Computer Network Course Project Part 1

Zhiyong Fang Q1826275261@SJTU.EDU.CN
Shanghai Jiao Tong University

Abstract

I code tools for 'ping' and 'iperf' in Java, test them in mininet environment, and get correct results compared with the 'delay' and 'bandwidth' predefined in the network. Furthermore, I implement the control plane and data plane of a router, and I can 'ping' and 'wget' from hosts to hosts and routers successfully. Techniques such as ARP looking up, ICMP packet replying, route table management are included. I make effort to adapt mininet and pox to the newest version by reading and modifying many source codes.

1. Pinger and Iperfer

1.1. Pinger

The pinger I implemented takes advantage of UDP packet transfer and reply. The client continously sends UDP packets and waits for reply from the server. A 'time out' value is properly set so that the client will not wait for a lost packet forever. Noticeably, the client should reject packets with sequence number different from the one it is waiting for, otherwise, the statistics will not be correct. This phenomenon will not occur in the test environment, but it should be considered in real world application.

1.2. Iperfer

The Iperfer utilizes the TCP packet transfer. The client continuously sends TCP packet during a period of time, and uses the amount of sent data to estimate the bandwidth. The server do the similar thing, but use the first received packet to indicate the start of transmission.

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

2. Measurement

2.1. Results

Pinger and Iperfer are used to test the latency and throughput of the topology in Figure 1, and the result is shown in Table 1.

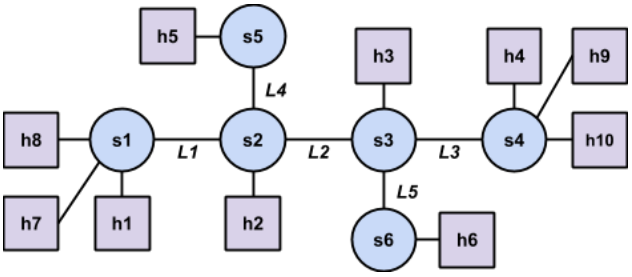

Figure 1: The topology to test Pinger and Iperfer

Table 1. Measurement of latency and throughput.

PATH	RTT (MS)	THROUGHPUT (MBPS)
LINK1	81.80	20.19
LINK2	21.45	40.79
LINK3	82.50	20.19
LINK4	11.60	20.41
LINK5	11.75	20.39
H1 - H4	182.55	19.95
H5 - H6	42.75	20.30

2.2. Analysis

Path Latency and Throughput A simple conclusion comes to us that, the latency of a path equals to the summation of all link latency of that path, and the overall throughput equals to the smallest link throughput in the path.

Simultanous Communication When two pairs of hosts connects to s1 and s4 ping each other at the same time, the average latency are 183.30 ms and 183.40 ms, respectively. The result is only slightly larger than the 182.55 ms result in Table 1 due to the effect of multiplexing. However, as for iperf, the throughput

becomes 13.68 Mbps and 8.33 Mbps, whose summation is around 20 Mbps as they share the link.

2.3. Discussion

The latency is slightly greater than standard values because the first ping operation have a relatively large latency. I suppose that this comes from the first arp looking up in the routers. After that, the router can use the cache, and becomes faster. It is worth noticing that I report the throughput from the server side, which is different from pinger. This is because the the iperf client does not ensure all data is received when it ends, so the amount of sent data is actually over estimated and not correct.

3. Router Data Plane

Data plane can be divided into two parts, forwarding and responding, which is separated by whether the packet is destined for one of the routers' own interfaces. In this part, routers read static route table from files and does no update.

3.1. Forwarding

Routers look up destination IP and mask IP in the route Table to decide next-hop IP, and look up arp cache to get next-hop mac address. The packet will wait if no result in arp cache is found, and be dropped if it waits for too long.

3.2. Responding

Routers will respond to ICMP(type: 8, code :0), i.e. ping message, with ICMP(type:0, code 0) and all payload from the original packet. It will generate all other ICMP messages with specific type and code corresponding to the problems such as time out, destination unreachable, and port unreachable, etc..

3.3. Overall results

As my router is running, pingall command in mininet can get through and wget can download index.html from the server. The topology is the single.topo.

4. Router Control Plane

Control plane basically manage route tables. At the launch of a router, it will send RIP requests among all interfaces. It will update its route table according to RIP packets from other routers, time out old ones, and also broadcast its own route table.

4.1. Longest Prefix Match

I implement the longest prefix match other than the common look up. This will help forwarding IP only with subnet information.

4.2. Distance Vector Routing

Though the course work does not ask us to do so, I implement the distant vector routing method. Route table entry is updated if a shorter path is found. This is done by adding a field 'metric' in class RouteTableEntry, and another field 'timeAdded' is added as well to support time out operation.

4.3. Split Horizon Broadcast

Routers will check their route table entry to ensure that no entry will be broadcast to the interface which serves as its next-hop entry. This alleviates the count-to-infinity problem.

4.4. Overall results

Pingall and wget can both get through in topology pair.topo and triangle.topo after 10 seconds from launching. (Since the first launched ones have to wait the latter to broadcast their route table.)

5. Conclusion

In all, I finish:

1. Pinger and Iperfer and their measurement.
2. Router data plane including IP forwarding and ICMP responding.
3. Router control plane including 'longest prefix match', 'distance vector routing' and 'split horizon broadcast'.