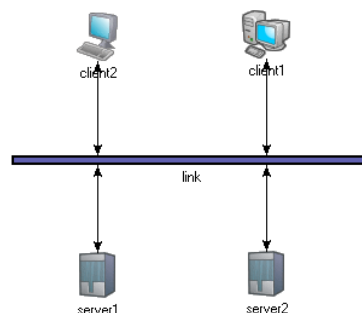
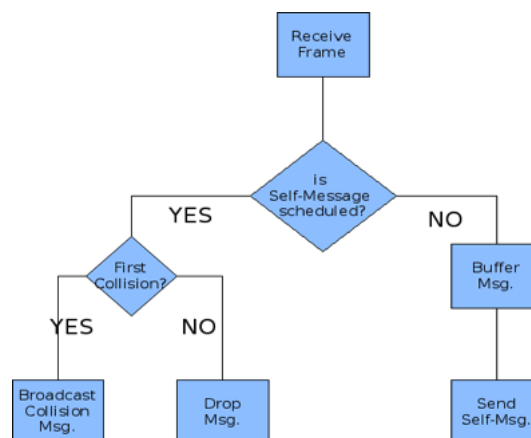


UE 7.1. Sicherungsschicht: ALOHA

Für die Sicherungsschicht soll eine Simulation des ALOHA-Protokolls implementiert werden. Bei diesem Protokoll kann jeder Host zu jeder Zeit einen Frame senden. Erfolgt eine Kollision, so wird der Frame einfach vom Host nach einem zufälligen Zeitabstand neu gesendet. Das geschieht so lange bis die Nachricht ohne Kollision ankommt.

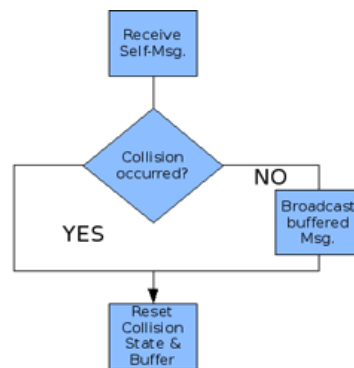


Das Netzwerk wurde im Vergleich zu den vorherigen Übungsblättern an die Aufgabe angepasst und enthält nun zwei Clients (*client1* und *client2*), zwei Server (*server1* und *server2*) und das ALOHA-Übertragungsmedium (*link*). Clients fragen bei den Servern um eine Berechnung an, *client1* kommuniziert mit *server1* und *client2* kommuniziert mit *server2*. Das ALOHA-Übertragungsmedium implementiert die Funktionalität der Bitübertragungsschicht und hat die Aufgabe eingehende ALOHA-Frames entgegenzunehmen und entweder (a) an alle verbundenen Hosts weiterzuleiten oder (b) im Falle einer Kollision alle Hosts darüber in Kenntnis zu setzen (d.h. leere cMessage mit dem Namen „COL“, es wird kein eigenes Protokoll verwendet). Um die Länge des ALOHA-Übertragungsmediums zu simulieren schickt das ALOHA-Übertragungsmedium eine Self-Message. Erst wenn die Self-Message empfangen wird und in der Zwischenzeit keine Nachricht empfangen wurde, wird der Frame an alle Hosts weitergeleitet. Die Funktionalität des ALOHA-Übertragungsmediums wird zur Verfügung gestellt.

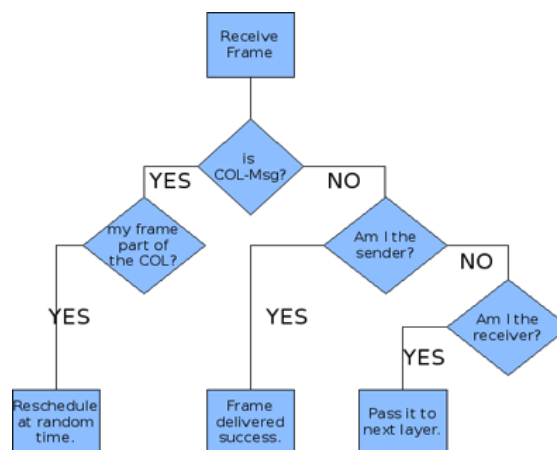


Obiges Flussdiagramm zeigt den Ablauf im ALOHA-Übertragungsmedium bei Erhalt eines Frames.

Der Erhalt einer Self-Message wird im folgenden Diagramm behandelt:



In der Sicherungsschicht der Hosts werden die Frames instanziiert, die IP-Datagramme werden in Frames verpackt und die Frames werden auf das ALOHA-Übertragungsmedium gelegt. Erhält der Host eine Nachricht vom ALOHA-Übertragungsmedium, so wird unterschieden, ob es sich um eine Kollisionsmeldung oder einen ALOHA-Frame handelt. Ist ein Frame eine Kollisionsmeldung (d.h. cMessage mit dem Namen „COL“) und ist der Frame der aktuellen Instanz von der Kollision betroffen, dann muss die Nachricht nochmals gesendet werden. Dies geschieht mit Hilfe des Befehls `sendDelayed`, der eine Nachricht nach einer gewissen Zeitspanne sendet. Ist es keine Kollisionsmeldung, dann wird unterschieden ob die aktuelle Instanz Absender, Ziel des Frames oder keines von beiden ist. Folgendes Flussdiagramm zeigt den Ablauf:



Ihre Aufgabe:

Implementieren Sie Sicherungsschicht der Hosts. Verwenden Sie dazu die bereitgestellten Stubs und ergänzen Sie den Quelltext in der Datei **Aloha.cc (aau_ALOHA_stub.zip)**.

Hinweise:

Verwenden Sie folgenden Befehl um Frames nach einer Kollision mit Zeitverzögerung erneut zu senden:

```
sendDelayed(pkt, simTime()+uniform(0.2, maxWaitTime), "outLowerLayer")
```

Achten Sie darauf, dass Sie keine Nachrichten doppelt verschicken dürfen. Verwenden Sie `cPacket->dup()` um Kopien von Nachrichten zu erstellen.