PR Rechnernetze und Netzwerkprogrammierung 621.801/621.802	WS 2017 / 2018
Institut für Informationstechnologie (ITEC)	Rainer/Timmerer
Übungsblatt 12	

U 12.1. Sicherungsschicht: CSMA/CA

Für die Sicherungsschicht soll eine Simulation des CSMA-CA-Protokolls (mit RTS/CTS Erweiterung) implementiert werden. Bei diesem Protokoll wird mit einem Request-To-Send (RTS) der Wunsch zu Senden ausgedrückt, dem der Empfänger mit einem Clear-To-Send (CTS) nachkommen kann. Alle anderen Teilnehmer hören diese Nachrichten ebenfalls und halten sich dementsprechend zurück, um eine kollisionsfreie Übertragung zu ermöglichen. Dadurch kann für die eigentlichen Nachrichten aus höheren Schichten das Problem der „*Hidden Station*“ umgangen werden.

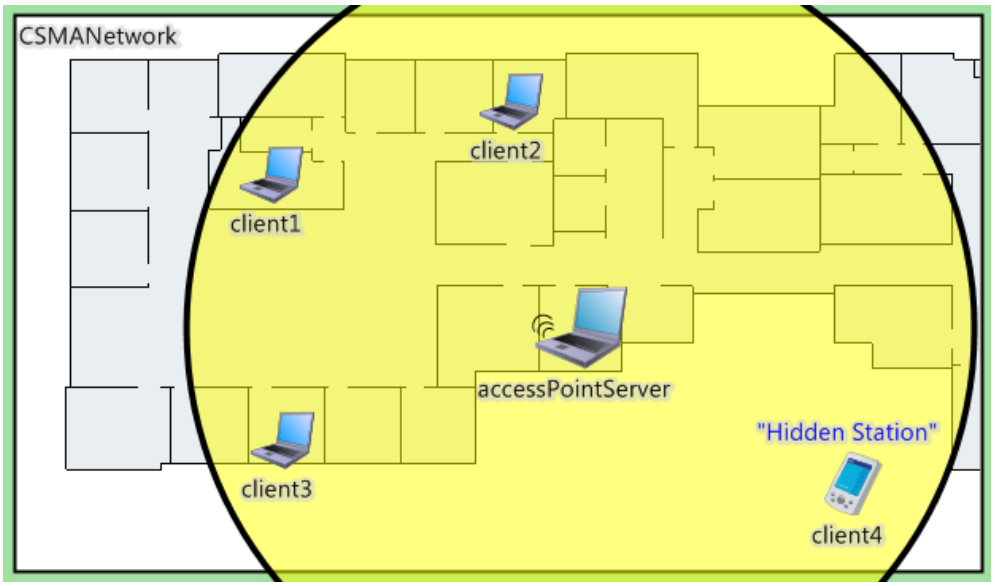

Abbildung 1: Aufbau des Netzwerks.

Das Netzwerk wurde im Vergleich zu den vorherigen Übungsblättern an die Aufgabe angepasst und enthält nun vier Clients (client1, client2, client3 und client4) und einen Server (accessPointServer), der einfachheitshalber auch ein Access Point der Clients ist. Alle Teilnehmer hören einander bei jeder Übertragung, mit Ausnahme von client4, welcher nur vom accessPointServer wahrgenommen wird. Dadurch wird das Problem der „*Hidden Station*“ simuliert.

Jeder Client möchte seine Nachricht so schnell wie möglich an den Server schicken. Um die Anzahl der Kollisionen möglichst gering zu halten, strebt das zu implementierende Protokoll folgenden Ablauf an:

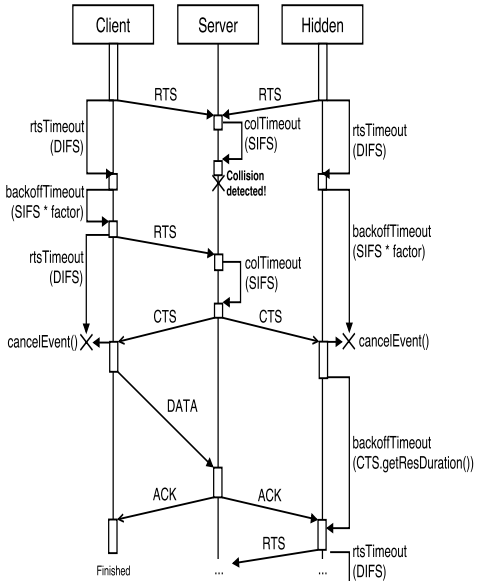

Abbildung 2: Ablauf des Protokolls.

Am Anfang senden alle Clients gleichzeitig einen RTS Frame. Um die Übertragungsdauer zu simulieren, verschickt der Server bei Erhalt eines Frames ein colTimeout der Länge SIFS (vorbereitete Konstante) an sich selbst. Treffen vor dem Timeout noch andere Nachrichten ein, wird statt eines neuen Timeouts ein Zähler erhöht. Ist dieser Zähler bei Erhalt des Timeouts > 1, hat eine Kollision stattgefunden und die Nachrichten werden verworfen.

Erhalten die Clients nach einer gewissen Zeit (rtsTimeout, DIFS) keine Antwort auf ihren RTS Frame, wird von einer Kollision ausgegangen. Um weitere zu vermeiden, warten sie eine zufällig bestimmte Zeit (backoffTimeout, SIFS * random factor) bevor ein erneuter Versuch gestartet wird.

Kann ein Sender einen RTS Frame kollisionsfrei empfangen, so wird mit einem CTS geantwortet. Der rechtmäßige Empfänger des CTS Frames kann darauf seine Nachricht aus der höheren Schicht an den Server schicken, der diese mit einem ACK bestätigt. Alle anderen Empfänger des CTS Frames brechen etwaige Timeouts ab und halten sich (mit einem neuen Timeout) für die Dauer der Reservierung zurück. Die gewünschte Zeitspanne (transmitDuration) wird im RTS Frame schon mitgesendet und über die CTS Frames an die anderen Teilnehmer verteilt.

Hinweise:

- Die Protokolle der höheren Schichten wurden bereits vollständig für Sie implementiert.
- Implementieren Sie die gewünschte Funktionalität ausschließlich in den Dateien CSMA.cc und CSMA.h, in den, mit entsprechenden Kommentaren gekennzeichneten, Quellcodebereichen. Zusammen mit den anderen Dateien ergibt das die vollständige Simulation.
- Die meisten benötigten Konstanten und Variablen sind bereits im Header (CSMA.h) des CSMA Moduls vorbereitet und werden in CSMA.cc mit den vorgesehenen Werten initialisiert.
- In dieser Simulation gibt es keine festen Verbindungen zwischen den Teilnehmern, da sie alle über WLAN kommunizieren. Verwenden sie zum Senden der Nachrichten die bereitgestellte Methode sendToAllReachableDevices(cMessage *msg).
- Sollten die Sourcefiles für <name>.msg Dateien nicht automatisch erzeugt werden, so generieren Sie diese manuell auf der Konsole (mingwenv.cmd im Omnet++ Installationsverzeichnis) mit zum Beispiel folgendem Befehl:

  opp_msgc <pfad zum projekt>/src/csma/<name>.msg