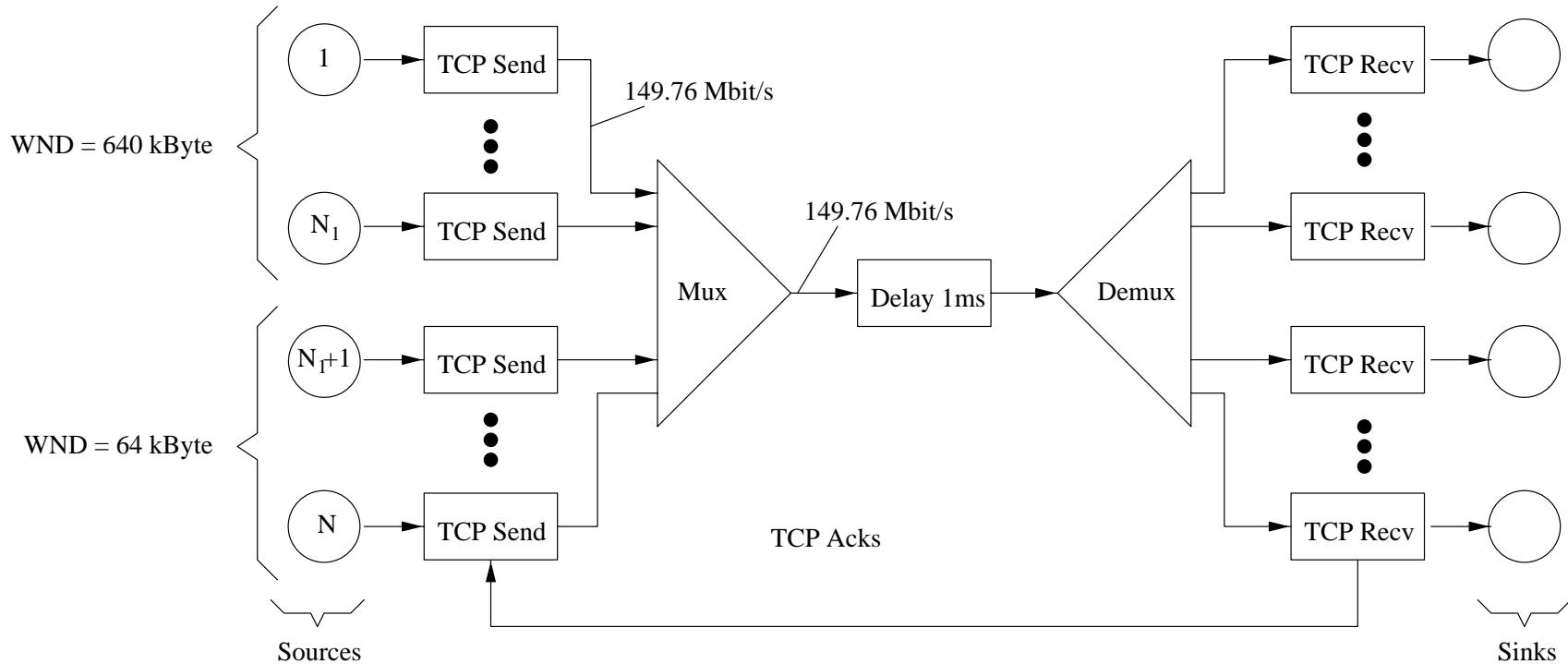


TCP Window Scale Option (2)

- Bedingungen für Benutzung:
 - ⇒ Unterstützung durch Betriebssystem
http://www.psc.edu/networking/perf_tune.html
 - ⇒ Sender und Empfänger müssen Option unterstützen
 - ⇒ Socketgröße (Einstellung, Speicher)
 - ⇒ durch alle Verbindungen nutzbar
(unabhängig von Bandbreite-Verzögerungsprodukt)
 - ⇒ Resultat: Mix von Verbindungen mit/ohne WSO
- Puffer in Routern heute groß, Speicher billig
- Frage: Fairness?



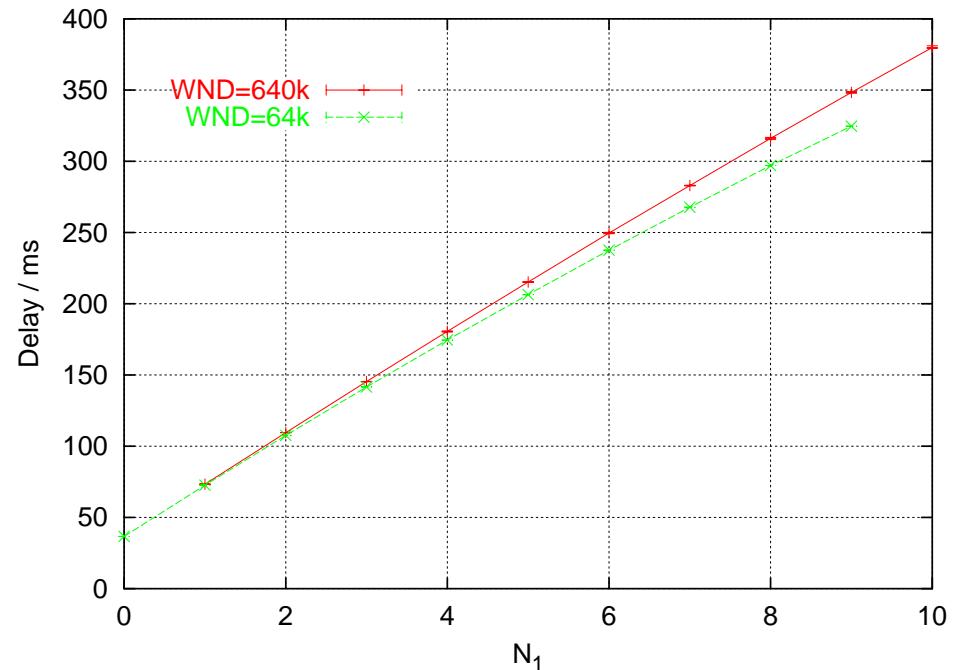
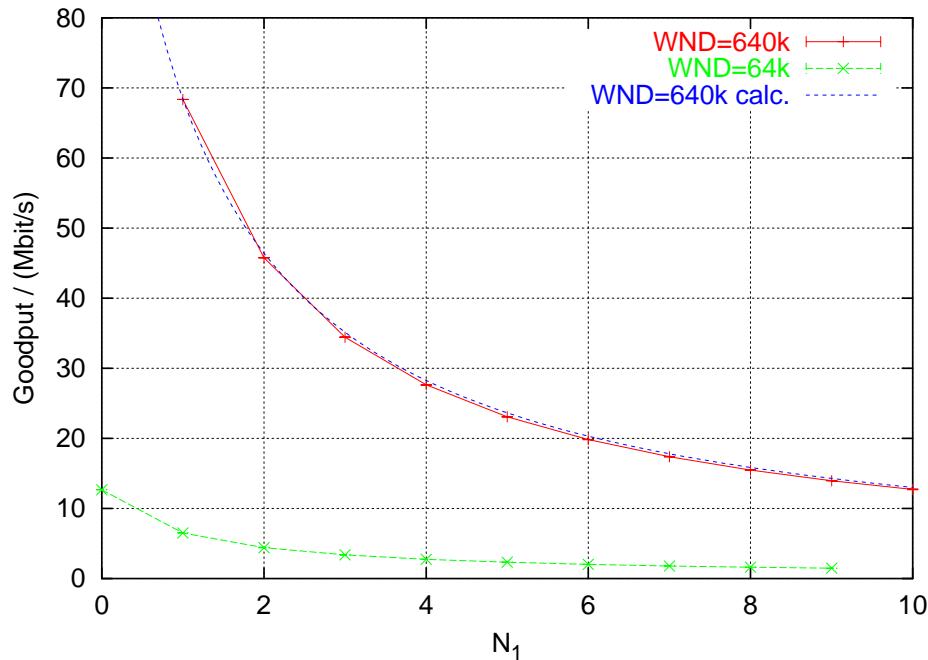
Untersuchungsszenario



- $N=10$ TCP Verbindungen
- N_1 variabel
- FIFO, Puffer sehr gross

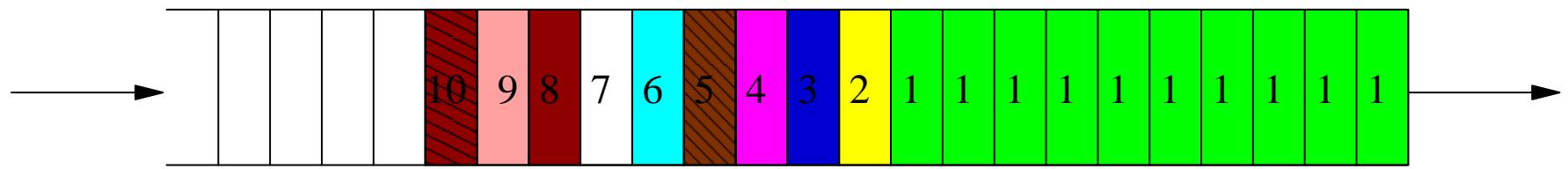


LAN, FIFO



- Bandbreite wird proportional zur max. Fenstergröße vergeben
- Aufteilung unfair: "normale" Verb. bekommen nur wenig Bandbreite
- erzeugt lange Bursts: große Verzögerung in Knoten

Erklärung Unfairness



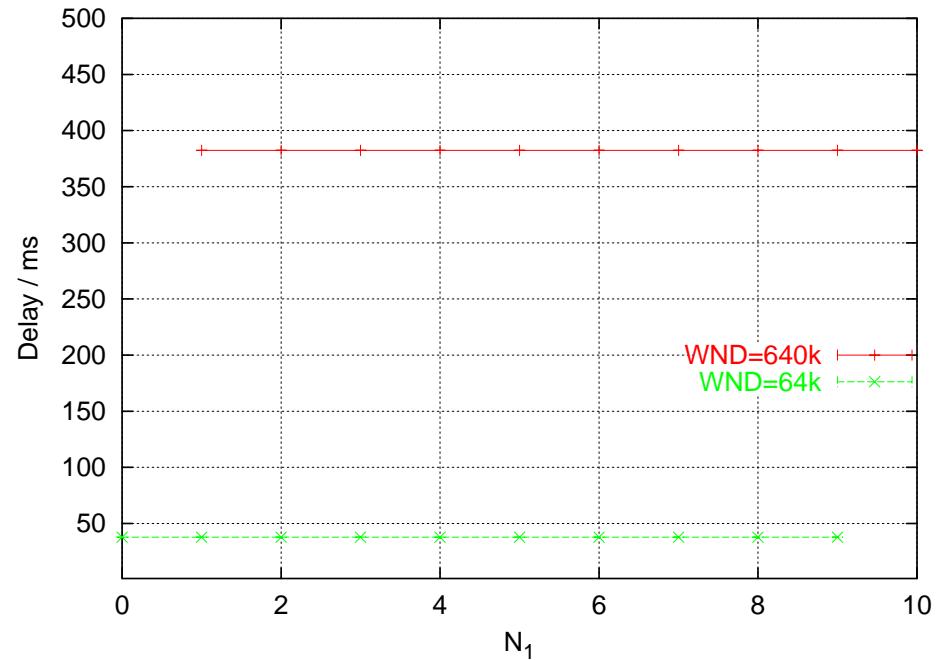
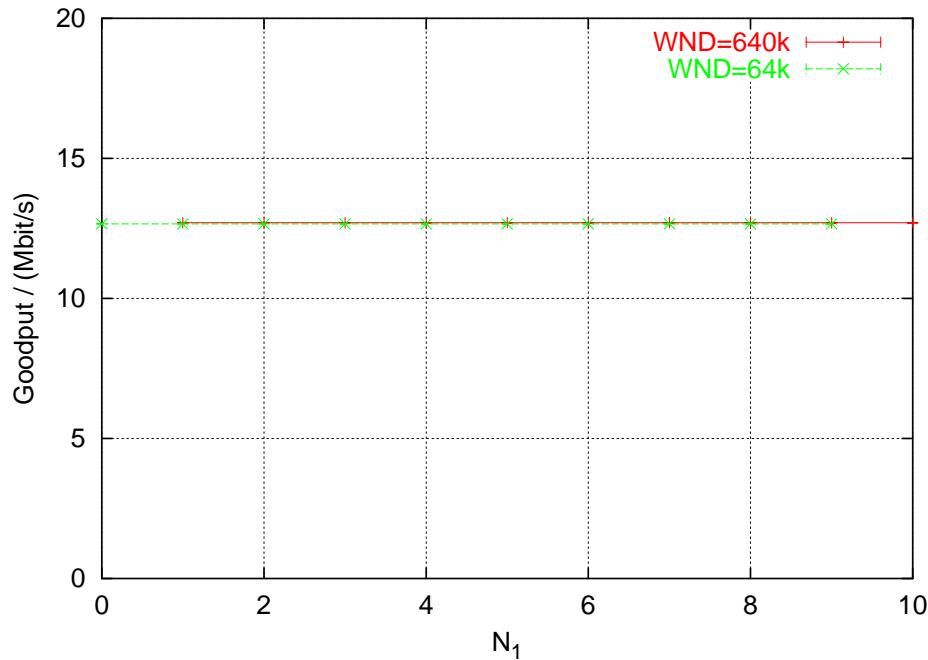
- FIFO:
 - ⇒ Durchsatzanteil eines Flows abhängig von Anteil Daten in Queue
 - ⇒ mittlere Wartezeit steigt mit Datenmenge in Queue
- großer Puffer
 - ⇒ keine Verluste
 - ⇒ TCP congestion control reagiert nicht

Ansätze zur Problemlösung

- Administration max. TCP Fenstergröße
 - ⇒ entsprechend RTT und Bandbreite
- Fair Queuing
 - ⇒ per flow queuing notwendig
- Begrenzung Puffer in Switch/Router
- Einsatz Puffer Management (z.B. RED)



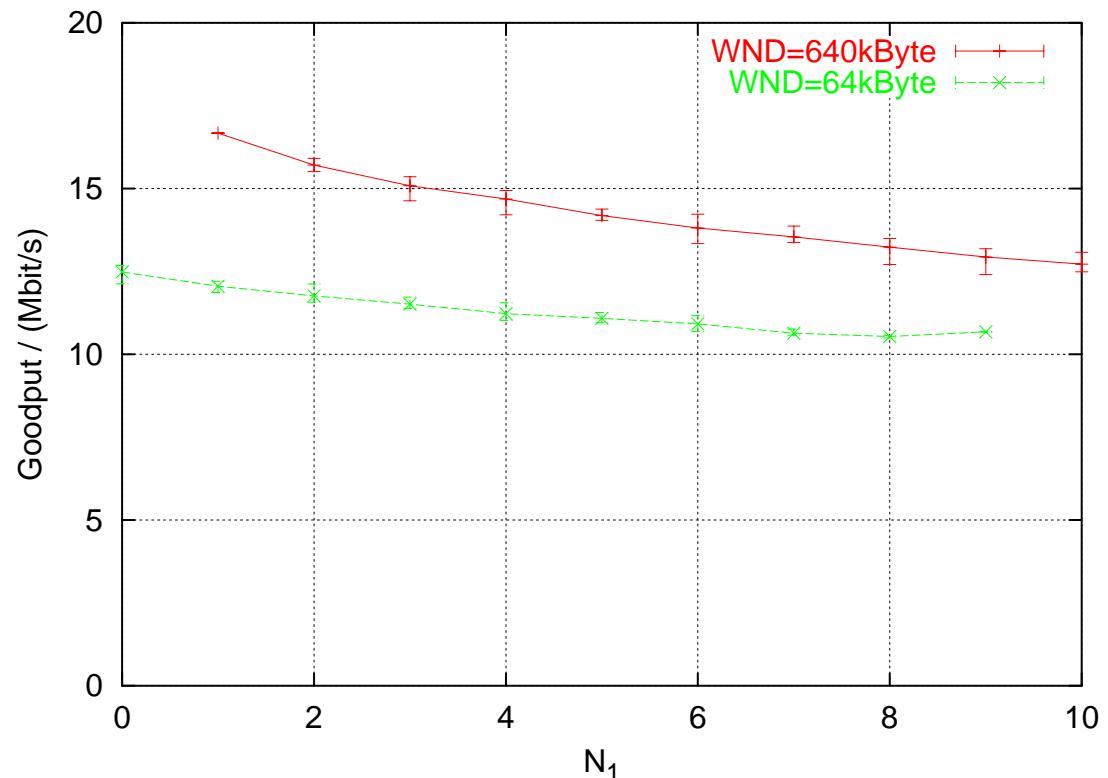
Fair Queuing (WFQ)



- keine Unfairness
- Isolierung der Verbindungen
- Verfahren aufwendig

begrenzter Puffer, Tail Drop

- Puffergröße 5000 Zellen ($\approx 230\text{kByte}$)
- geringe Unfairness
- nicht dargestellt:
 - ⇒ FIFO und WFQ gleich
 - ⇒ Verzögerungsproblem bei FIFO entschärft

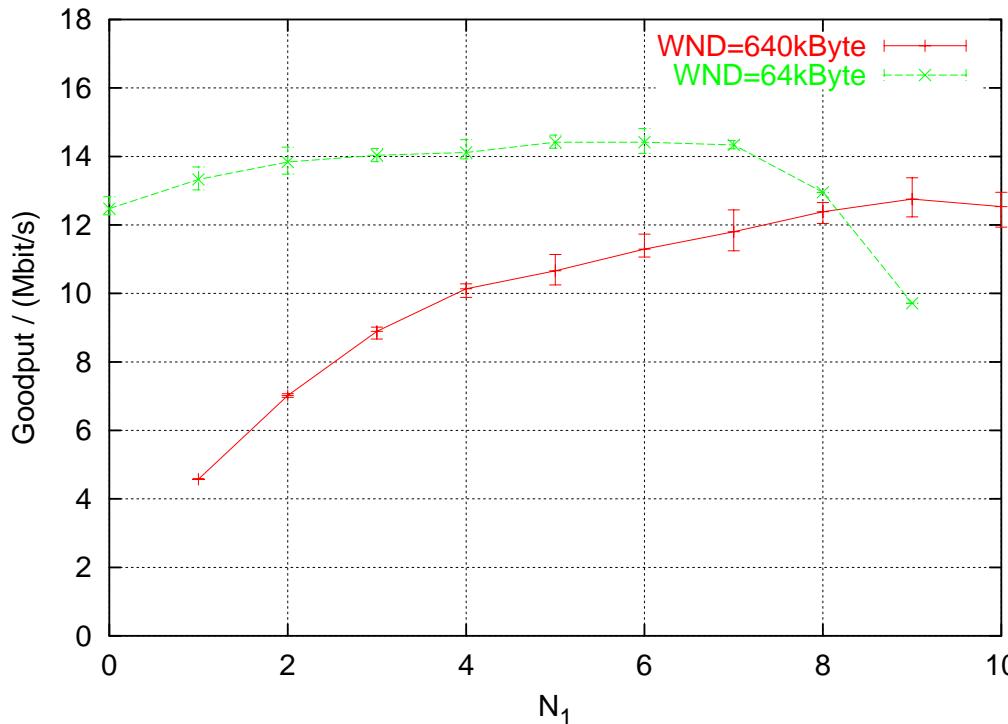


WAN-Szenario

- WSO für Weitverkehrsverbindungen gedacht
- WSO-Verbindungen
 - ⇒ Leitungsverzögerung 20 ms
 - ⇒ max. Fenster von 640 kByte
 - ⇒ ermöglicht Durchsatz >260 MBit/s



WAN, begrenzter Puffer



- Nachteil für Verbindungen mit großer Verzögerung
 - ⇒ große RTT, Erhöhung CWND langsam
 - ⇒ nach Timeout (CWND =1 MSS) problematisch

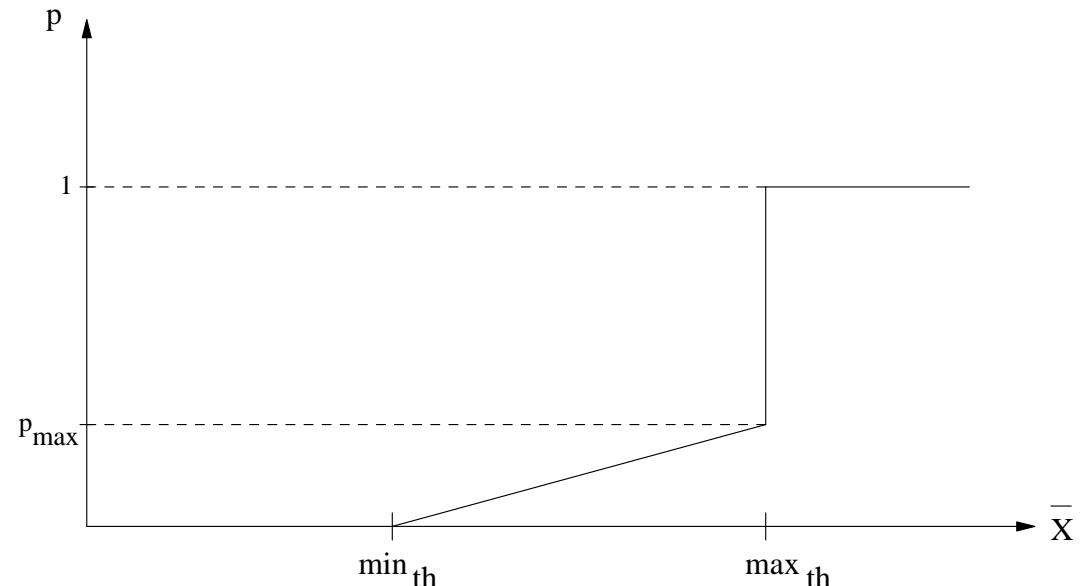
Random Early Detection (RED)

● Arbeitsweise

- ⇒ Markieren von Paketen bevor Puffer voll
- ⇒ $p = f$ (mittlere Queue-länge)
- ⇒ kein ECN -> Verwerfen

● Parameterwahl:

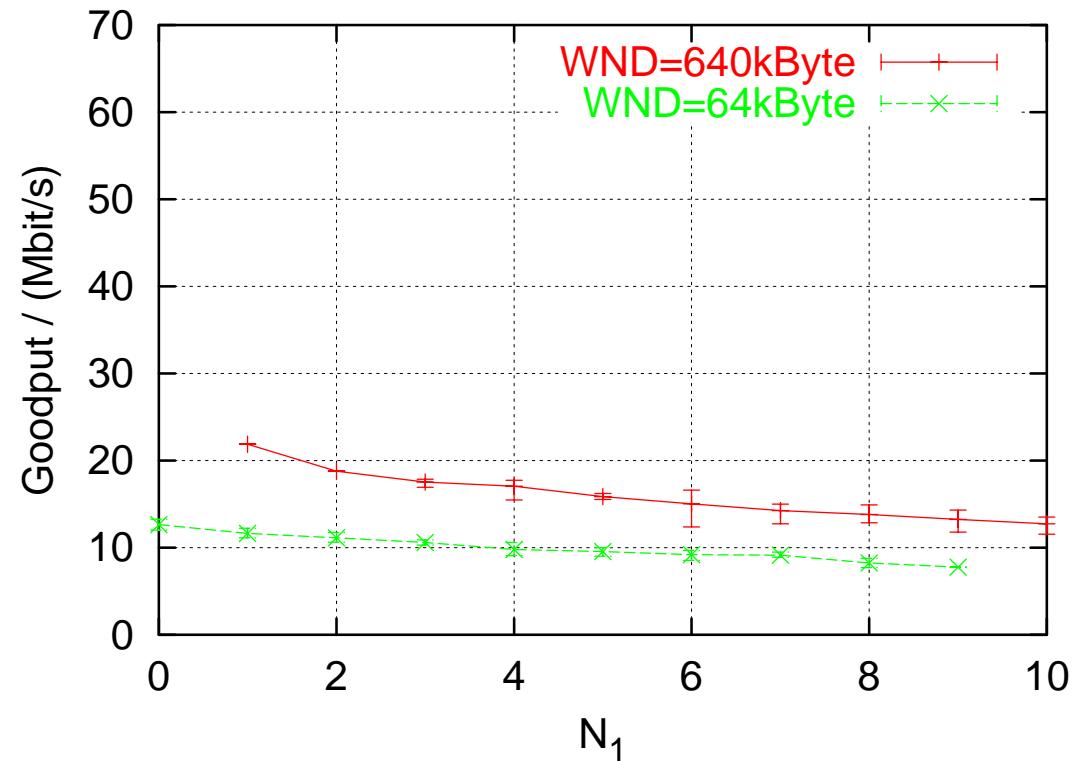
- ⇒ min_th: wenn zu groß, keine Wirkung
- ⇒ min_th=0,
max_th=50000 Zellen,
 $p_{\text{max}}=0.1$,
 $w_q=0.02$



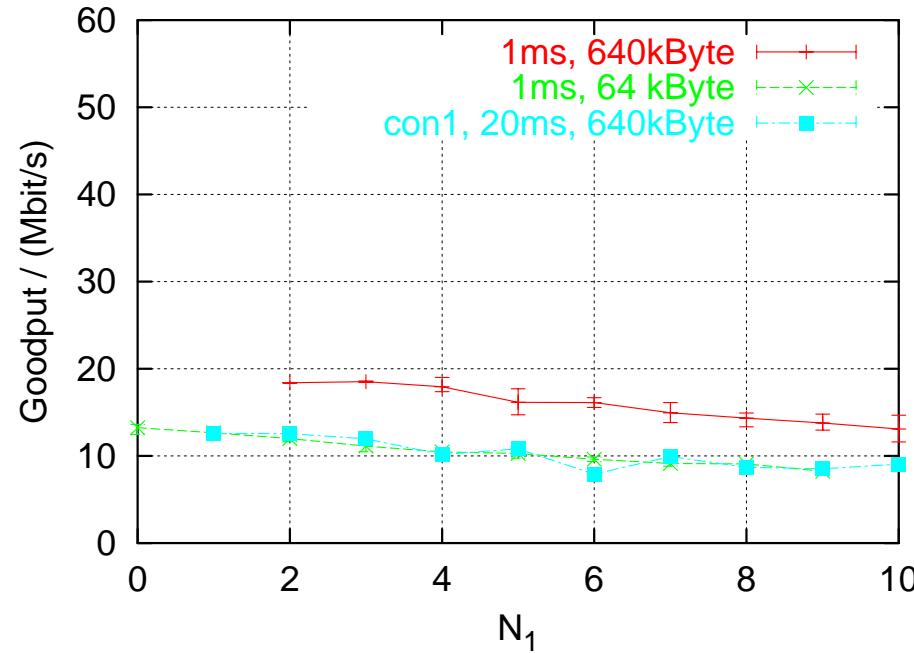
Ergebnisse RED, LAN

- Fairness

- ⇒ Verbesserung gegenüber FIFO + großer Puffer
- ⇒ keine Verbesserung gegenüber Drop-Tail



Ergebnisse RED, 1 WAN-Verb.



- RTT (Verbindung 1) = 20ms
- geringe Unfairness
- Verbindungen mit großer RTT etwas benachteiligt

Zusammenfassung

- FIFO+großer Puffer
 - ⇒ geringe Fairness
- Fair Queuing:
 - ⇒ fair
 - ⇒ kompliziert
- Pufferbegrenzung (Drop-Tail):
 - ⇒ ausreichende Fairness
 - ⇒ problematisch bei großen RTTs
- RED:
 - ⇒ ausreichende Fairness
 - ⇒ unterschiedliche RTT's
 - ⇒ entspr. Parametrisierung notwendig

