

Praktikum Mobile AdHocNetze

Aufgabe 1: Experimente im Netzwerk-Simulator NS-3

Bearbeiter: Andy Labitzke, labitzke@studserv.uni-leipzig.de
 Gregor Tätzner, wir08ehh@studserv.uni-leipzig.de

04.11.2012

Inhalt

In diesem Dokument beschreiben wir die Lösung der 1. Praktikumsaufgabe. Die Gliederung orientiert sich an den einzelnen Unteraufgaben des vorgegebenen Arbeitsblattes.

A Laden und kompilieren des NS-3 Package

Wie in dem Tutorial vorgegeben haben wir die aktuellste Version des NS-3 Simulators installiert. Die Lösungen der weiteren Aufgaben haben wir für diese Version des Simulators entwickelt.

B Funktionsweise kennenlernen

Wir haben das Skript “first.cc” entsprechend des Tutorials erweitert und unserer Lösung beigefügt. Es trägt den Namen “tutorial_first_extended.cc”. Nachdem das Skript in den Ordner “scratch” der NS-3-Distribution kopiert wurde, kann es mit dem Befehl

```
./waf --run scratch/myfirst
```

ausgeführt werden. Dabei werden die folgenden Trace-Dateien erzeugt:

- myfirst-0-0.pcap
- myfirst-1-0.pcap
- myfirst.tr

Diese Dateien befinden sich im Ordner “output”. Die pcap-Dateien haben wir entsprechend der Anleitungen des Tutorials mittels *tcpdump* ausgewertet. Dazu wurden die folgenden Befehle genutzt:

```
tcpdump -nn -tt -r myfirst-0-0.pcap  
und  
tcpdump -nn -tt -r myfirst-1-0.pcap
```

Die Ausgaben von *tcpdump* sind in den folgenden Listings zu finden.

```
2.000000 IP 10.1.1.1.49153 > 10.1.1.2.9: UDP, length 1024  
2.514648 IP 10.1.1.2.9 > 10.1.1.1.49153: UDP, length 1024
```

Listing 1: Auswertung von myfirst-0-0.pcap

```
2.257324 IP 10.1.1.1.49153 > 10.1.1.2.9: UDP, length 1024  
2.257324 IP 10.1.1.2.9 > 10.1.1.1.49153: UDP, length 1024
```

Listing 2: Auswertung von myfirst-1-0.pcap

C Eigenes Testszenario

Für das Testszenario einer Ketten-Topologie mit variabler Länge haben wir das Skript “aufgabeEins.cc” erstellt. Nachdem der gesamte Projektordner “MANET_Praktikum” in den Ordner “scratch” der NS-3-Distribution kopiert wurde, kann die Simulation mit dem folgenden Befehl gebaut und ausgeführt werden:

```
make aufgabe1
```

Das Makefile fuehrt mehrere Simulationen mit verschiedener Hop-Anzahl durch. Dabei wird die jeweilige Topologielänge berechnet und ueber den Parameter *lTopologie* angegeben. Bei der Ausführung wird für jeden Knoten eine pcap-Trace-Datei erstellt, welches die empfangenen und verschickten Nachrichten enthält. Wir haben unserer Lösung einige der pcap-Dateien beigefügt, welche bei der Ausführung des Skriptes mit dem Wert 400 für den Parameter *lTopologie* entstanden sind. Diese tragen den Namen “aufgabeEins-<KnotenNummer>-0-pcap“. In den Dateien sind mehrere Ereignisse festzustellen. Zum einen sind AODV-Nachrichten protokolliert, welche durch das Routing verursacht werden. Dabei sind Anfragen zum Ermitteln der Route zwischen den einzelnen Knoten mit dem Text “rreq“ und Mitteilungen von Knoten, dass sie noch vorhanden sind, mit “rrep“ versehen. Des weiteren finden sich Nachrichten, welche mit dem Kürzel “ARP“ beginnen. Diese dienen der Ermittlung und Bekanntgabe der MAC-Adressen der an dem FTP-Fluss beteiligten Knoten. Abschließend sind auch noch die für die Auswertung notwendigen TCP-Nachrichten enthalten. Diese sind an dem Text “Flags“ zu erkennen. Dabei sind Nachrichten, welche für den Aufbau der TCP-Verbindung nötig sind, mit dem Flag “S“ versehen, während die Nachrichten für den Datentransport und die Bestätigung mit dem Flag “.“ versehen sind.

D Auswertung der Ergebnisse

Nach dem Durchlauf des Skriptes erhalten wir folgendes Ergebnis:

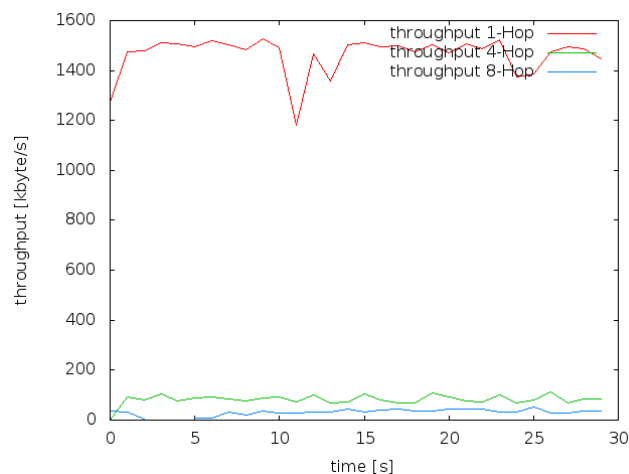


Abbildung 1: Throughput fuer FTP-Verbindung mit 1, 4 und 8 Hops

Offensichtlich ist die Höhe des Datendurchsatzes direkt von der Anzahl der Hops bzw. Knoten abhängig. Dabei gilt: Je mehr Knoten die Datenpakete durchlaufen müssen, desto langsamer ist die Verbindung.

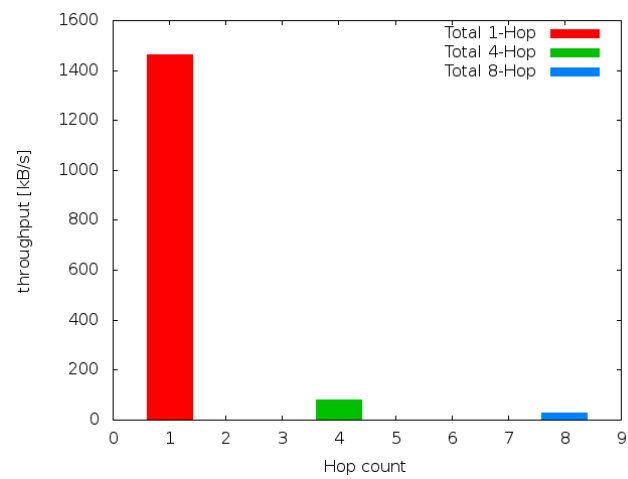


Abbildung 2: Mittlerer Durchsatz in Abhaengigkeit von Anzahl der Hops