



Assignment-1:

A Simple Multicast Tunnel (SMT)



Goals of the Assignment

- ▶ To gain practical experience of Network Programming
- ▶ To better the understanding of network protocols
- ▶ To better the understanding of implementation details and issues associated with it



Overview of the SMT

- ▶ A Sample Scenario:
 - Assume, you are in a network which can receive Multicast packets
 - Some of your friends are interested in receiving those Multicast packets, but they are in a different network which does not receive the Multicast data
 - your friend's connection can be behind a NAT
 - your friend's cannot receive UDP
 - Need to find a way to forward the packets to them
- ▶ Using SMT program, you can solve the above issues
 - SMT is a simple mechanism for the assignment purpose (but may not be the best mechanism)

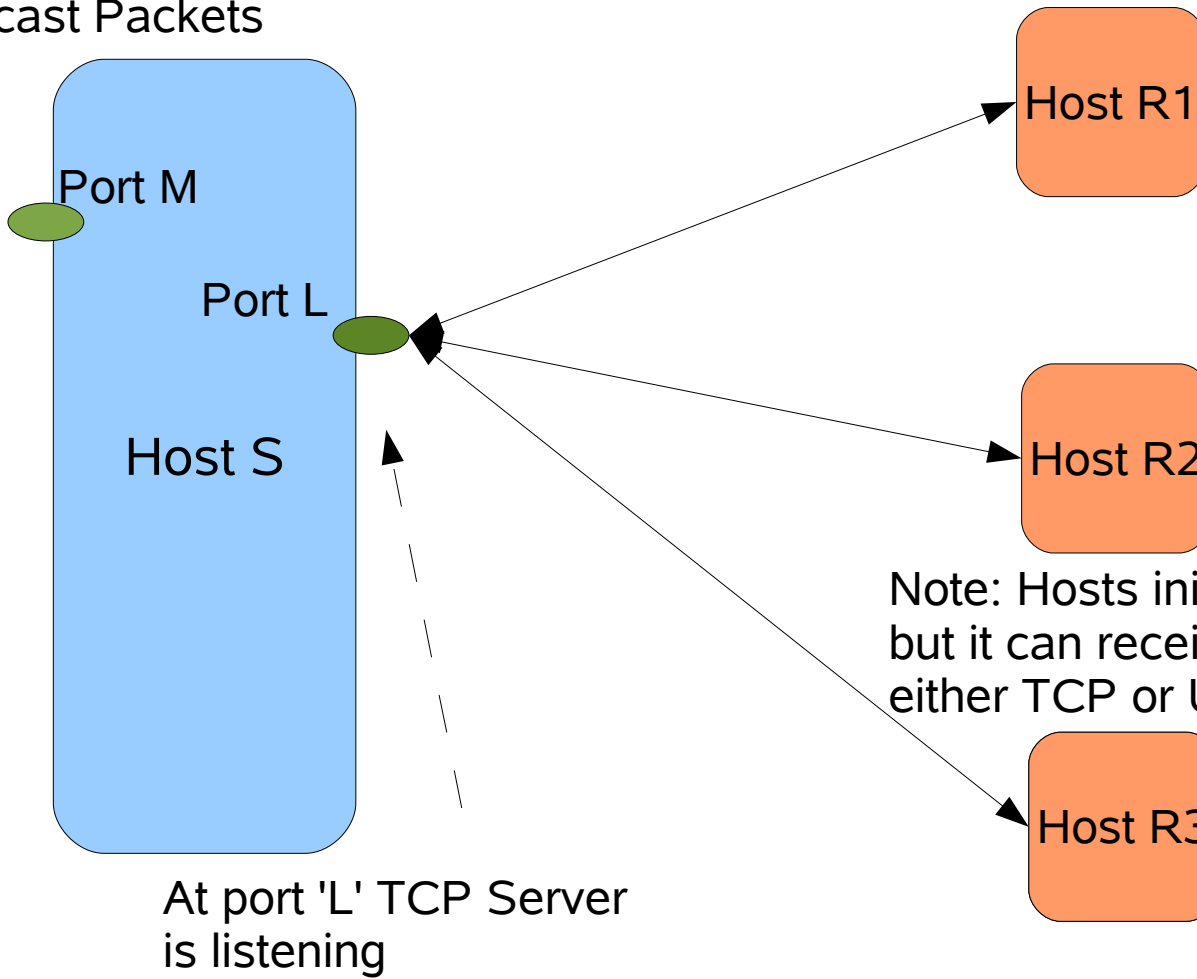


Tasks of SMT program

- ▶ Receive multicast packets from a specified port
- ▶ Listen on a specified port for TCP connections
 - Hosts(your friends :)) interested in the multicast packets shall connect to this port
 - The connected host provides two important data
 - Whether it intends to receive via TCP or UDP
 - if UDP, then the Target address and port number is also specified
 - if TCP, then the multicast packets are forwarded in the same connection itself

Pictorial Overview of SMT

Port 'M' is receiving Multicast Packets



Note: Hosts initially connects via TCP, but it can receive the multicast data via either TCP or UDP



Message sent by connecting client to SMT

- ▶ Message Example:1 (To receive via TCP)

TCP\r\n

\r\n

- ▶ Message Example:1 (To receive via UDP)

UDP\r\n

130.233.x.y\r\n

5678\r\n

\r\n

- ▶ You are also free to choose other message encoding mechanism (you could try something better :))



Forwarding Multicast Packets

- ▶ **if UDP** was requested by the connected client
 - Repack the multicast datagram into a unicast datagram (Target Address is specified by the connecting client)
- ▶ **if TCP**,
 - There is an important repacking issue
 - Datagram to Streams (TCP does not preserve packet boundaries)
 - **Important: Add a two byte header to every datagram, before forwarding via TCP**
 - The two byte header is used by the connected client to reconstruct the datagram, before forwarding to any of its application



SMT: Required Command Line Interface

- ▶ Multicast address and port number
 - from where packets are received
- ▶ Local address and port number
 - where new TCP connections are listened for
- ▶ Duration of execution of program
 - `./smt -m <M_IP:M_PORT>`**
 - `-l <L_IP:L_PORT>`** or **`<LocalHostName:Port>`**
 - `-d <duration in seconds>`**
- ▶ Ex:1 `./smt -m 239.255.255.255:5678 -l 130.233.x.y:3456 -d 100`
- ▶ Ex2: `./smt -m 239.255.255.255:5678 -l xyz.hut.fi:3456 -d 100`



Test Client Program

- ▶ The test clients(who intend to receive the stream) need to be prepared by yourself(it is simple and needed for debugging)
 - Command line interface required for test clients
 - -d Target IP Address and port number to connect to
 - -t receive packets via TCP
 - -u receive packets via UDP
 - -l Local IP address and port number (only with -u option)
 - -m Log short information about received packet to stdout
 - Format: next slide
 - Ex:1 `./testClient -d 130.233.x.y:4567 -t -m`
 - Ex:2 `./testClient -d 130.233.x.y:4567 -u -l 130.23.y.z:2345 -m`



Test Client Program contd..2

- ▶ Regarding Log/Monitor format

**<Reception Timestamp> <From IP : Port> <Received at IP: Port>
<Packet size in bytes>**

- ▶ Example:

● 23.456 130.233.1.4:5000 130.233.1.5:5004 512 bytes

- ▶ Other Log Data, that you feel necessary can be added too :)



Deliverables for this assignment

- ▶ Working Implementation (Should be testable in Maarintalo's Unix machines)
- ▶ SMT source code
- ▶ Test Client source code
- ▶ A simple readMe file, on compilation and execution instructions
- ▶ A brief comment about your assignment (max: 1 page)
 - Implementation issues faced
 - Comments/Suggestions if any
 - Extra features if any
 - Anything that you would like to tell us



Others:

- ▶ A simple MULTICAST sender would be provided through the course web page (that can be run Maari machines)
- ▶ Many networks disable Multicast packets
 - FYI: Maari-A machines seem to allow Multicast packets (atleast within Maari-A machines)
- ▶ DEADLINE: Date would be put in the web in a couple of days