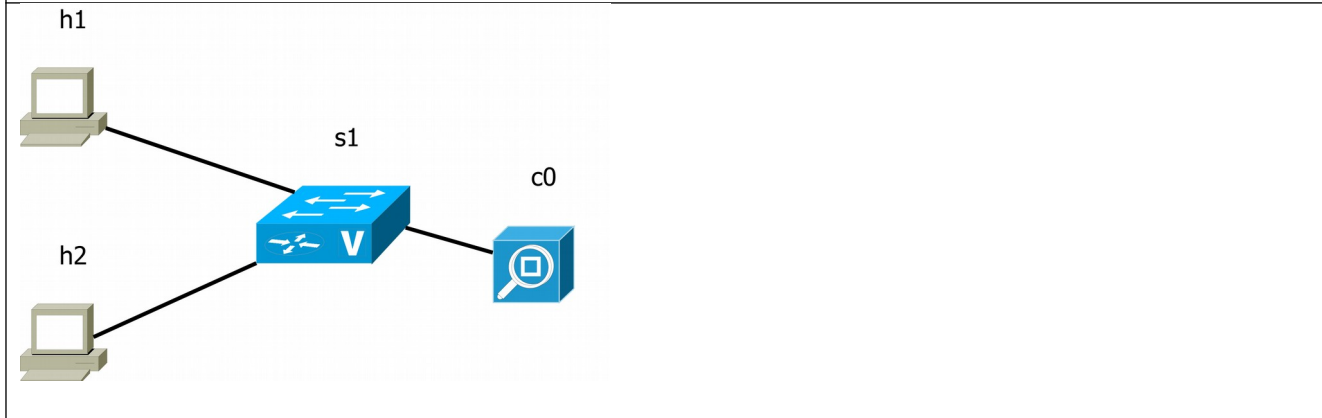


UEB 13: Mininet

Lerngebiet: FNT	Makrosequenz-Nr.: UEB 13 TCP Congestion-Control-Algorithmus: Reno vs. Cubic			
Zeitbedarf 2 Std.	Verfasser K. Dorau	Erstellungsdatum 20.04.2019	Überarbeiter	Überarbeitungsdatum 20.11.2019

Auftrag:



Aufgaben:

- Schalten Sie den Algorithmus einmal auf **cubic** und einmal auf **reno** mit `congestion.sh <algorithmus>`
- Starten Sie das Python-Skript `uebung13.py` einmal für **cubic** und einmal für **reno**
 - Begrenzen Sie die Datenrate auf 100Mbit/s im Python-Skript
 - Stellen Sie ein Delay der Verbindungen von 300ms (200-400ms) ein
 - Starten Sie auf h2 den `iperf3`-TCP-Server, Datenratenformat Mbit/s (-f), Ausgabe in eine Datei (>data.out)
 - Starten Sie auf h1 den `iperf3`-TCP-Client, 60s lang (-t), Datenratenformat Mbit/s (-f)
 - Wandeln Sie die Ausgabedatei des Servers in eine Gnuplot-Datei (`cubic/reno.dat`) um. Nutzen Sie das Bash-Skript `i2g.sh`. Aufruf des Skripts ohne Parameter zeigt die Syntax.
- Starten Sie beide Gnuplot-Skripte am Besten gleichzeitig

Fragen:

- Analysieren Sie die Messreihe beider Algorithmen
 - Bewerten Sie den Verlauf der Graphen. Begründen Sie, warum der aktuelle Linux-Kernel den Algorithmus cubic benutzt.

Informationen für den Einstieg:

- Netzwerk-Skript
- TCP, Slow Start
- `man iperf3`
- `i2g.sh`