

Quality of Service UDP

Aufbau

Um Paketverlust und Übertragungsfehler zu messen werden vom Server N Pakete erwartet und vom Client N Pakete gesendet. Anhand eines Timeouts wird bestimmt ob die Pakete tatsächlich angekommen sind oder ob in der Übertragung einige verloren gegangen sind. Die Veränderung wird anhand einer an die Daten angehängten CRC Summe über die Daten bestimmt.

Experiment

Netzwerk

Der Server ist eine Debian VM die per virtueller Brücke mit dem lokalen Netzwerk verbunden ist. Der Client ist ein Linux Desktop der direkt mit dem Lan verbunden ist. Innerhalb des bis zu Gigabit fähigen lokalen Netzes sind die Geräte von zwei Switches getrennt.

Nutzdaten

Die Nutzdaten bestehen aus 1024 Byte pro Paket zufällig generierten Daten und einem 8 Byte CRC zur Überprüfung der Integrität der Daten.

- Beginn: Tue Nov 17 Tue Nov 17 23:56:35 UTC 2020
- Ende: Tue Nov 17 23:59:58 UTC 2020
- IP Server: 192.168.0.71
- IP Client: 192.168.0.25
- Pakete gesendet: 10000
- Pakete empfangen: 10000 -> 100%
- Pakete: fehlerhaft: 0 -> 0%

Anmerkungen

Es wäre möglich gewesen eine Sequenznummer einzuführen und den Server bestätigen zu lassen das ein Paket mit einer bestimmten Nummer angekommen ist. Darauf wurde hier verzichtet um zum einen das Netzwerk nicht doppelt zu testen und um dem Wahnsinn TCP nachzuimplementieren zu entgehen. Stattdessen wurde angenommen das ein Paket wenn es nicht innerhalb einer Sekunde über eine LAN Verbindung angekommen ist, verloren ist. In der Praxis (bei absichtlicher Überlastung des Netzes) ist es nur vorgekommen das die Pakete nie angekommen sind und die letzten 20 - 30 erwarteten Pakete per Timeout als invalide markiert wurden.

Ein Test über das WLAN ergab die gleichen Ergebnisse.

Das Programm wurde per `javac org/gleumes/qos_udp.java` kompiliert.

Der Server wurde per `java org.gleumes.QoSUDP s 9090 1000 1000` ausgeführt.

Der Client wurde per `java org.gleumes.QoSUDP c 192.168.0.71 9090 1000` ausgeführt.

Latenz und Durchsatz

Latenz könnte man anhand der Zeit bestimmen die es braucht um ein Paket zum Server und zurück zu senden, jedoch ist im WAN nicht garantiert das für beide Richtungen der gleiche Weg genommen wird. Ein anderer Ansatz wäre sicherzustellen das beide Geräte die gleiche Zeit besitzen und als Payload die Zeit des Clients zu senden und diese auf dem Server zu vergleichen. Der Durchsatz ließe sich testen in dem man versucht die Anzahl der Pakete mit der Zeit zu erhöhen und zu messen ab welchem Punkt welche verloren gehen, wie es in ähnlicher Weise bei TCP gemacht wird.