

U 6.1 OMNeT++ installieren & TicToc-Tutorial

Installieren Sie den Netzwerksimulator OMNeT++ 5.2 (siehe <http://www.omnetpp.org>) und machen Sie sich mit der Software vertraut.

Arbeiten Sie sich in einem ersten Schritt durch das TicToc-Tutorial (siehe <http://www.omnetpp.org/doc/omnetpp/tictoc-tutorial/>) und erstellen bzw. starten Sie die entsprechenden Dateien. Modifizieren Sie dieses Tutorial, um sich mit OMNeT++ und dessen Eigenheiten vertraut zu machen. Beispielsweise können Sie die Netzwerktopologie verändern oder das Nachrichtenformat um eigene Felder zu erweitern, usw.

Gehen Sie vor allem auf folgende Themen bzw. Fragestellungen ein und verfassen Sie einen Erfahrungsbericht inkl. Screenshots:

- Wie wird eine Topologie erstellt und dessen Funktionalität realisiert?
- Kompilieren und Starten einer Simulation.
- Hinzufügen von graphischen Elementen, Ausgaben zur Fehlersuche, Zustandsvariablen und (zufälligen) Parametern.
- Vererbung, Verzögerung, Zeitüberschreitungen und dessen Aufhebungen.
- Netzwerktopologien mit mehr als zwei Knoten.
- Wie kann ein eigenes Nachrichtenformat definiert und verwendet werden?
- Hinzufügen von Statistiken zur anschließenden Auswertung und Visualisierung.

U 6.2 OMNeT++ .ned-Datei erstellen

Erstellen sie die folgende Topologie `simpleNet.ned` (siehe Abbildung 1) mit dem graphischen Editor.

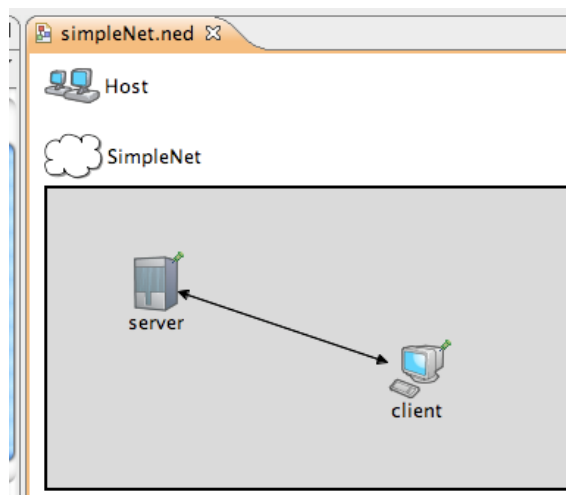


Abbildung 1: Beispiel-Netzwerktopologie.

Die Module `server` und `client` sollen als einfaches Modul dargestellt. Der darunterliegende Kanal (`channel`) soll eine Bandbreite von 10 Mbit/sec und eine Verzögerung (`delay`) von 5 Millisekunden haben. Erstellen Sie eine C++-Datei mit den notwendigen Klassendefinitionen und leeren Methodenimplementierungen.

U 6.3 OMNeT++ Request-Response-Protokoll

Implementieren Sie basierend auf U 6.2 ein simples Request-Response-Protokoll vergleichbar mit DNS. Der Client soll in regelmäßigen Abständen einen Request an den Server senden, der nur aus einem Hostnamen besteht. Der Server behandelt den Request und sendet nach einer kurzen Bearbeitungszeit eine Antwort (Response) zurück. Die Antwort besteht aus einem Statuscode (OK bzw. Fehler) und einer Adresse (4 Bytes). Definieren Sie geeignete Nachrichtenformate fixer Länge für Request und Response.

Halten Sie Ihre Implementierung einfach und verwenden Sie folgende Annahmen:

- Das Intervall zwischen den Anfragen des Clients sei exponentialverteilt mit Erwartungswert `intReq`, wobei `intReq` in der Konfigurationsdatei `omnetpp.ini` konfigurierbar sein sollte.
- Der Client soll insgesamt `n` Anfragen durchführen, wobei `n` ebenfalls via `omnetpp.ini` konfiguriert werden sollte.
- Die Anfrage bestehe aus einer zufälligen Zeichenkette (max. Länge von 50 Zeichen).
- Die Bearbeitungszeit des Servers sei normalverteilt mit $\mu = 1 \text{ ms}$ und $\sigma = 0.2 \text{ ms}$.

Nach der Bearbeitungszeit sendet der Server mit einer Wahrscheinlichkeit von 0.9 eine positive Antwort zurück (Statuscode OK). Die Adresse wird dabei zufällig generiert. Ansonsten wird ein negativer Statuscode zurückgesendet und die Adresse erhält den definierten Wert `0xFFFFFFFF`.