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 dargestellten 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 per **Register**-Nachricht 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.
* Beim Eintreffen des Invites an unserem UAS beginnt der IGMP Sender mit dem Versand von Multicast Nachrichten.
* Daraufhin haben wir uns mit unserer UAC-Komponente lokal auf unseren UAS mittels **Invite** verbunden und als Ergebnis unsere Multicast-Nachrichten in der eingangs erwähnten Textarea angezeigt bekommen.
* 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.
* Nachdem keine Session mehr vorhanden war stoppte unser IGMP Sender mit dem Versand von Multicast Nachrichten.

### Beobachtungen

Beim Programmstart hat sich die UAS-Komponente beim SIP-Proxy registriert (vgl. Abbildung 2). Hierzu wurde an den Proxy (mit der IP 141.22.27.3) ein SIP **REGISTER** Requestgeschickt, auf das nach einer **TRYING** eine **OK**-Response vom Proxy folgte, womit wir am Proxy registriert waren.

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

Abbildung 2 – Register UAS

Abbildung 3 zeigt, dass über den SIP-Proxy eine **INVITE**-Nachricht durch unsere Partnergruppe (IP 141.22.27.35) an unsere UAS-Komponente gesendet wurde und diese daraufhin der Multicastgruppe beitritt (**IGMP-Join**).

Weiterhin erkennt man im dem Trace, dass das  **Invite** durch unseren UAS mit einer **OK**-Response bestätigt und über den SIP-Proxy versendet wird.

Anschließend wird das **OK** durch ein **acknowledge** der Partnergruppe bestätigt, welches wir über den Proxy erhalten.

Unser Programm beginnt daraufhin mit dem Senden von Nachrichten an die Multicastgruppe, zu erkennen ist dies an den UDP-Paketen im Trace mit der Ziel IP 239.238.237.17.

Zusätzlich fällt im Trace auf, dass das **IGMP-Join** der Partnergruppedurch den Betriebsystem-Stack zur Sicherheit zweimal, nach einem zufälligen Back-off, versendet wird.

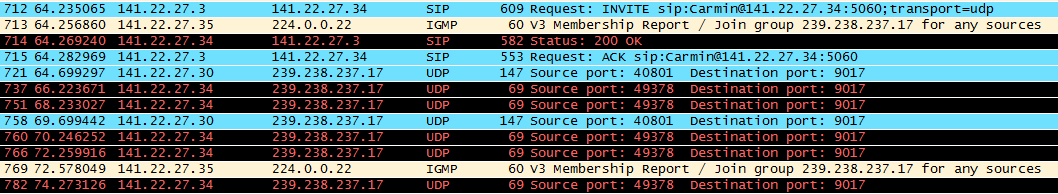


Abbildung 3 - Invite durch Gruppe "Steenbuck/Harms"

Nachdem wir, wie bereits in der Versuchsdurchführung beschrieben, mit unserem lokalen UAC eine Session mit unserem UAS aufgebaut haben, sendet die Partnergruppe ein **BYE** an unseren UAS und beendet damit die Session (vgl. Abbildung 4).

Daraufhin sendet die Partnergruppe ein **IGMP-Leave** und verlässt die Multicastgruppe. Hierbei ist zu beachten, dass trotz des Verlassens der Multicastgruppe weiterhin Multicastverkehr am Rechner der Partnergruppe ankam.

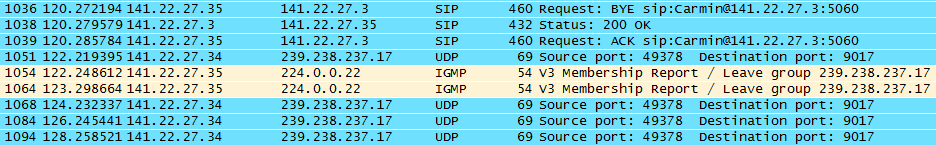


Abbildung 4 - BYE aus Sicht der Partnergruppe

Zudem 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 erkannt und 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,** noch das **IGMP-Leave** eine Auswirkung auf den Multicastverkehr. Zudem sind die Switches im Labornetz wohl nicht in der Lage auf Ethernetebene die speziellen Mutlicast-Macadressen auszuwerten, wodurch keine Steuerung des Multicastverkehrs auf dieser Ebene stattfindet. Somit werden die Multicastpakete lediglich gebroadcastet und erreichen daher auch nach dem **IGMP-Leave** weiterhin den UAC.