

Im Rahmen des Seminar „Concurrent Programming“ an der ZHAW wurde erfolgreich ein einfacher Mesh Switch entwickelt. Dieser musste in C unter der Verwendung von Pthreads implementiert werden. Ein einfaches Protokoll wurde in der Klasse definiert und festgehalten. Daraus ergab sich auch die Anforderung, dass die Implementationen der verschiedenen Studenten untereinander funktionieren müssen.

Der Mesh Switch hat die Aufgabe Pakete durch ein Mesh-Netzwerk zu routen um so von einem Anfangspunkt zu einem Endpunkt zu gelangen. Damit dies gelingen kann, wird anfänglich jeder Switch das Netzwerk fluten. Der Empfänger einer Nachricht bestätigt dem Sender den Erhalt. Daraus müssen sich die einzelnen Switche eine Route „berechnen“.

Das Protokoll definiert ein Paket wie folgt: Eine fixe Grösse von 132 Bytes ist aufgeteilt in eine 2- Byte-ID, eine Adresse (0 oder 1), den Typ des Paketes ('C','O','N') und ein 128 Byte grosser Inhalt. Der Mesh Switch ist in der Lage, die drei hier definierten Paket-Typen zu verarbeiten: Ein „Content“-Paket (C) muss an die Adresse (0 oder 1) gesendet werden. Ein „OK“-Paket (O) bestätigt den Empfang eines Paketes und muss den Weg zurück durch das Netzwerk finden, in der umgekehrten Reihenfolge wie ein C-Paket. Das „Neighbor“-Packet (N) weist den Mesh Switch an, sich mit einem anderen Switch zu verbinden. Die IP-Adresse und der Port werden dabei als Inhalt des N-Paketes übermittelt.

Die Implementation beruht auf reinem C, ohne zusätzliche Libraries. Die Verwendung von Threads erlaubt einfache blockierende Ein-/Ausgabe. Um mit verschiedenen Funktionen der Pthreads-Library zu experimentieren, aber auch um verschieden Datenstrukturen auszuprobieren, wurden neben einer blockierenden Queue auch eine Kombination aus einem Ring-Buffer und einer Hashtabelle implementiert.

Abbildung 1 zeigt die Architektur des Programs, „Meshy“ genannt. „Receiver Threads“, einen pro Verbindung empfangen Pakete vom Netzwerk. Pakete die weiter gesendet werden müssen, werden in eine Queue zur weiteren Verarbeitung gegeben. Eine Reihe von „Sender Threads“ bearbeiten diese Queue als Worker und senden anhand des „Routing“ das Paket weiter, entweder als Unicast oder als Broadcast. Der „Packet ID Cache“ verhindert, dass Pakete mehr als einmal weitergeleitet werden und stellt sicher, dass ein OK-Paket den Weg zurück findet.

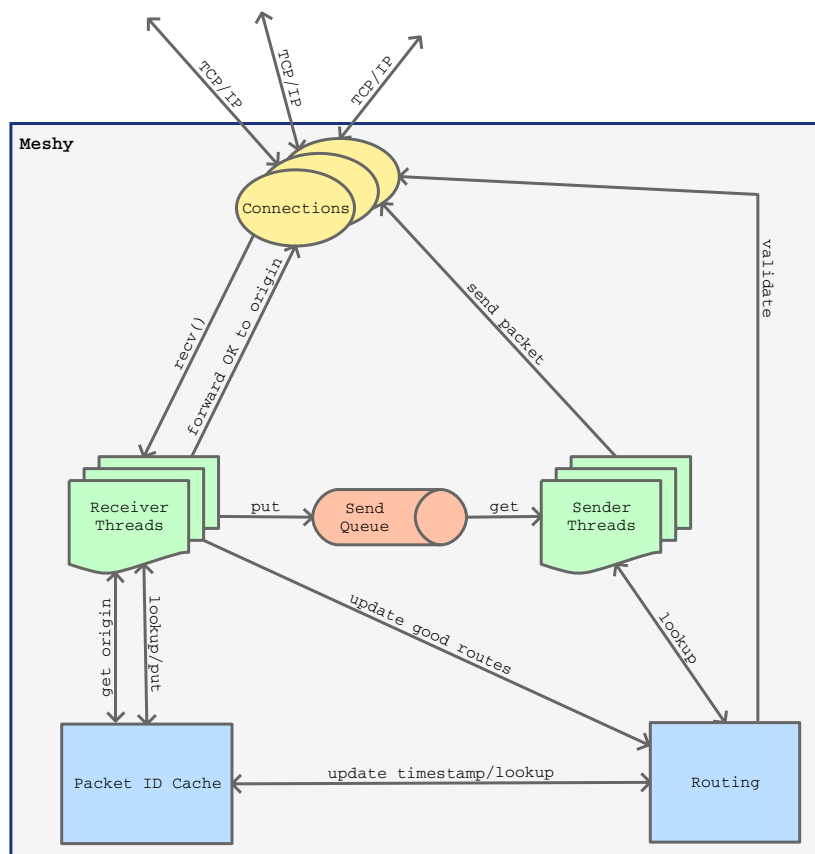


Abbildung 1: Übersicht der Architektur von Meshy