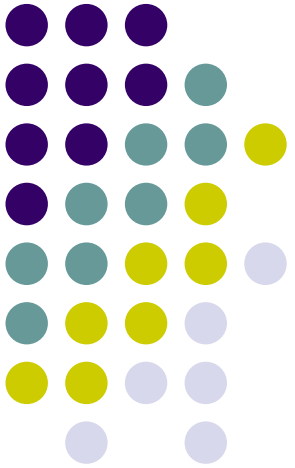
ssmping

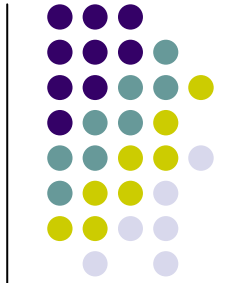
No draft yet, sorry

Stig Venaas

venaas@uninett.no

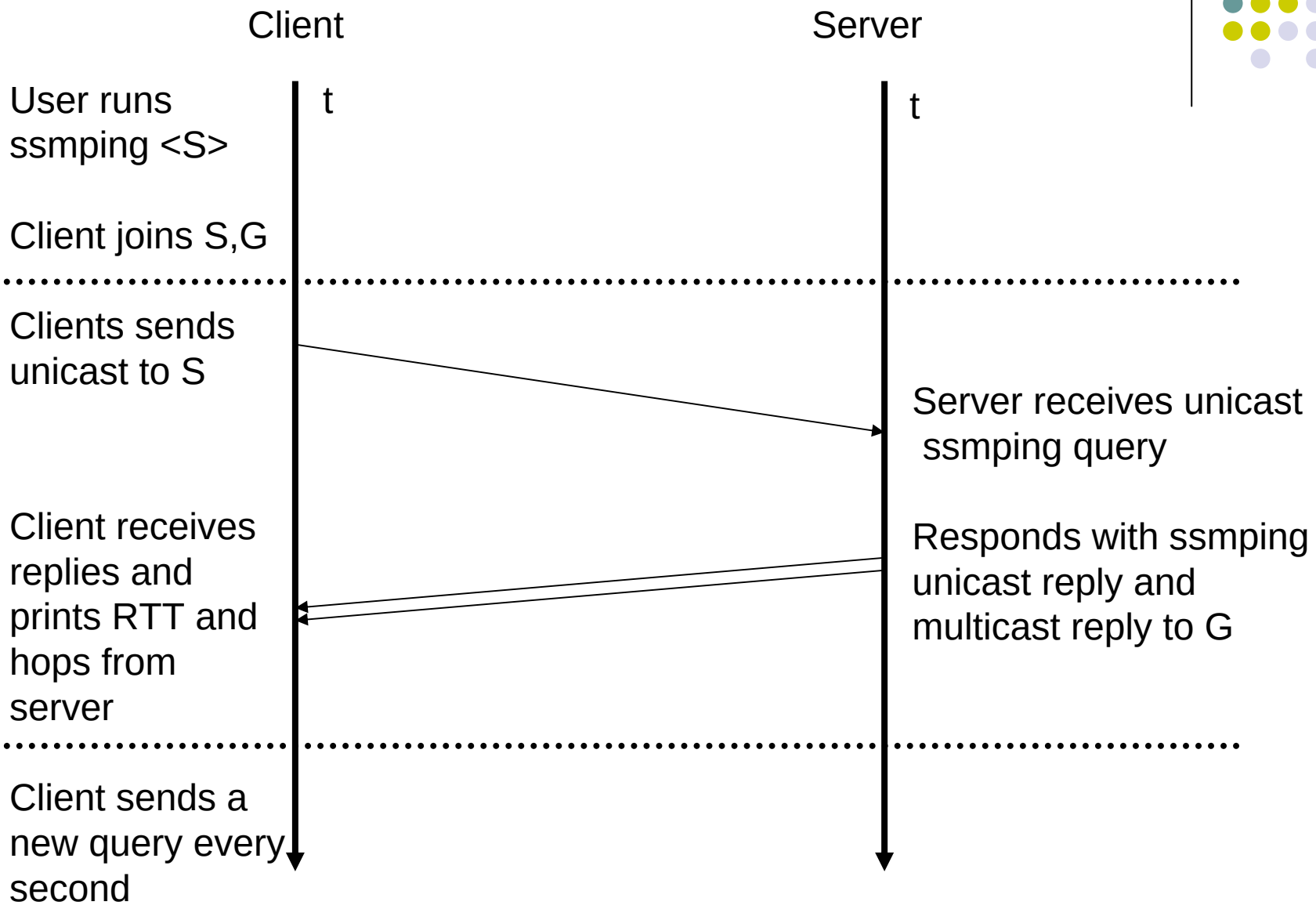


ssmping

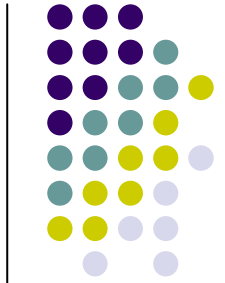


- A tool for testing multicast connectivity and more
- Behaviour is a bit like normal icmp ping
- Implemented at application layer using UDP
  - No additional requirements on the operating system
  - The operating system and network must support SSM
- A server must run ssmpingd
- A client pings server by sending unicast ssmping query
- The server replies with both unicast and multicast ssmping replies
- In this way a client can check that it receives SSM from the server
  - You can run your own server, also several public IPv4 and IPv6 servers on the Internet
  - And also parameters like delay, number of router hops etc.

How it works

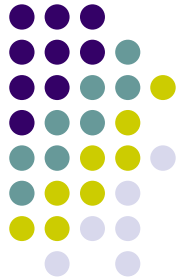

Client
Server
User runs ssmping <S>
t
t
Client joins S,G
Clients sends unicast to S
Server receives unicast ssmping query
Client receives replies and prints RTT and hops from server
Responds with ssmping unicast reply and multicast reply to G
Client sends a new query every second

Example output



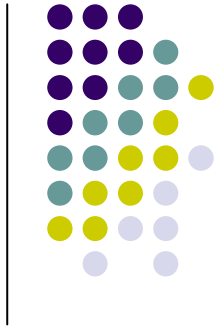
```
$ ssmping -c 5 -4 flo.nrc.ca
ssmping joined (S,G) = (132.246.2.20,232.43.211.234)
pinging S from 158.38.63.20
  unicast from 132.246.2.20, seq=1 dist=13 time=122.098 ms
  unicast from 132.246.2.20, seq=2 dist=13 time=122.314 ms
multicast from 132.246.2.20, seq=2 dist=13 time=125.061 ms
  unicast from 132.246.2.20, seq=3 dist=13 time=122.327 ms
multicast from 132.246.2.20, seq=3 dist=13 time=122.345 ms
  unicast from 132.246.2.20, seq=4 dist=13 time=122.334 ms
multicast from 132.246.2.20, seq=4 dist=13 time=122.371 ms
  unicast from 132.246.2.20, seq=5 dist=13 time=122.360 ms
multicast from 132.246.2.20, seq=5 dist=13 time=122.384 ms

--- 132.246.2.20 ssmping statistics ---
5 packets transmitted, time 5003 ms
unicast:
   5 packets received, 0% packet loss
   rtt min/avg/max/std-dev = 122.098/122.286/122.360/0.394 ms
multicast:
   4 packets received, 0% packet loss since first mc packet (seq 2)
   recvd
   rtt min/avg/max/std-dev = 122.345/123.040/125.061/1.192 ms
```

What does the output tell us?



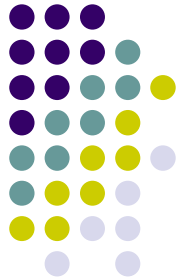
- 13 unicast hops from source, also 13 for multicast
- Multicast RTTs are slightly larger and vary more
  - The difference in unicast and multicast RTT shows one way difference for unicast and multicast replies, since they are replies to the same request packet
- The multicast tree is not ready for first multicast reply, ok for 2nd
- No unicast loss, no multicast loss after tree established

Is it useful?



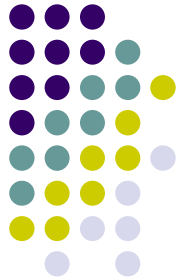
- Complements multicast beacons
- Useful for “end users” or others that want to perform a “one-shot” test rather than continuously running a beacon
- Beacons don’t show how long it takes to establish the multicast tree, they only show the “steady state”
  - We’ve seen cases where it takes much longer than expected
- Neither do they compare unicast and multicast
- Are there other data than RTT and hops that should be measured?
  - Hops are measured by always using a ttl/hop count of 64 when sending replies

History



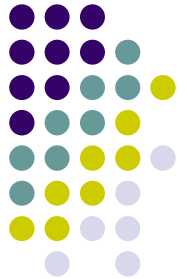
- Based on an idea by Pavan Namburi, Kamil Sarac University of Texas and Kevin C. Almeroth UCSB
  - http://www.utdallas.edu/~ksarac/research/publications/draft-sarac-mping-00.txt
  - http://www.utdallas.edu/~ksarac/research/publications/CIIT 04-1.pdf
- Their idea involves extending IGMP/MLD
  - Presented IETF 58 mboned meeting, not much interest, I believe it was suggested to just use UDP
- This does some of the same, but doesn’t require network support
  - Only uses UDP
  - But it requires server to run ssmpingd

Summary



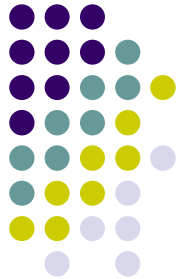
- Tool and further documentation available from http://www.venaas.no/multicast/ssmping/
- You can deploy your own server, or check that you can receive from the public servers listed at the above URL
- Supports both IPv4 and IPv6
- Currently it works for Linux, Solaris, Windows XP/Vista and some BSD systems
  - Note that ssmping client requires SSM support

Also asmping. Example output:



```
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
```

Next steps



- Want to reserve port number and/or SRV name
  - Tool is getting pretty popular, so think about time
- Might be useful to reserve IPv4/IPv6 multicast address
- Some of this requires an RFC
- Specifying the protocol also allows other implementations
  - Currently one other independent server implementation
- I will submit draft well before the next meeting
  - Unfortunately got too late for this one
- Input on protocol welcome
  - Preventing misuse/DoS
  - Make sure it’s flexible/extendible