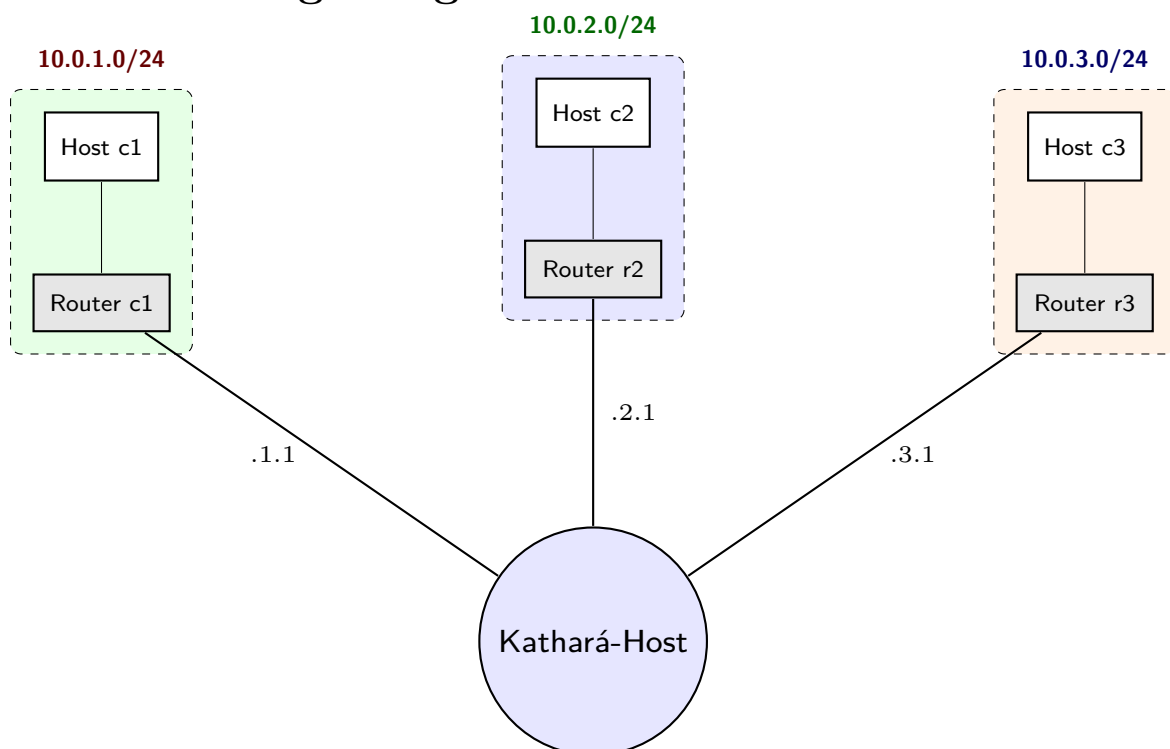


Teil 3: Transportschicht - Verhalten unter schlechten Netzwerkbedingungen

Einleitung

In diesem Praktikum analysieren Sie verschiedene Netzwerkbedingungen in einer simulierten Kathará-Umgebung. Ziel ist es, die Auswirkungen von Latenz, Paketverlust und Bandbreitenbegrenzung auf Netzwerkanwendungen und TCP-Verhalten zu untersuchen. Die Studierenden sollen dabei praktische Erfahrungen mit Messwerkzeugen wie `ping` und `iperf3` sammeln und das Verhalten von TCP-Verbindungen analysieren.

Netzwerkumgebung



Es werden 3 Netzwerke dargestellt, mit unterschiedlichen Einschränkungen der Verbindungsqualität:

A) Satellitenverbindung (Netzwerk 1)

- Sehr hohe Latenz (bei geostationären Satelliten $RTT \geq 500$ ms)

B) WLAN-Richtfunkstrecke (Netzwerk 2)

- Signalpfad wird durch Vegetation gestört, was zu erhöhtem Paketverlust führt

C) ISDN-Standleitung (Netzwerk 3)

- Stark begrenzte Bandbreite (64 kbit/s)

Werkzeuge

kathara

kathara lstart	Startet das gesamte Lab-Szenario basierend auf der aktuellen Verzeichnisstruktur.
kathara wipe	Beendet alle laufenden Container und löscht die temporären Dateien (<i>Alles weg</i>).
kathara connect c1	Öffnet ein weiteres Terminal-Fenster im bereits laufenden Container <i>c1</i> .

iperf3

Command	Description
<code>iperf3 -s</code>	Server starten (läuft bereits auf Laborrechnern)
<code>iperf3 -c <server></code>	TCP test (10s, 100 Mbps)
<code>iperf3 -c <server> -t 10</code>	Testdauer: 10 Sekunden
<code>iperf3 -c <server> -u -b 1M</code>	UDP test (1 Mbps)
<code>iperf3 -c <server> -w 100K</code>	TCP window size: 100 KB
Key Parameters	
<code>-b <rate></code>	UDP bandwidth (e.g. <code>-b 1M</code>)
<code>-w <size></code>	TCP window size (e.g. <code>-w 100K</code>)
<code>-t <time></code>	Test duration (seconds)
<code>-u</code>	UDP mode (default: TCP)

Netcat & Helfer

- **TCP-Transfer mit Echtzeit-Progress (GNU dd):**
`dd if=/dev/zero bs=1M count=10 status=progress | nc host0 5000`
- **TCP-Transfer mit Progress (Alpine/Busybox via pv):**
`dd if=/dev/zero bs=1M count=10 | pv | nc host0 5000`
- **Server für Datenübertragung (Gegenstelle):**
`nc -l 5000 > /dev/null`

Wichtige Hinweise:

- `status=progress` ist ein GNU-spezifisches Feature und in der Busybox-Version von `dd` (also in den Kathará-Containern) nicht verfügbar. Nutzen Sie dort `pv` (Pipe Viewer) als Alternative

Aufgaben

1. Netzwerkcharakterisierung

- Messen Sie die Round-Trip-Time (Latenz) und den Paketverlust für jedes Netzwerk mit `ping`.
- Bestimmen Sie die verfügbare Bandbreite mit `iperf3`.
- Vergleichen Sie die Messwerte zwischen den drei Netzwerken und dokumentieren Sie Unterschiede.

Hinweis: Auf den Lab-Rechnern läuft bereits ein iperf-Server. Benutzen sie daher `iperf3 -c` von den Containern aus in Richtung Host

2. Analyse von TCP-Verhalten

- Wiederholen sie die Performance-Tests bei laufender Wireshark-Aufnahme. Finden sie die speziellen Pakete, mit denen sich der TCP-Stack an die Netzwerkbedingungen anpasst.

Hinweis: Nützlich ist hier der Displayfilter `tcp.analysis.flags`, der alle Pakete mit Anomalien bei der Sequenznummeranalyse anzeigt

- Analysieren Sie, wie sich Latenz, Paketverlust und Bandbreitenbegrenzung auf den TCP-Durchsatz auswirken.
- Diskutieren Sie, welche Mechanismen in TCP für die beobachteten Effekte verantwortlich sind.

3. TCP-Algorithmen (optional)

Es ist möglich für einzelne Kathará-Hosts andere TCP Congestion Algorithmen zu wählen. Einige Optionen sind unter Anderem: *cubic (default)*, *bbr*, *westwood*, *reno*. Betrachten sie für diese Aufgabe nur die Verbindung über r2-c2, da hier am ehesten Unterschiede im Durchsatz durch verschiedene Strategien zu erwarten sind.

Hintergrund: Klassische TCP-Algorithmen wie Tahoe oder Reno setzen Paketverlust grundsätzlich mit Congestion auf dem Netzwerkpfad gleich. Jeder erkannte Verlust führt daher zu einer Reduktion des Sendefensters und des Datendurchsatzes, selbst wenn der Paketverlust nicht durch Überlastung, sondern z. B. durch fehlerhafte oder verlustbehaftete Leitungen verursacht wird.

- Untersuchen sie den Zustand des TCP-Stacks während der Dateiübertragung mittels `ss -ti`. Interpretieren sie die dargestellten Parameter, soweit bekannt.
- Stellen sie in der `lab.conf` den simulierten Host auf BBR-Algorithmus um:
`c1[sysctl]="net.ipv4.tcp_congestion_control=bbr"`
Danach müssen sie das Kathará-Szenario neustarten.
- Testen sie jetzt den Durchsatz mit `iperf3`, oder `netcat+dd`
- Testen sie noch mindestens einen anderen Algorithmus. Welcher erzielt im Vergleich die beste Durchsatzleistung?