Carsten Noetzel 04.01.2012

Armin Steudte

# Technik und Technologie (Praktikum 3/4)

## Anleitung zum Programm

Das fertige Programm liegt gepackt als ausführbares Eclipse-Projekt im ZIP-Format vor.

#### Ausführen des Programms

1. Projekt in Eclipse importieren
2. In der Datei *„GUI.java“* muss die Konstante *HOST* auf die aktuelle IP-Adresse des Rechners angepasst werden
3. Bei Bedarf kann mit Hilfe der Konstanten *USERNAME* der gewünschte Username festgelegt werden
4. Danach das Programm über die *„GUI.java“* in Eclipse ausführen

#### Bedienung

1. Im Textfeld *„Username“* kann der Name des Clients eingetragen werden, welcher eingeladen werden soll
2. Im Feld *„Host“* kann die IP-Adresse des SIP-Proxies oder für eine direkte Kommunikation die IP-Adresse des entfernten UAS eintragen werden
3. Der Button *„Invite“* sendet eine **Invite**-Message an den entfernten UAS
4. Mit dem Button *„Bye“* kann eine **Bye**-Message gesendet und die Session abgebaut werden
5. In der linken Textarea werden die empfangenen Multicast-Nachrichten angezeigt
6. In der Textarea *„Inbound Invites“* werden die mit der lokalen UAS-Komponente verbundenen Clients angezeigt
7. Die Status der IGMP-Komponenten können über die definierten Label eingesehen werden

## D:\Repositories\WS_2011\TT1\Prak2\Ausarbeitung_Steudte_Noetzel\Grafiken\GUI.png

Abbildung 1 - Programmoberfläche

## Multicast Paketverteilung

In diesem Protokoll wird die Verteilung der Nachrichten per Multicast untersucht. Hierzu wird zunächst die Durchführung des Versuches beschrieben um im Anschluss auf die Beobachtungen eingegangen. Zum Schluss wird ein kurzes Fazit zu den in der Aufgabenstellung gestellten Fragen gezogen.

### Versuchsdurchführung

* Zu Beginn des Versuches fungierte unser Programm als Server (mit der IP 141.22.27.34) und hat sich beim Programmstart automatisch am SIP-Proxy registriert.
* Zuerst hat die Gruppe „Steenbuck/Harms“ (mit der IP 141.22.27.35) ein **Invite** an unseren UAS gesendet und im Vorfeld das Capturing mittels Wireshark gestartet.
* Daraufhin haben wir uns mit unserer UAC-Komponente lokal auf unseren UAS mittels **Invite** verbunden.
* Im Anschluss hat die Gruppe „Steenbuck/Harms“ die Session mit unserem UAS mittels **Bye**-Nachricht beendet.
* Zum Abschluss des Versuches haben auch wir die Session unseres UAC mittels **Bye** beendet.

### Beobachtungen

Beim Programmstart hat sich die UAS-Komponente beim SIP-Proxy registriert (vgl. Abbildung 2)

D:\Repositories\WS_2011\TT1\Prak2\Ausarbeitung_Steudte_Noetzel\Grafiken\1 SIP-Register.png

Abbildung 2 – Register UAS

In Abbildung 3 kann man sehen, dass über den SIP-Proxy (IP: 141.22.27.3) an unsere UAS-Komponente durch unsere Partnergruppe gesendet wurde und diese daraufhin der Multicastgruppe joined.

Auch erkannt man im dem Trace, dass das  **Invite** durch uns mit einem **OK** bestätigt und über den SIP-Proxy versendet wird.

Anschließend wird das **OK** durch ein **acknowledge** der Partnergruppe bestätigt.

Daraufhin beginnt unser Programm mit dem Senden von Nachrichten an die Multicastgruppe, zuerkennen an den UDP-Paketen inm Trace.

Zusätzlich fällt im Trace auf, dass das **IGMP-Join** durch den Betriebsystem-Stack

Zur Sicherheit zweimal, nach einem zufälligen Back-off, versendet wird.

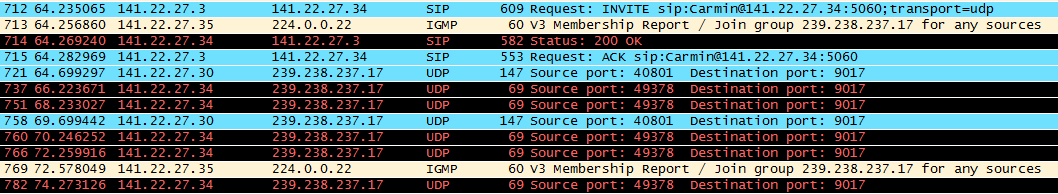


Abbildung 3 - Invite durch Gruppe "Steenbuck/Harms"

Nachdem wir, wie bereits in der Versuchsdurchführung beschrieben, mit einem lokalen UAC unseren UAS inviten sendet die Partnergruppe ein **Bye** an unseren UAS und beendet die Session (Abbildung 4).

Daraufhin sendet die Partnergruppe ein **IGMP-Leave** und verlässt die Multicastgruppe. Dabei fiel auf, dass trotz des Verlassens der Multicastgruppe weiterhin Multicasttraffic am Rechner der Partnergruppe ankam.

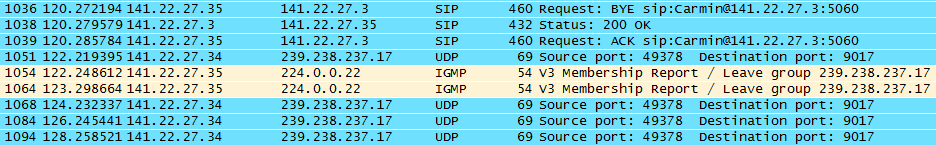


Abbildung 4 - BYE aus Sicht der Partnergruppe

Außerdem zeigt der Trace das Problem der Trennung zwischen UAC und UAS, das dazu führte dass jedes **OK** durch ein **ACK** des UAC bestätigt wurde.

Dieser Fehler wurde innerhalb des Praktikums behoben.

### Fazit

Die Beobachtung, dass Multicastpakete auch nach dem **IGMP-Leave** der Multicastgruppe weiterhin den Rechner erreichen, ist der Tatsache geschuldet, dass die Netzwerkinfrastruktur des Labornetzes sowohl auf IP- als auch auf Ethernet-Protokollebene keine Unterstützung für Multicast bietet.

Somit hat weder das **IGMP-Join,** als auch das **IGMP-Leave** eine Auswirkung auf den Multicastverkehr. Auch die sind die Switsches im Labornetz wohl nicht in der Lage Auf Ethernetebene die speziellen Mutlicast-Macadressen auszuwerten und somit findet auch auf dieser Ebene keine Steuerung des Multicatsverkehrs statt. Somit werden die Multicastpakete lediglich gebroadcastet.