

# **ITM-Praktikum**

## **Versuch 7: Mobile IP und OpenVPN**

Andreas Klingler, Hannes Stahl, Simon Lüke

30. Juni 2009

# Inhaltsverzeichnis

<b>1</b>	<b>Vorbereitende Fragen</b>	<b>2</b>
1.1	Simulationen . . . . .	2
1.2	NS v2 . . . . .	2
1.3	Schlüsselwörter in OTcl . . . . .	2
1.4	Network Animator . . . . .	2
1.5	Beispielasudruck . . . . .	2
1.6	.nam Datei . . . . .	2
1.7	node und link . . . . .	2
1.8	queue-type und queue-size . . . . .	3
1.9	queues . . . . .	3
1.10	OTcl Netzwerk Grundgerüst . . . . .	3
1.11	TCP- und UDP-Verbindung . . . . .	3
1.12	FTP- und CBR-Übertragung . . . . .	3
1.13	Farben und Knotenorientierung im NAM . . . . .	4
1.14	Packet Loss . . . . .	4
<b>2</b>	<b>Versuchsdurchführung</b>	<b>4</b>

# 1 Vorbereitende Fragen

## 1.1 Simulationen

Simulationen dienen zur Erprobung von Aufbauten, die ansonsten zu zeit- und kostenintensiv zu realisieren wären.

## 1.2 NS v2

Der NS v2 dient zur Simulation der meisten heute verwendeten Netzwerk-Topologien, es wird dabei z.B. TCP, Routing, Multicast und kabellose sowie kabelgebundene Netzwerke unterstützt. Er basiert auf...

## 1.3 Schlüsselwörter in OTcl

**proc** definiert eine Prozedur.

**set** schreibt und liest Variableninhalte.

**puts** gibt einen String aus.

**\$** ermöglicht den Zugriff auf eine zuvor deklarierte Variable.

**#** steht für Kommentare.

## 1.4 Network Animator

Der NAM (Network Animator) visualisiert die Simulation. Außerdem kann damit eine Simulation (Netztopologie und Datenströme) erstellt und eine Skriptdatei zur Steuerung an den Simulator übergeben werden. Der NAM ist damit ein Frontend für den NS.

## 1.5 Beispielasudruck

set lässt die Variable nf auf die Datei output.nam zeigen, output.nam wird dabei schreibend geöffnet (open ... w). Die nächste Zeile extrahiert alle namtrace relevanten Daten des Netzwerksimulators ns und schreibt diese in die Datei output.nam.

## 1.6 .nam Datei

Die .nam Datei enthält das Ergebnis der Simulation. Auf der Grundlage dieser Daten kann der NAM die Simulation visualisieren. In der .nam-Datei ist pro Zeile ein Paket zu finden, das zur Zeit x von Knoten a zu Knoten b gesendet wird.

## 1.7 node und link

**node** ist ein Knotenpunkt irgendeiner Art: z.B. eine Verzweigung, ein Endpunkt oder ähnliches.

**link** ist die Verbindung zwischen den verschiedenen Knoten.

## 1.8 queue-type und queue-size

**queue-type** Art der Warteschlangen, z.B. FIFO oder LIFO

**queue-size** Puffergröße der Warteschlange.

## 1.9 queues

Für die Queues sind die gleichen Typen wie bei Versuch Möglich: SFQ, FIFO, DRR, CBQ und RED.

## 1.10 OTcl Netzwerk Grundgerüst

```
% Neue Simulator Instanz erstellen
set ns [new Simulator]
% Zwei Knoten erstellen
set n0 [$ns node]
set n1 [$ns node]
% Link erzeugen
$ns duplex-link $n0 $n1 1.5Mb 12ms RED
% Queue-Size auf 15 setzen
$ns queue-limit $n0 $n1 15
```

## 1.11 TCP- und UDP-Verbindung

Eine TCP-Verbindung:

```
% Eine TCP Quelle (Agent) wird erzeugt und an Knoten n0 gebunden
set tcp [new Agent/TCP]
$ns attach-agent $n0 $tcp
% ... und die dazugehoerige Quelle an Knoten n1
set sink [new Agent/TCPSink]
$ns attach-agent $n1 $sink
% Nun werden noch Quelle und Senke verbunden. Warum?
$ns connect $tcp $sink
```

Analog dazu die Verbindungslose UDP Verbindung, ohne Spezifikation des Empfänger-Agenten:

```
set udp [new Agent/UDP]
$ns attach-agent $n0 $udp
set null [new Agent/Null]
$ns attach-agent $n1 $null
$ns connect $udp $null
```

## 1.12 FTP- und CBR-Übertragung

```
% Erzeugen des FTP-Agenten und binden an die TCP-Verbindung
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP

set cbr [new Application/Traffic/CBR]
$ftp attach-agent $udp
$ftp set type_ CBR
```

### 1.13 Farben und Knotenorientierung im NAM

**Farben** Mit dem Befehl

```
$ns color fid FARBE
```

lässt sich dem Datenstrom mit der Flowid fid die Farbe FARBE zuweisen.

**Orientierung** Der Befehl

```
duplex-link-op $Knoten1 $Knoten2 orient right
```

gibt an, dass Knoten2 rechts von Knoten1 angeordnet werden soll.

### 1.14 finish-Methode

Mit einer finish-Methode wird die Simulation beendet und NAM aufgerufen.

```
proc finish {} {  
    global ns nf  
    $ns flush-trace  
    close $nf  
    % NAM starten und Prozedur beenden  
    exec nam out.nam &  
    exit 0  
}
```

### 1.15 Packet Loss

Unter Packet Loss versteht man den Verlust von Paketen während der Übertragung. Mit genügend Bandbreite und Puffern kann dieser vermieden werden.

## 2 Versuchsdurchführung