

Rechnernetze: Praktische Aufgabe 2

Kevin Seidel
943147

2013-06-22

Im folgenden Bericht werden die Ergebnisse von zwei verschiedenen Netzwerksimulationen dargelegt. Diese wurden nach den Vorgaben der **Praktischen Aufgabe 2** ([PA2]) der Veranstaltung Rechnernetze durchgeführt. In den Vorgaben wurden die zu verwendenden Durchsatzraten und die Netzwerktopologie festgelegt. Es sollten zwei Szenarien betrachtet werden. Im ersten Szenario wird ein voll funktionsfähiges Netzwerk simuliert, wohingegen im zweiten Szenario ein Netzwerk mit fehlerhafter Kabelverbindung betrachtet wird. Das weitere Vorgehen dieses Berichtes orientiert sich am **Measurement Cookbook** ([RNE3]), welches in der Veranstaltung Rechnernetze vorgestellt wurde.

1 Aufgabenstellung

Die Aufgabe bestand darin, zwei Netzwerksimulationen mit dem Simulationsprogramm **ns-3** ([NS3]) durchzuführen. Bei diesen Simulationen ging es um die Beobachtung der Datenraten bei einer TCP-Verbindung zwischen einem Server und 5 Clients, welche über einen Router miteinander verbunden sind. Die erste Simulation betrachtet ein funktionierendes Netzwerk ohne Fehler, wohingegen in der zweiten Simulation eine theoretische Beschädigung der Verbindung zwischen Router und Server simuliert wird. Diese Beschädigung bewirkt einen Paketverlust von 10% auf dieser Verbindung. Dabei soll ein Vergleich zwischen den Datenraten der beiden Simulationen erstellt werden und die Ergebnisse analysiert und interpretiert werden.

2 Netzwerkmodell

Wie in *Abbildung 1* zu sehen, besteht das zu simulierende Netzwerk aus fünf Clients, einem Router und einem Server. Der Server hat eine Verbindung zum Router. Diese erlaubt einen maximalen Durchsatz von **1Mbit/s** und hat eine Verzögerung von **20ms**. Die fünf Clients verfügen ebenfalls über jeweils eine Verbindung zum Router, welche auch über einen Durchsatz von **1Mbit/s** und eine Verzögerung von **20ms** verfügen. Alle

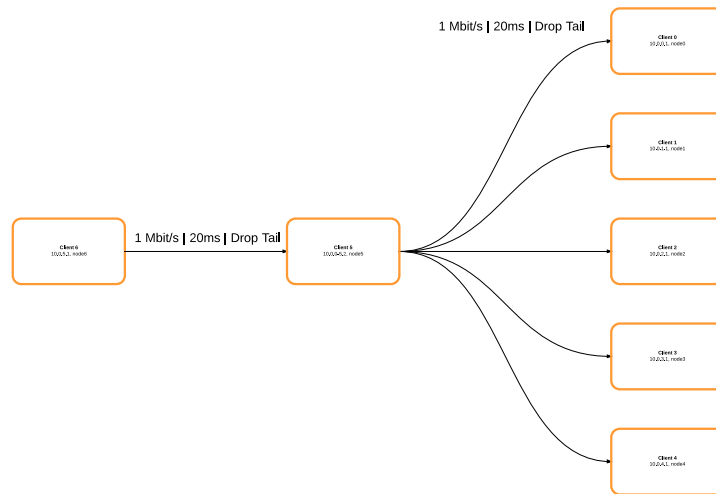


Abbildung 1: Netzwerktopologie

Netzwerkgeräte verfügen über eine **Drop Tail Queue**, das heißt es werden erst Pakete verworfen, sobald die jeweilige Queue komplett gefüllt ist. In der Simulation wird ein TCP-Stream vom Server zu jedem Client aufgebaut, welcher über den Router führt. Dabei sendet der Server kontinuierlich Daten an jeden Client. Jeder Client versucht dabei einen Durchsatz von 1Mbit/s zu erreichen, wobei zwischen Router und Server ein maximaler Durchsatz von 1Mbit/s realisiert werden kann, welcher sich auf die Clients aufteilen muss.

3 Leistungsmetrik

Für unsere Aufgabe betrachten wir sowohl den Durchsatz der einzelnen Clients, als auch den akkumulierten Durchsatz. Des Weiteren wird der aufsummierte Durchsatz betrachtet, um die Ergebnisse besser zu vergleichen.

4 Variable Parameter

Der einzige variable Parameter in unserem Aufbau ist die Packetverlustrate der Server-Router-Verbindung zwischen den beiden Simulationen. In der ersten Simulation beträgt diese 0%, heißt also, dass alle Pakete ordnungsgemäß übertragen werden. In der zweiten Simulation gehen 10% der Pakete zwischen Router und Server verloren.

5 Modellüberführung in Software

Für die Simulation wird die Software **ns-3** verwendet. Dabei handelt es sich um eine Simulationssoftware für Netzwerke. Zu Beginn der Simulation sollen noch keine Daten gesendet werden. Erst nach 5 Sekunden startet der Server damit, Pakete mittels TCP an den Client0 zu senden. Daraufgehend starten die anderen TCP-Streams in 5 Sekunden Abständen. Das bedeutet, dass nach 25 Sekunden die Streams zu allen 5 Clients aktiv sind. Diese werden für 40 weitere Sekunden aufrecht erhalten, bis alle Streams bei Sekunde 65 beendet werden. Nach insgesamt 70 Sekunden endet die Simulation. Diese Simulation ist in den Programmen **ReNe_SoSe_PA2[a-b]_943147.cc** implementiert.

Die Größe der TCP-Pakete die vom Server zu den Clients gesendet werden, beträgt 578 Byte. Die Acknowledgements (ACK) und Synchronize-Acknowledgement (SYN-ACK) haben eine Größe von 42 Byte. Dabei wird standardmäßig TCP NewReno ([TCPNR]) genutzt.

6 Konfiguration zur Datenerzeugung

Das Simulationsprogramm erzeugt für die spätere Analyse ein **ASCII-Trace** mit allen gesendeten, eingeeuerten und ausgeueuerten Paketen. Zusätzlich wird eine **XML-Datei** erzeugt, welche es mittels des Programmes **NetAnim** erlaubt, die gesendeten Pakete in einer grafischen Animation darzustellen. Des Weiteren werden **Package Capture** (PCAP) Dateien erstellt, welche die erhaltenen und gesendeten Pakete an den jeweiligen Nodes enthalten. Diese PCAP Dateien lassen sich mit dem Programm Wireshark betrachten. Alle Dateien wurden nach dem, auf dem Aufgabenblatt geforderten, Namensschema erstellt.

Erste Simulation:

ASCII-Trace: ReNe_SoSe_2013_PA2a_943147.tr

NetAnim : ReNe_SoSe_2013_PA2a_943147.xml

PCAP: ReNe_SoSe_2013_PA2a_943147_-NodeNr]-[DeviceNr]

Zweite Simulation:

ASCII-Trace: ReNe_SoSe_2013_PA2b_943147.tr

NetAnim : ReNe_SoSe_2013_PA2b_943147.xml

PCAP: ReNe_SoSe_2013_PA2b_943147_-NodeNr]-[DeviceNr]

7 Simulationsdurchführung und Datenerfassung

Die beiden Simulationen wurden mit den bereits oben gegebenen Parametern durchgeführt. Für die weiteren Untersuchungen wurde der von **ns-3** erstellte **ASCII-Trace**

verwendet. Zur späteren Weiterverarbeitung wurden aus dem ASCII-Trace die relevanten Daten extrahiert und in verschiedene CSV-Dateien geschrieben. Dabei wird ein CSV-Datei mit dem Durchsatz pro Zeiteinheit mit gesamtem Durchsatz, eines für den Durchsatz pro Zeiteinheit ohne gesamten Durchsatz und eines für den aufsummierten Datendurchsatz erstellt. Die geschieht mit Hilfe eines selbstgeschriebenen Python-Programmes (`ReNe_SoSe_2013_PA2_TR_Parser.py`).

CSV-Dateien:

Akkumulierter Durchsatz: `ReNe_SoSe_2013_PA2[a-b]_943147_accuthroughput.csv`

Durchsatz mit Summe: `ReNe_SoSe_2013_PA2[a-b]_943147_throughput.csv`

Durchsatz ohne Summe: `ReNe_SoSe_2013_PA2[a-b]_943147_throughput_stacked.csv`

8 Präsentation und Interpretation der Ergebnisse

Für die grafische Präsentation der Ergebnisse wird die Programmiersprache **R** ([RPRO]) genutzt, welche sich gut für die Erstellung von Plots und Diagrammen eignet. Für die grafische Darstellung wurde in den CSV-Files ein Zeitintervall von 1/4 Sekunden benutzt.

8.1 Erste Simulation

Bei der ersten Simulation ergab sich eine fast vollständige Auslastung der Verbindung zwischen Server und Router, wie auch in *Abbildung 2* zu sehen. (xy-Plot des Durchsatzes im Kapitel 10 *Abbildung 6*)

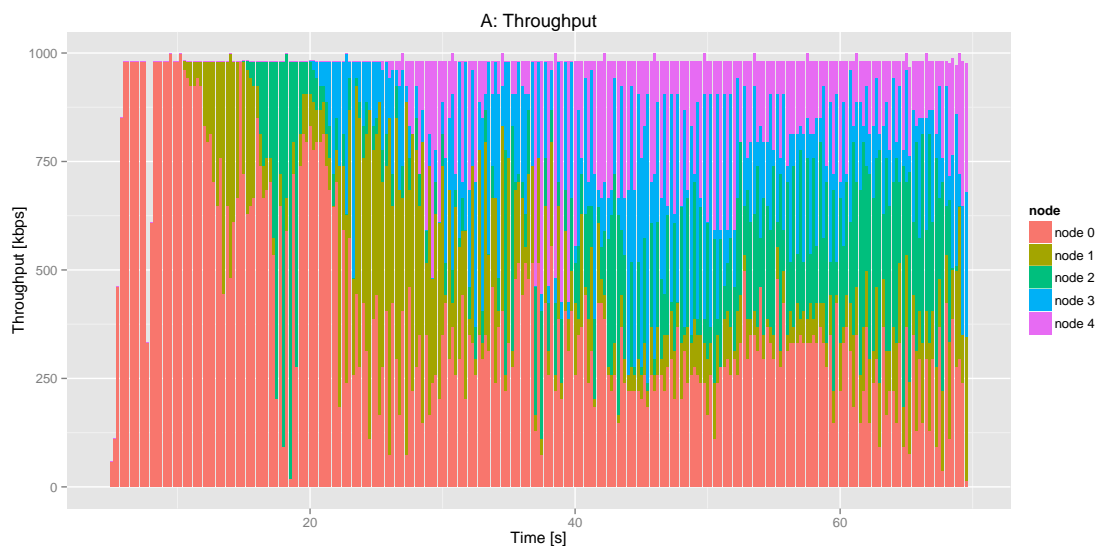


Abbildung 2: Simulation 1 - Durchsatz der einzelnen Clients

Da die Verbindung zwischen Server und Router der Flaschenhals des Aufbaus ist, ist nur ein maximaler Durchsatz von 1Mbps möglich, welcher unter den einzelnen Clients verteilt wird. Im Diagramm ist gut zu erkennen, dass die einzelnen Datenraten der Clients relativ stark schwanken, der Gesamtdurchsatz jedoch ziemlich konsant bleibt. Dies spiegelt sich auch im akkumulierten Durchsatz wieder. Diese steigt konstant an, bis die Verbindungen beendet werden, wie auch in *Abbildung 3* zu sehen.

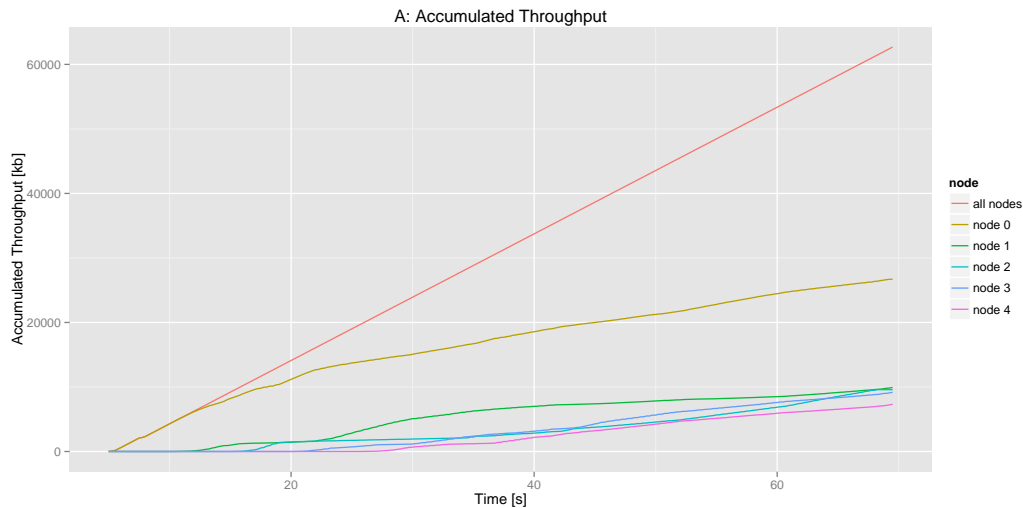


Abbildung 3: Simulation 1 - Akkumulierter Durchsatz

8.2 Zweite Simulation

Bei der zweiten Simulation zeigt sich ein komplett anderes Bild.

Hier schwankt nicht nur der Durchsatz der einzelnen Clients, sondern auch der Gesamtdurchsatz stark. Außerdem hat sich, wie in *Abbildung 4* zu sehen, der maximale Durchsatz halbiert und dass obwohl nur 10% der Pakete verloren geht. Der akkumulierte Durchsatz beträgt nur noch etwa 1/6 des Durchsatzes der ersten Simulation. (xy-Plot des Durchsatzes im Kapitel 10 *Abbildung 7*)

Dies lässt sich auf die Funktionsweise von TCP zurückführen, welches versucht sicherzustellen, dass alle gesendeten Packet ankommen und dabei versucht den höchst möglichen Durchsatz zu erreichen. Dazu wird für jedes Packet ein Acknowledgement gesendet, um den Empfang quasi zu quittieren. Geht nun ein Packet oder das dazugehörige Acknowledgement verloren, wird dies nach Ablauf des Timeouts vom Client erkannt und es ist eine Neuübermittlung nötig. Außerdem wird dem Client so vorgespielt, dass die Verbindung ausgelastet ist und die Frequenz der Paketsendung wird vom Client verringert. So kommt es in der zweiten Simulation dazu, dass die Congestion-Avoidance Phasen sehr kurz sind und damit das Congestion Window nicht sehr groß wird, da relativ viele Pakete verloren gehen und diese dadurch unterbrochen wird.

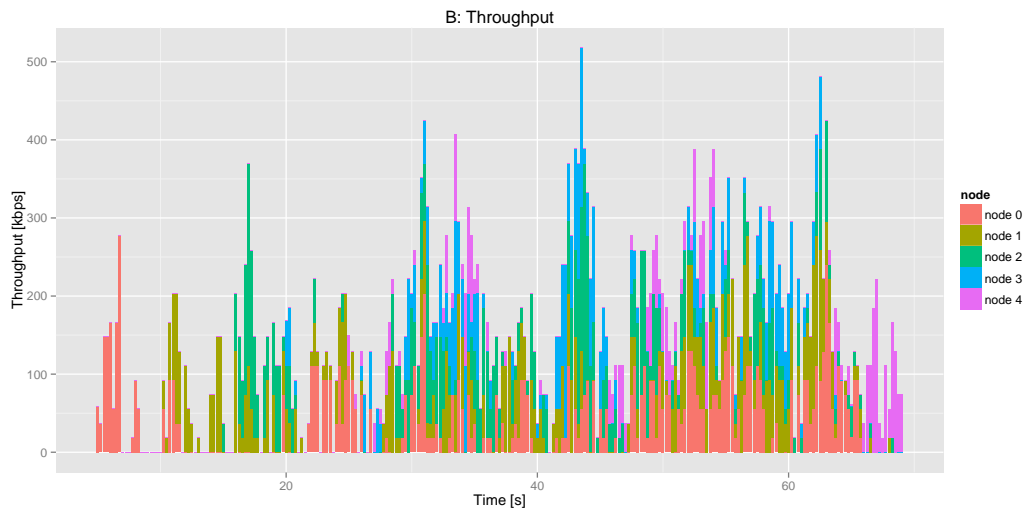


Abbildung 4: Simulation 2 - Durchsatz der einzelnen Clients

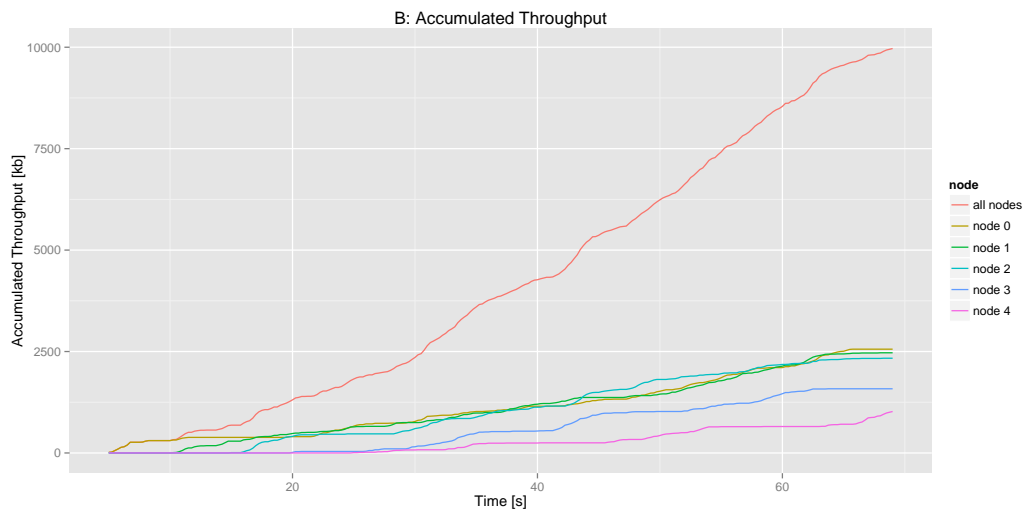


Abbildung 5: Simulation 2 - Akkumulierter Durchsatz

9 Fazit

TCP zielt auf die maximale Ausnutzung der verfügbaren Verbindung ab und stellt gleichzeitig auch die erfolgreiche Übermittlung aller Pakete sicher. Die Übermittlung wird dabei durch die Bestätigung der Paketübermittlung mittels Acknowledgements sichergestellt. Anhand des Verlusts von Acknowledgements bzw. von Pakete wird auch die Größe des Congestion Windows bestimmt. In *Simulation 1* sieht man, dass dieses Prinzip bei vollfunktionsfähiger Verbindung gut funktioniert und die Verbindung nahezu komplett auslastet. In *Simulation 2* erkennt man nun, dass bei einer fehlerhaften Verbindung die Datenrate massiv einbricht. Dies ist auf die verlustfreie Übermittlung aller Pakete zurückzuführen. Durch den Verlust von Datenpaketen oder Acknowledgements denkt TCP, dass die Verbindung schon vollständig ausgelastet ist und das Congestion Window kann aufgrund der häufigen Paketverluste nicht sehr stark vergrößert werden. Dies spiegelt sich dann in der Datenrate wieder, da das Congestion Window angibt, wieviele Pakete gesendet werden dürfen bis ein entsprechendes Acknowledgement empfangen wurde.

Diese beiden Simulationen zeigen die Stärken und Schwächen von TCP auf. In einem vollständig funktionsfähigem Netzwerk nutzt TCP fast die komplette verfügbare Datenrate aus und stellt eine vollständige Übertragung aller Datenpakete sicher. Im Gegensatz dazu, verschlechtert sich die Leistung von TCP in einem beschädigtem Netzwerk enorm. Es wird zwar noch immer eine sichere Übertragung aller Pakete sichergestellt, die Datenrate verschlechtert sich dadurch jedoch enorm. Daher wäre es empfehlenswert die beschädigte Netzwerkverbindung sofort zu reparieren, da hierdurch die Datenrate wieder enorm gesteigert werden könnte. Falls keine 100% sichere Ankunft der Pakete sichergestellt werden muss, kann auch ein anderes Protokoll genutzt werden, welches ohne Acknowledgements arbeitet und verlorene Pakete nicht neu überträgt.

10 Weitere Plots

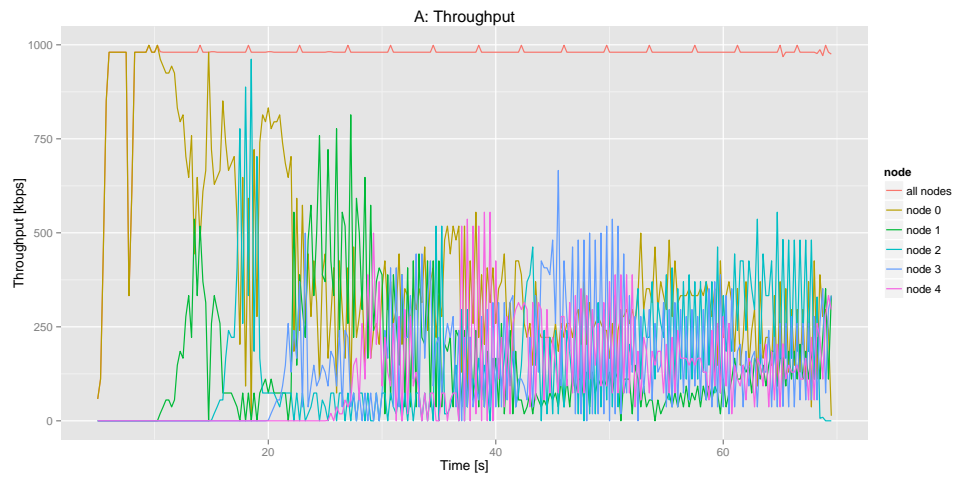


Abbildung 6: Simulation 1 - Durchsatz

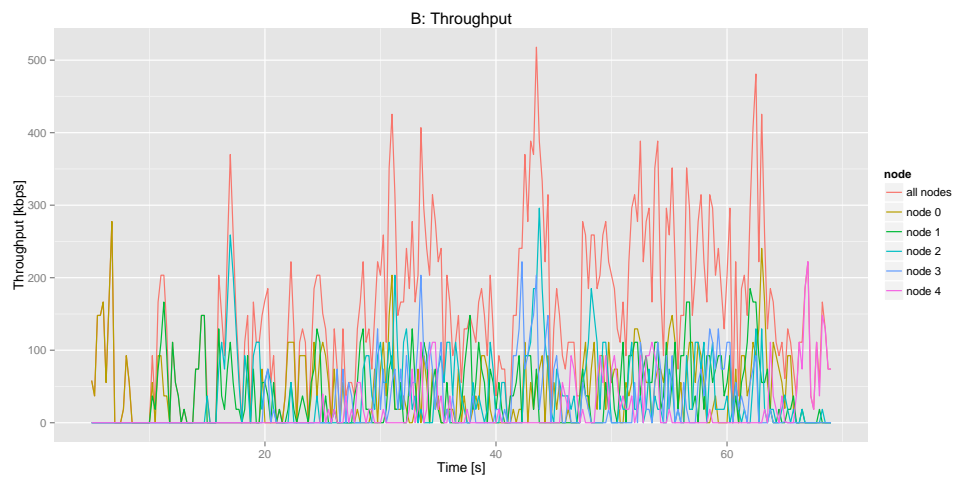


Abbildung 7: Simulation 2 - Durchsatz

11 Quellen

Quellenverzeichnis

[RNE3] Nils Aschenbruck, *Rechnernetze Kapitel 3 : Messungen und Simulation*, <http://sys.cs.uos.de/lehre/rene/2013/folien/Kap3-Messungen.pdf>, Juni 2013.

[PA2] Nils Aschenbruck, Jan Bauer, Alexander Bothe, Matthias Schwamborn, *Rechnernetze Praktisches Übungsblatt Nr.2*, http://sys.cs.uos.de/lehre/rene/2013/aufgaben/PA_Blatt02.pdf, Juni 2013.

[RPRO] R Project, *Webseite des R Project*, <http://www.r-project.org/>, Juni 2013.

[NS3] ns-3 project, *Webseite von nsnam*, <http://www.nsnam.org/>, Juni 2013.

[TCPNR] _____, *Webseite von nsnam*, [http://www.nsnam.org/docs/release/3.10 /manual/html/tcp.html](http://www.nsnam.org/docs/release/3.10/manual/html/tcp.html), Juni 2013.