

# **ITM-Praktikum**

## **Versuch 5: Videostreaming: Setup mit Multicast/Unicast**

Andreas Klingler, Hannes Stahl, Simon Lüke

18. Juni 2009

# Inhaltsverzeichnis

<b>1</b>	<b>Vorbereitende Fragen</b>	<b>2</b>
1.1	Routing . . . . .	2
1.2	Unicast, Multicast und Broadcast . . . . .	2
1.3	Unterschiede, Vor-und Nachteile von Unicast und Broadcast . . . . .	2
1.4	Vermeidung der Nachteile . . . . .	2
1.5	Adressbereich von Multicast . . . . .	2
1.6	Bedeutung der TTL . . . . .	2
1.7	Abbildung der IP Adresse auf eine MAC Adresse . . . . .	2
1.8	IGMP . . . . .	3
1.9	DVMRP . . . . .	3
1.10	PIM Sparse Mode und PIM Dense Mode . . . . .	3
1.11	Reliable Multicast . . . . .	3
1.12	Verweigerung von Multicasts durch ISPs . . . . .	3
1.13	Multicast-Routing . . . . .	4
<b>2</b>	<b>Versuchsdurchführung</b>	<b>4</b>
2.1	IGMP Verkehr . . . . .	4
2.2	Unicast Übertragung . . . . .	4
2.3	Multicastübertragung . . . . .	5
2.4	Video on Demand . . . . .	6

# 1 Vorbereitende Fragen

## 1.1 Routing

Unter Routing versteht man das vermitteln von Paketen in unterschiedlich vermaschten Rechnernetzen, wobei für jedes Paket neu entschieden wird, welchen Weg es nehmen soll.

## 1.2 Unicast, Multicast und Broadcast

Eine **Unicast**-Nachricht wird von einem Sender ausgesendet und vom anderen dedizierten Teilnehmer empfangen oder auch nicht (Ende-zu-Ende). Dagegen erreicht ein **Multicast** mehrere unterschiedliche Empfänger. Ein **Broadcast** eines Senders wird pauschal an alle Teilnehmer eines Netzes geschickt, aber nicht in andere Subnetze geroutet.

## 1.3 Unterschiede, Vor- und Nachteile von Unicast und Broadcast

- Unterschiede: Unicast  $\Rightarrow$  Ende-zu-Ende Nachrichten, Broadcast  $\Rightarrow$  ein Sender, viele Empfänger (auch Teilnehmer die das Paket nicht brauchen und verwerfen)

	Vorteile	
Broadcast	Bandbreite für Sender vervielfacht sich nicht	Nachrichten bleiben nur innerhalb des Netzwerkes
Unicast	Nur gezielt gewählte Teilnehmer bekommen die Nachricht	bei vielen Empfängern vorteilhaft

## 1.4 Vermeidung der Nachteile

Die Nachteile können mit Multicast vermindert werden. Mit Multicast wird ein Paket nur an mehrere, ausgewählte Empfänger versendet, obgleich der Sender nur eines Paket ausgesendet hat. Damit ist es sehr Trafficschonend für den Sender. Allerdings werden Multicasts immer seltener geroutet, weshalb multicastfähige Netzwerke nun über Backbonetunnel verbinden muss.

## 1.5 Adressbereich von Multicast

Der vorgesehene Adressbereich ist von 224.0.0.0 bis 239.255.255.255 (mit einigen Einschränkungen in bestimmten IP Bereichen). Diese Adressen kann sich ein Client zusätzlich zuweisen, um einen bestimmten Stream zu empfangen.

## 1.6 Bedeutung der TTL

Die TTL ist bei Multicasts extrem wichtig, da sonst die Pakete endlos weitergeschickt werden, sobald eine Ring o. ä. in der Vernetzung auftritt. Es gibt ja keinen direkten Empfänger, bei dem das Paket als zugestellt gilt.

Allgemein wird die TTL im Falle von Multicast relativ nieder gewählt, damit die Netzbelastung nieder bleibt. Unter Umständen kann dann ein über viel Hops verbundener Rechner einen Multicaststream nicht mehr empfangen, sollte die TTL zu nieder sein.

## 1.7 Abbildung der IP Adresse auf eine MAC Adresse

Die unteren 23 Bit einer IP Adresse werden auf die unteren 23 Bit einer MAC Adresse abgebildet. Die erste MAC Adresse ist dabei auf 01-00-5e-00-00-00 festgelegt worden. Werden von dieser

Adresse an die 23 LSB aufgefüllt, ergeben sich 2 volle Oktette (=16 Bit)+ ein halbvoller Oktett (=20 Bit) + 3 Bit im letzten halben Oktett. Damit ist die höchste Adresse die 01-00-5e-7f-ff-ff

## 1.8 IGMP

**IGMP** (Internet Group Management Protocol) wird bei Multicastanwendungen verwendet, um einem Host oder Zwischenrouter klar zu machen, dass ein Client nun einen speziellen Stream empfangen will. Dies verursacht aber noch nicht die Entsendung und Verteilung des Streams, sondern leitet dies nur ein. Die richtige Verteilung wird mittels DVMRP oder PIM zwischen den Routern initiiert, der Sender interessiert sich dabei nicht für IGMP, PIM o. ä..

## 1.9 DVMRP

**DVMRP** (Distance Vector Multicast Routing Protocol) ist ein veraltetes Protokoll zur Ermittlung des kürzesten oder kostengünstigsten Weges bei Streaminganwendungen. Bei diesem Verfahren hält der Router eine Tabelle bereit, in welcher die nächsten Router für die gewünschten Inhalte eingetragen sind. Die Kommunikation für dieses Verfahren wurde mit IGMP Paketen realisiert. Das Protokoll wird heute kaum noch verwendet.

## 1.10 PIM Sparse Mode und PIM Dense Mode

**PIM** (Protocol Independent Multicast) erfüllt ähnliche Zwecke wie das DVMRP: Es soll Multicasts effizient über mehrere Knoten hinweg leiten. Prinzipiell kann es in zwei Modi arbeiten:

- **Sparse-Mode:** Im Sparse Mode fragt ein Router einen Broadcast bei einem Rendezvous-Server an und bekommt von ihm den nächsten Streamknoten vermittelt. Somit versorgt dieser anfragende Router nur sein eigenes Subnetz. Es eignet sich also für geringe Dichten im Subnetz.
- **Dense-Mode:** Wenn bei einem übergeordneten Router eines Netzes ein Multicast angefragt wird, beginnt er diesen Multicast an alle Unterrouter in seinem Netz diesen Broadcast zu schicken. Jeder Unterrouter könnte nun sein Netz mit diesem Stream versorgen. Somit macht dieses Verfahren nur Sinn, wenn in diesen ganzen Unternetzen auch Teilnehmer vorhanden sind.

## 1.11 Reliable Multicast

Beim Reliable Multicast ist eine Möglichkeit implementiert, verlorene Pakete zu dedektieren und neu anzufordern. Dies kompensiert die Schwächen von UDP, ohne auf UDP zu verzichten. Allerdings ist das Handling von „Reliable Multicast“ grenzwertig: Sendet man ein Paket neu, ergeht dieses an alle aufgeschalteten Streamempfänger. Andererseits könnten, wenn ein Paket früh verloren geht, sehr viele Empfänger ein verlorenes Paket anfordern. ⇒ UDP ist dafür einfach nicht ausgelegt!

## 1.12 Verweigerung von Multicasts durch ISPs

Die ISPs wollen ihre Leitungen schonen. Sie müssen die Pakete letztendlich verteilen, während der Sender nur den Traffic einer einzelnen Upstreamverbindung bezahlen muss. Außerdem ist der Umgang mit den Multicastadressen recht schwierig, da es keine zentrale Verwaltung und Registratur dieser Adressbereiche gibt.

### 1.13 Multicast-Routing

- Ein Server sendet einen Stream an z. B. an die 225.1.1.1 (Destination Feld im IP Frame!)
- Ein Client sendet einen "Join" für diese Gruppenadresse via IGMP an seinen Router
- Die zwischengeschalteten Router bauen einen Versorgungsbaum bis in das Subnetz des Empfängers auf, wobei dieser Baum mehrere Verzweigungen in andere Subnetze aufweisen kann. Prinzipiell kann es nun an jedem auf dem Weg liegenden Router einen Verzweigungspunkt geben.
- Der Client erhält nun in seinem Subnetz Pakete die an die 255.1.1.1 gerichtet sind, obwohl er eigentlich eine ganz andere IP Adresse für seine Kommunikation benutzt.

## 2 Versuchsdurchführung

### 2.1 IGMP Verkehr

Wir starteten einen normalen Capture auf „timestamp“ und starteten den MRTD Server. Dabei sind vorerst keine Pakete eingegangen.

Ein weiterer Caputure im Promiscuous Mode zeigte uns folgende Pakete:

No. .	Time	Source	Destination	Protocol	Info
1	0.000000	10.5.20.99	224.0.0.13	PIMv2	Hello
2	12.015373	10.5.20.99	224.0.0.1	IGMP	V2 Membership Query
3	16.426831	10.5.20.99	224.0.0.251	IGMP	V2 Membership Report
4	16.494824	10.5.20.99	224.0.0.2	IGMP	V2 Membership Report
5	20.236864	10.5.20.99	224.0.0.13	IGMP	V2 Membership Report
6	59.048510	10.5.20.99	224.0.0.13	PIMv2	Hello
7	79.052860	10.5.20.99	224.0.0.13	PIMv2	Hello
8	97.083762	10.5.20.99	224.0.0.13	PIMv2	Hello

Abbildung 1: IGMP Verkehr an timestamp

Anscheinend hat also der MRTD Server diese Pakete über sein Interface 10.5.20.99 ausgesendet, aber Timestamp hat sich nicht für diese Pakete interessiert und diese verworfen. Dies Netzwerkkarte nimmt die Pakete mit der „Multicast-Macadresse“ im Destinationfeld nicht an, somit musste in den Promiscuous Mode geschaltet werden.

Im Screenshot (Bild 1) spielte sich der gesamte Verkehr für dieses Subnetz im Multicast Adressbereich ab:

- 224.0.0.13 ist für die PIM Kommunikation zwischen den Routern
- 224.0.0.1: „Membership Query“ des Routers. In gewissen Zeitabständen prüft der Router, ob noch Members in einer Gruppe sind oder ob er die Gruppe auflösen kann.
- 224.0.0.251 scheint eine spezielle mDNS (Multicast DNS) Nachricht zu sein
- 224.0.0.2 teilt anderen Routern mit, dass ein Multicastserver verfügbar ist

### 2.2 Unicast Übertragung

Bei einer Unicastübertragung von „ping“ nach „timestamp“ konnte im Netz von „checksum“ keine Aktivität festgestellt werden. Es ließen sich demnach auch keine Videodaten empfangen. Auch die Mitschnitte im Netz von „ping“ und „timestamp“ sind wenig spektakulär: Es wurden keine Multicastadressen verwendet, sondern direkt die IP Zieladressen und Ports in die Pakete eingesetzt, ohne Absprachen mit dem MRTD Servern.

## 2.3 Multicastübertragung

Nun senden wir vom Rechner „ping“ einen Multicast Stream an die IP 225.1.1.1. Wie der folgende Screenshot (Bild 2) zeigt, gehen von der IP 10.5.30.1 (Rechner „ping“) keine Steuerpakete per IGMP oder PIM aus, der Stream wird einfach per UDP an die 225.1.1.1 gesendet:

No. .	Time	Source	Destination	Protocol	Info
1	0.000000	10.5.30.99	224.0.0.1	IGMP	V2 Membership Query
2	5.327558	10.5.30.99	224.0.0.13	IGMP	V2 Membership Report
3	5.558365	10.5.30.99	224.0.0.2	IGMP	V2 Leave Group
4	5.559244	10.5.30.99	224.0.0.2	IGMP	V2 Leave Group
5	9.027241	10.5.30.99	224.0.0.251	IGMP	V2 Membership Report
6	28.648838	10.5.30.99	224.0.0.1	IGMP	V2 Membership Query
7	28.650124	10.5.30.99	224.0.0.13	PIMv2	Hello
8	28.653962	10.5.30.99	224.0.0.2	IGMP	V2 Membership Report
9	28.653994	10.5.30.99	224.0.0.13	IGMP	V2 Membership Report
10	28.664284	10.5.30.99	224.0.0.13	PIMv2	Hello
11	28.829913	10.5.30.99	224.0.0.13	IGMP	V2 Membership Report
12	30.050026	10.5.30.99	224.0.0.13	IGMP	V2 Membership Report
13	32.326268	10.5.30.99	224.0.0.251	IGMP	V2 Membership Report
14	35.654887	10.5.30.99	224.0.0.2	IGMP	V2 Membership Report
15	37.750282	10.5.30.99	224.0.0.2	IGMP	V2 Membership Report
16	47.824301	10.5.30.99	224.0.0.13	PIMv2	Hello
17	59.858350	10.5.30.99	224.0.0.1	IGMP	V2 Membership Query
18	64.089426	10.5.30.99	224.0.0.13	IGMP	V2 Membership Report
19	64.981508	10.5.30.99	224.0.0.2	IGMP	V2 Membership Report
20	67.557768	10.5.30.99	224.0.0.251	IGMP	V2 Membership Report
21	71.874668	10.5.30.99	224.0.0.13	PIMv2	Hello
22	105.925992	10.5.30.99	224.0.0.13	PIMv2	Hello
23	128.960289	10.5.30.99	224.0.0.13	PIMv2	Hello

Abbildung 2: IGMP Verkehr an ping während dem Streamaufbau

Weiterhin haben wir festgestellt, dass im weiteren Verlauf des Streamens keine weiteren Ereignisse am Streamingserver „ping“ eingehen. Er streamt einfach „blind“ an die 225.1.1.1 egal ob Rechner hinzukommen oder den Stream verlassen.

Nun betrachten wir einmal konkret den Fall, dass ein Rechner sich für einen Stream anmeldet. Diesen Fall illustriert der folgende Screenshot (Bild ??) ab Paket 18:

No. .	Time	Source	Destination	Protocol	Info
1	0.000000	10.5.10.99	224.0.0.1	IGMP	V2 Membership Query
2	0.000791	10.5.10.99	224.0.0.13	PIMv2	Hello
3	0.007562	10.5.10.99	224.0.0.2	IGMP	V2 Membership Report
4	0.007575	10.5.10.99	224.0.0.13	IGMP	V2 Membership Report
5	0.025840	10.5.10.99	224.0.0.13	PIMv2	Hello
6	2.631602	10.5.10.99	224.0.0.251	IGMP	V2 Membership Report
7	8.151694	10.5.10.99	224.0.0.13	IGMP	V2 Membership Report
8	9.443723	10.5.10.99	224.0.0.2	IGMP	V2 Membership Report
9	9.895728	10.5.10.99	224.0.0.2	IGMP	V2 Membership Report
10	12.207777	10.5.10.99	224.0.0.13	IGMP	V2 Membership Report
11	30.052508	10.5.10.99	224.0.0.13	PIMv2	Hello
12	31.057059	10.5.10.99	224.0.0.1	IGMP	V2 Membership Query
13	31.748176	10.5.10.99	224.0.0.2	IGMP	V2 Membership Report
14	37.148276	10.5.10.99	224.0.0.13	IGMP	V2 Membership Report
15	39.132316	10.5.10.99	224.0.0.251	IGMP	V2 Membership Report
16	57.081049	10.5.10.99	224.0.0.13	PIMv2	Hello
17	99.125858	10.5.10.99	224.0.0.13	PIMv2	Hello
18	111.824390	10.5.10.1	225.1.1.1	IGMP	V2 Membership Report
19	113.776487	10.5.10.1	225.1.1.1	IGMP	V2 Membership Report
20	123.569075	10.5.10.1	225.1.1.1	IGMP	V2 Membership Report
21	202.647716	10.5.10.1	224.0.0.2	IGMP	V2 Leave Group
22	202.648368	10.5.10.99	225.1.1.1	IGMP	V2 Membership Query

Abbildung 3: IGMP Verkehr an checksum beim Beitreten zu Stream 225.1.1.1

- Pakete 18, 19 und 20 wurden von der IP 10.5.10.1 („checksum“) als „Membership Report“ an unsere Multicastadresse 225.1.1.1 gesendet. Dadurch sagt „checksum“ dem Router, dass er diesen Stream empfangen will. Ab jetzt fließen die Multimediadaten als UDP Pakete in dieses Subnetz.
- Nach ca. 80 Sekunden meldet sich „checksum“ von diesem Stream mit einer „Leave Group“ Nachricht ab. Diese Nachricht ist an die IP 224.0.0.2 Adressiert und wird somit von jedem Router empfangen.

Analog verhält es sich, wenn in einem anderen Subnetz ein Client einen Stream anfordert: Als wir beispielsweise mit „timestamp“ ebenfalls diesen Stream angefordert haben, hat dieser die gleiche An- und Abmeldeprozedur durchgeführt, ohne die anderen Subnetze zu beeinflussen. Auch sendet „ping“ weiterhin seine UDP Pakete an die 225.1.1.1, obwohl alle Rechner aus dem Stream ausgetreten sind.

## 2.4 Video on Demand

Wir haben nun auf einem Webserver Video-on-Demand in unserem Testnetzwerk angeboten. Der Server war wieder „ping“ mit der IP 10.5.30.1 und der Client war „checksum“ mit der IP 10.5.10.1.

No.	Time	Source	Destination	Protocol	Info
18	10.692155	10.5.30.99	224.0.0.13	PMv2	Hello
19	10.696005	10.5.30.99	224.0.0.2	IGMP	V2 Membership Report
20	10.696041	10.5.30.99	224.0.0.13	IGMP	V2 Membership Report
21	10.821649	10.5.30.99	224.0.0.13	PMv2	Hello
22	13.768233	10.5.30.99	224.0.0.2	IGMP	V2 Membership Report
23	16.228481	10.5.30.99	224.0.0.13	IGMP	V2 Membership Report
24	17.049381	10.5.10.1	10.5.30.1	TCP	2329 > www [SYN] Seq=0 Len=0 MSS=1460 TSV=2293109 TSER=0 WS=2
25	17.049574	10.5.30.1	10.5.10.1	TCP	www > 2329 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 TSV=2290434 TSER=2293109 WS=2
26	17.049936	10.5.10.1	10.5.30.1	TCP	2329 > www [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSV=2293109 TSER=2290434
27	17.050345	10.5.10.1	10.5.30.1	TCP	[TCP segment of a reassembled PDU]
28	17.050473	10.5.30.1	10.5.10.1	TCP	www > 2329 [ACK] Seq=1 Ack=53 Win=5792 Len=0 TSV=2290434 TSER=2293109
29	17.050806	10.5.10.1	10.5.30.1	HTTP	GET /MATRIX_de_ultra.mpg HTTP/1.1
30	17.050851	10.5.30.1	10.5.10.1	TCP	www > 2329 [ACK] Seq=1 Ack=195 Win=6864 Len=0 TSV=2290434 TSER=2293109
31	17.124571	10.5.30.99	224.0.0.2	IGMP	V2 Membership Report
32	17.461063	10.5.30.1	10.5.10.1	HTTP	HTTP/1.1 206 Partial Content (video/mpg)
33	17.461218	10.5.30.1	10.5.10.1	HTTP	Continuation or non-HTTP traffic
34	17.462085	10.5.10.1	10.5.30.1	TCP	2329 > www [ACK] Seq=195 Ack=1449 Win=8736 Len=0 TSV=2293212 TSER=2290537
35	17.462196	10.5.30.1	10.5.10.1	HTTP	Continuation or non-HTTP traffic
36	17.462277	10.5.30.1	10.5.10.1	HTTP	Continuation or non-HTTP traffic
37	17.462342	10.5.10.1	10.5.30.1	TCP	2329 > www [ACK] Seq=195 Ack=2387 Win=11632 Len=0 TSV=2293212 TSER=2290537
38	17.462315	10.5.30.1	10.5.10.1	HTTP	Continuation or non-HTTP traffic
39	17.462471	10.5.30.1	10.5.10.1	HTTP	Continuation or non-HTTP traffic
40	17.463024	10.5.10.1	10.5.30.1	TCP	2329 > www [ACK] Seq=195 Ack=4345 Win=14528 Len=0 TSV=2293212 TSER=2290537
41	17.463103	10.5.30.1	10.5.10.1	HTTP	Continuation or non-HTTP traffic
42	17.463174	10.5.30.1	10.5.10.1	HTTP	Continuation or non-HTTP traffic
43	17.463237	10.5.10.1	10.5.30.1	TCP	2329 > www [ACK] Seq=195 Ack=5793 Win=17424 Len=0 TSV=2293212 TSER=2290537
44	17.463310	10.5.10.1	10.5.30.1	TCP	2329 > www [ACK] Seq=195 Ack=7241 Win=20320 Len=0 TSV=2293212 TSER=2290537

Abbildung 4: Verbindungsaufbau bei Video-on-Demand

Wie man im Screenshot (Bild 4) sieht, wird eine TCP Verbindung aufgebaut und mit einer gewöhnlichen HTTP-GET Anfrage das Video „MATRIX-de-ultra.mpg“ (Paket 29) angefordert.

Nun analysieren wir den Paketstrom bei pausiertem Videoplayer am Client:

566	26.645949	10.5.30.1	10.5.10.1	HTTP	Continuation or non-HTTP traffic
567	26.685798	10.5.10.1	10.5.30.1	TCP	2331 > www [ACK] Seq=200 Ack=1275711 Win=544 Len=0 TSV=2295518 TSER=2292833
568	26.925364	10.5.30.1	10.5.10.1	HTTP	[TCP Window Full] Continuation or non-HTTP traffic
569	26.925991	10.5.10.1	10.5.30.1	TCP	[TCP ZeroWindow] 2331 > www [ACK] Seq=200 Ack=1276255 Win=0 Len=0 TSV=2295578 TSER=2292903
570	27.157369	10.5.30.1	10.5.10.1	TCP	[TCP Keep-Alive] www > 2331 [ACK] Seq=1276254 Ack=300 Win=6864 Len=0 TSV=2292301 TSER=2295578
571	27.157768	10.5.10.1	10.5.30.1	TCP	[TCP ZeroWindow] 2331 > www [ACK] Seq=200 Ack=1276255 Win=0 Len=0 TSV=2295635 TSER=2292903
572	27.621375	10.5.30.1	10.5.10.1	TCP	[TCP Keep-Alive] www > 2331 [ACK] Seq=1276254 Ack=200 Win=6864 Len=0 TSV=2293077 TSER=2295635
573	27.621810	10.5.10.1	10.5.30.1	TCP	[TCP ZeroWindow] 2331 > www [ACK] Seq=200 Ack=1276255 Win=0 Len=0 TSV=2295751 TSER=2292903
574	28.549449	10.5.30.1	10.5.10.1	TCP	[TCP Keep-Alive] www > 2331 [ACK] Seq=1276254 Ack=200 Win=6864 Len=0 TSV=2293309 TSER=2295751

Abbildung 5: Paketverkehr bei pausiertem Video

Hier ist sind die Pakete ab 1669 interessant:

- Paket 1669 ist vom Client an den Server, der Client teilt dem Server plötzlich eine Windowsize von „0“ mit. Er täuscht dem Server also vor, dass sein Pufferspeicher derzeit voll ist und er keine weiteren Pakete annehmen kann.
- Die Pakete 1770, 1772 und 1774 des Servers sind daraufhin immer nur noch „Keep-Alive“ Pakete, damit die Verbindung bestehen bleibt. Wie man an diesen Paketen sieht, ändert sich auch die Sequenznummer nicht, der Server versendet also keinerlei Nutzdaten.