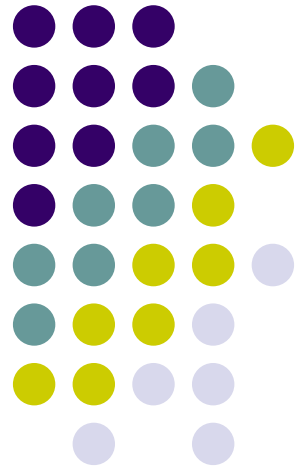
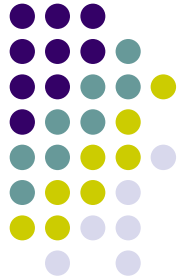
asmping

Stig Venaas

sv@ecs.soton.ac.uk

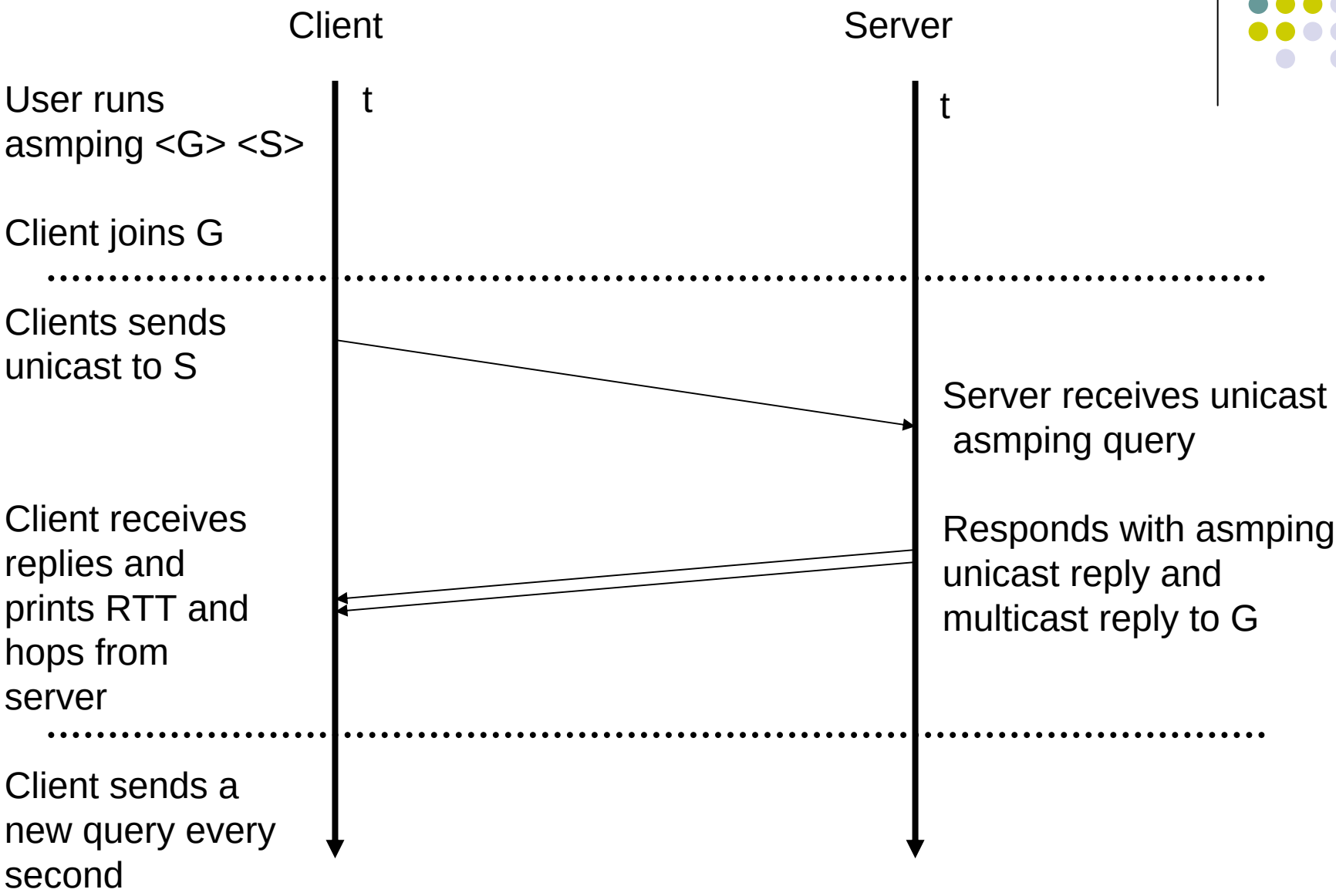


What is asmping?

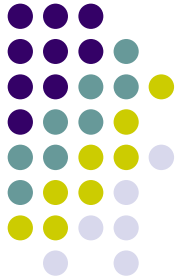


- A tool for testing multicast connectivity
- Behavior is a bit like normal ping
- A server must run ssmpingd (latest version supports asmping)
- asmping is ASM version of ssmping
- A client can ping a server by sending unicast asmping query
- Server replies with both unicast and multicast asmping replies
- In this way a client can check that it receives ASM from the server
  - And also parameters like delay, number of router hops etc.

How it works

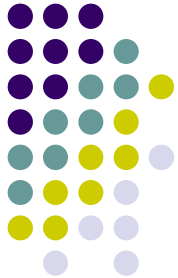

Client
Server
t
t
User runs asmping <G> <S>
Client joins G
Clients sends unicast to S
Server receives unicast asmping query
Client receives replies and prints RTT and hops from server
Responds with asmping unicast reply and multicast reply to G
Client sends a new query every second

Example output



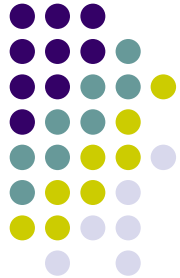
```
[venaas@storhaugen ~]$ asmping -c 5 224.1.2.234 ssmping.net.switch.ch
ssmping joined (S,G) = (130.59.35.130,224.1.2.234)
pinging S from 158.38.63.22
  unicast from 130.59.35.130, seq=1 dist=11 time=66.203 ms
  unicast from 130.59.35.130, seq=2 dist=11 time=66.042 ms
multicast from 130.59.35.130, seq=2 dist=11 time=66.492 ms
  unicast from 130.59.35.130, seq=3 dist=11 time=66.515 ms
multicast from 130.59.35.130, seq=3 dist=11 time=66.520 ms
  unicast from 130.59.35.130, seq=4 dist=11 time=66.316 ms
multicast from 130.59.35.130, seq=4 dist=11 time=66.321 ms
  unicast from 130.59.35.130, seq=5 dist=11 time=66.407 ms
multicast from 130.59.35.130, seq=5 dist=11 time=66.956 ms

--- 130.59.35.130 ssmping statistics ---
5 packets transmitted, time 5000 ms
unicast:
   5 packets received, 0% packet loss
   rtt min/avg/max/std-dev = 66.042/66.296/66.515/0.326 ms
multicast:
   4 packets received, 0% packet loss since first mc packet (seq 2) recvd
   rtt min/avg/max/std-dev = 66.321/66.572/66.956/0.296 ms
```

More interesting example



```
sv@xiang /tmp $ asmping 224.3.4.234 ssmping.uninett.no
ssmping joined (S,G) = (158.38.63.22,224.3.4.234)
pinging S from 152.78.64.13
  unicast from 158.38.63.22, seq=1 dist=23 time=57.261 ms
  unicast from 158.38.63.22, seq=2 dist=23 time=56.032 ms
multicast from 158.38.63.22, seq=2 dist=7 time=207.876 ms
multicast from 158.38.63.22, seq=2 dist=7 time=208.567 ms (DUP!)
  unicast from 158.38.63.22, seq=3 dist=23 time=56.852 ms
multicast from 158.38.63.22, seq=3 dist=21 time=70.352 ms
multicast from 158.38.63.22, seq=4 dist=21 time=57.208 ms
  unicast from 158.38.63.22, seq=4 dist=23 time=57.910 ms
  unicast from 158.38.63.22, seq=5 dist=23 time=56.206 ms
multicast from 158.38.63.22, seq=5 dist=21 time=57.375 ms
```

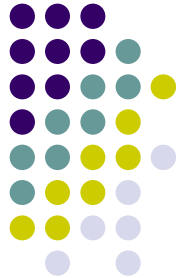
What does output tell us?



- 23 unicast hops from source
- For multicast we first have 7, later stays at 21
- We also get some info about loss and RTTs

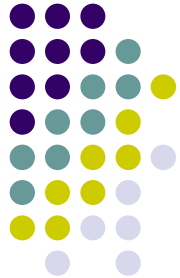
- Number of hops is perhaps the most interesting
- In the stable situation with 21 there is probably native forwarding all they way
- Forwarding is probably on shortest path tree from source to receiver
- In theory we might also detect switch from RPT to SPT if number of hops are different
  - Number of hops on RPT are then probably larger than number of unicast hops
- But why only 7 hops initially?
- Why did we get the packet twice, and why do both have large ttl?

The mystery of the 7 hops



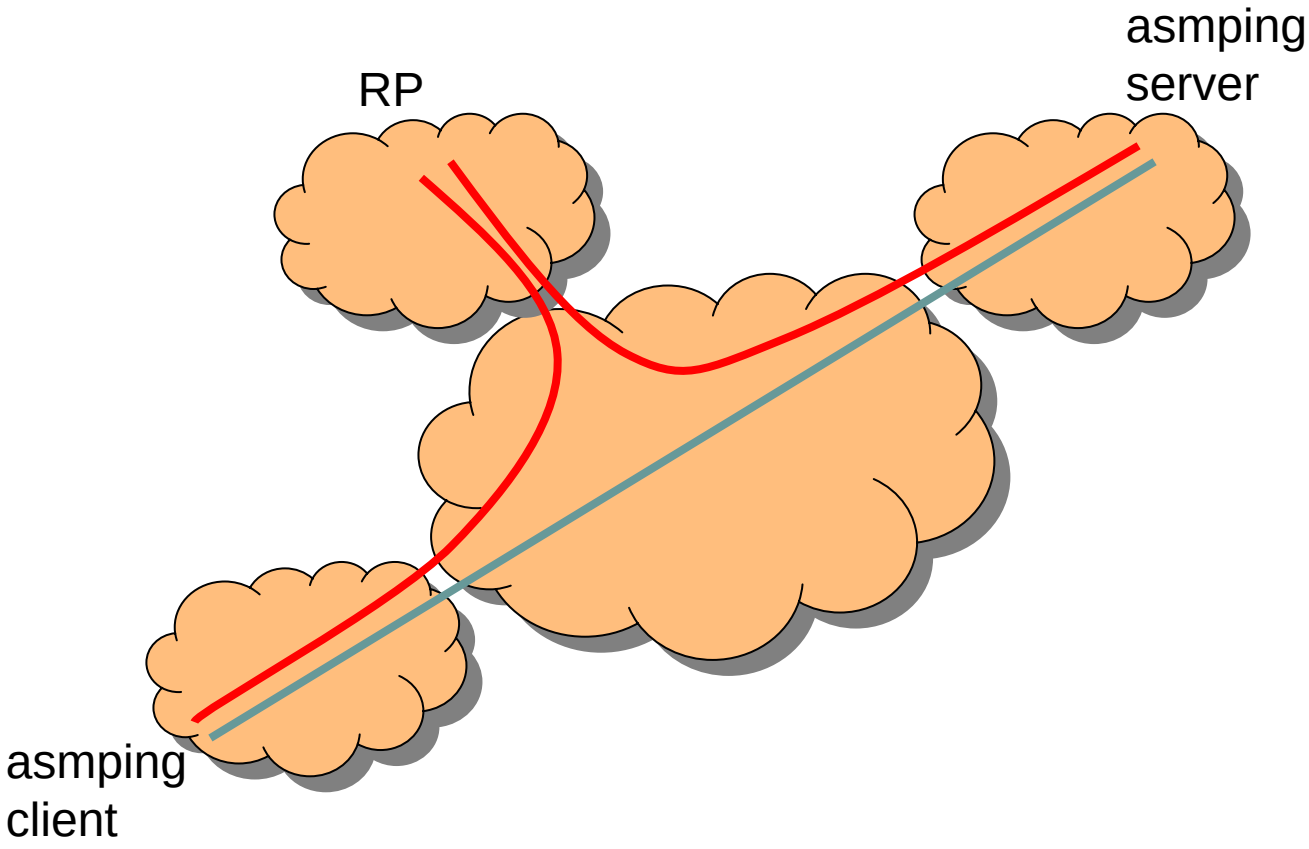
- So why only 7, and why large RTT and duplicate?
- The reason is MSDP and PIM registers
- In this case we know for sure that packets must have been encapsulated in MSDP. We are more than 7 hops away from UNINETT where the RP is.
- Assuming there are no other listeners, the packet was first encapsulated in PIM register going to the local RP. Then it was encapsulated in MSDP all the way to our local RP
  - Not sure exactly when ttl is decremented with regard to register and MSDP, but in this case the receiver is 4-6 hops from the local RP
- So that explains the number of hops, but why large RTT and duplicate?

Large RTT and duplicate



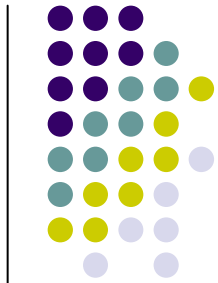
- So why the large RTT and duplicate?
- The reason is again MSDP
  - At least that's the theory
- The register packets are in this case passed with MSDP to the RP, and probably cached there
- When our (*,G) reaches our RP, we receive the packet from the cache, which by now is 200ms old
- Next, our last-hop router switches to SPT, and what we think happens, is that the (S,G)-join also reaches the RP (RP is on the SPT path), and the RP again forwards it's copy when this happens
- This also gives us a measure of how long the RPT to SPT switch did take, if we are correct in our speculations

Embedded-RP



- With embedded-RP, client chooses server but also RP
- Can verify that source in one domain can register to RP in another domain, that client can receive via the RP, and also see the switches

Summary



- It might be used to simply check connectivity, but also gives extra info like RTT and hops
- Compared to ssmping we might be able to derive some more interesting information due to the complexities of PIM register, MSDP and tree switching
- It supports both IPv4 and IPv6, and might have some interesting uses for testing IPv6 embedded-RP
- Both asmping and ssmpingd should work on most systems, no SSM support needed
- See http://www.venaas.no/multicast/ssmping/ for more info