ECE 697J – Advanced Topics in Computer Networks

Network Measurement

12/02/03

Overview

- Lab 3 requires performance measurement
  - Throughput
  - Collecting of packet headers
- Network Measurement
  - Active measurement
  - Tools
  - Passive measurement
  - Anonymization of data

Network Measurements

- Why do we need measurements?
  - Debugging
  - Performance tuning
  - Discovery of network structure
  - Understanding of network behavior (reverse-engineering)
  - Discovery of security holes and attacks
  - Etc.
- How can we measure networks?
  - Inject packets and see what happens (active measurement)
  - Observe traffic (passive measurement)
- What are pros and cons of measurement?

Active Measurement

- Metrics that can be measured
  - Connectivity
  - Round-trip time
  - Loss rate
  - Reordering
  - Available bandwidth
  - Bandwidth capacity
- Some metrics are available per-hop, others only end-to-end
- Some tools need software on both sides of measurement

Connectivity

- Simples case of active measurement
- Typically done with ICMP Echo Request
  - Recipient will reply with ICMP Echo Response
- Implemented in *ping* tool:
  - Sends ICMP echo requests to specified IP address
  - Prints responses
  - Reports TTL, round-trip time, loss rate (both ways)
- Useful parameters
  - -c or -n count
  - On Unix: -n numeric output (no IP address translation)
  - -f flood ping ☺
- Very common and useful tool

Ping

```
wolf@Legolas ~
$ ping www.umass.edu

Pinging gemini.oit.umass.edu [128.119.166.100] with 32 bytes of data:

Reply from 128.119.166.100: bytes=32 time=2ms TTL=252
Reply from 128.119.166.100: bytes=32 time=3ms TTL=252
Reply from 128.119.166.100: bytes=32 time=2ms TTL=252
Reply from 128.119.166.100: bytes=32 time=3ms TTL=252

Ping statistics for 128.119.166.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 3ms, Average = 2ms

wolf@Legolas ~
$
```

Ping Limitations

- What are the limitations of ping?
  - ICMP disabled
  - NAT boxes / firewalls
  - No information on route (other than TTL)
  - No information on performance (other than RTT)
- Other interesting observations
  - TTL in packets can reveal OS type (useful for hackers)

Route

- How can route of packet be measured?
- *traceroute* approach:
  - Send packets with limited TTL towards destination
  - Packets will "expire" and cause ICMP error message
  - Source of error message is intermediate hop
  - Repeat with increasing TTL
- Output:
  - Each router with RTT

traceroute

```
wolf@Legolas ~
$ tracert www.umass.edu

Tracing route to gemini.oit.umass.edu [128.119.166.100]
over a maximum of 30 hops:

  1     *        *        *     Request timed out.
  2     2 ms     3 ms     1 ms  know-rt-04-1.gw.umass.edu [128.119.91.254]
  3     5 ms     3 ms     5 ms  lgrc-rt-106-8.gw.umass.edu [128.119.2.238]
  4     2 ms     2 ms     4 ms  gemini.oit.umass.edu [128.119.166.100]

Trace complete.

wolf@Legolas ~
$
```

traceroute Limitations

- What are the limitations of traceroute?
  - Not all routers respond
  - Route asymmetry leads to wrong TTL results
  - Data path vs. control path processing leads to wrong TTL results

Hop	%Loss	IP Address	ms	Graph
0		192.168.0.100		0 … 129
1	100			
2		10.221.248.1	20	
3		24.91.0.89	10	
4		[illegible]	12	
5		[illegible]	11	
6		[illegible]	16	
7		[illegible]	16	
8		[illegible]	16	
9		[illegible]	18	
10		[illegible]	20	
11		12.122.10.22	30	
12		12.123.1.125	21	
13		192.205.32.242	22	
14		129.250.2.217	23	
15		129.250.4.198	44	
16		129.250.5.71	43	
17		129.250.4.45	53	
18		129.250.30.76	57	
19		199.217.166.226	56	
20		128.252.1.201	59	
21		**128.252.173.244**	57	

Roundtrip time to www.wustl.edu, average = 57ms, min = 55ms, max = 91ms --

Bandwidth

- How to measure bandwidth?
  - TCP vs. UDP
  - Inject packets at high rates
  - Reporting of result?
  - Requires software on both sides
- Issues to consider
  - Measurement reports currently available bandwidth
  - Reports only bottleneck bandwidth
  - TCP behavior needs to be considered
  - Timing of UDP packet is critical
- Tool: *iperf* (and many others)
  - Client acts as sender
  - Server sinks traffic and reports statistics

iperf

- iperf report:

```
------------------------------------------------------------
Client connecting to 192.168.1.2, TCP port 9044
TCP window size: 8.00 KByte (default)
------------------------------------------------------------
[  3] local 128.1.1.2 port 3930 connected with 192.168.1.2 port
  9044
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-212.8 sec  94.6 MBytes  3.73 Mbits/sec
```

- iperf options
  - -s run as server
  - -c run as client
  - -u uses UDP instead of TCP
  - Man other options for packet size and rate (UDP)
  - -b binds output interface (very useful)

iperf Limitations

- What are the limitations of iperf?
  - Same as for any other bandwidth measurement tool
  - Control overhead
  - Many options -> possible misconfiguration
- Need tool to observe network traffic to verify correct measurement setup

tcpdump

- Passive network measurement tool: *tcpdump*
- tcpdump collects packets from interface and displays headers
  - Only one interface can be observed at any point of time
  - All traffic on interface can bee seen (promiscuous mode)
  - Filter allows pre-filtering of output
  - Payload can be preserved (if necessary)
  - Timestamp of packet arrival and transmission
- Very useful to check network setup
- Useful options
  - -n no address translation
  - -r and -w to read and write files
  - -s determines length of preserved data
  - -vv very verbose output
- Results can be displayed nicely with *ethereal*

tcpdump

```
wolf@Legolas ~
$ windump -i 2 -n -v -c 8
c:\WINDOWS\system32\windump.exe: listening on \Device\NPF_{6F9E9E7E-1D1F-4B94-AF
34-56F42279AAD9}
12:26:57.669558 IP (tos 0x0, ttl 42, id 25958, len 40) 210.120.247.66.6210 > 192
.168.0.100.3568: . [tcp sum ok] ack 167328060 win 5840 (DF)
12:26:57.669611 IP (tos 0x0, ttl 128, id 55511, len 40) 192.168.0.100.3568 > 210
.120.247.66.6210: . [tcp sum ok] ack 1 win 0 (DF)
12:27:00.698418 IP (tos 0x0, ttl 128, id 55516, len 714) 192.168.0.100.3611 > 21
6.239.41.99.80: P 761568430:761569104(674) ack 3906399445 win 64235 (DF)
12:27:00.713904 IP (tos 0x0, ttl 51, id 48342, len 40) 216.239.41.99.80 > 192.16
8.0.100.3611: . [tcp sum ok] ack 674 win 30660 (DF)
12:27:00.730965 IP (tos 0x0, ttl 51, id 49927, len 1464) 216.239.41.99.80 > 192.
168.0.100.3611: P 1:1425(1424) ack 674 win 32120 (DF)
12:27:00.731032 IP (tos 0x0, ttl 51, id 50023, len 176) 216.239.41.99.80 > 192.1
68.0.100.3611: P 2885:3021(136) ack 674 win 32120 (DF)
12:27:00.731062 IP (tos 0x0, ttl 128, id 55517, len 40) 192.168.0.100.3611 > 216
.239.41.99.80: . [tcp sum ok] ack 1425 win 62811 (DF)
12:27:00.732684 IP (tos 0x0, ttl 51, id 50022, len 1500) 216.239.41.99.80 > 192.
168.0.100.3611: P 1425:2885(1460) ack 674 win 32120 (DF)
9 packets received by filter
0 packets dropped by kernel

wolf@Legolas ~
$
```

Bonus Questions

- How can you measure bandwidth capacity of a link?
- How can you measure the delay incurred by a single router?

Passive Measurement

- tcpdump is an example of passive network measurement
- Passive measurement consists of several phases
  - Data collection
  - Data storage
  - Extraction and calculation of metrics
- Passive measurement metrics
  - Traffic volume (link utilization)
  - Traffic mix (e.g., by protocol type, by destination)
  - TCP flow behavior (packet retransmissions)
- Passive measurement challenges?
  - Data rates to process
  - Only partial view of network
  - Staleness of data

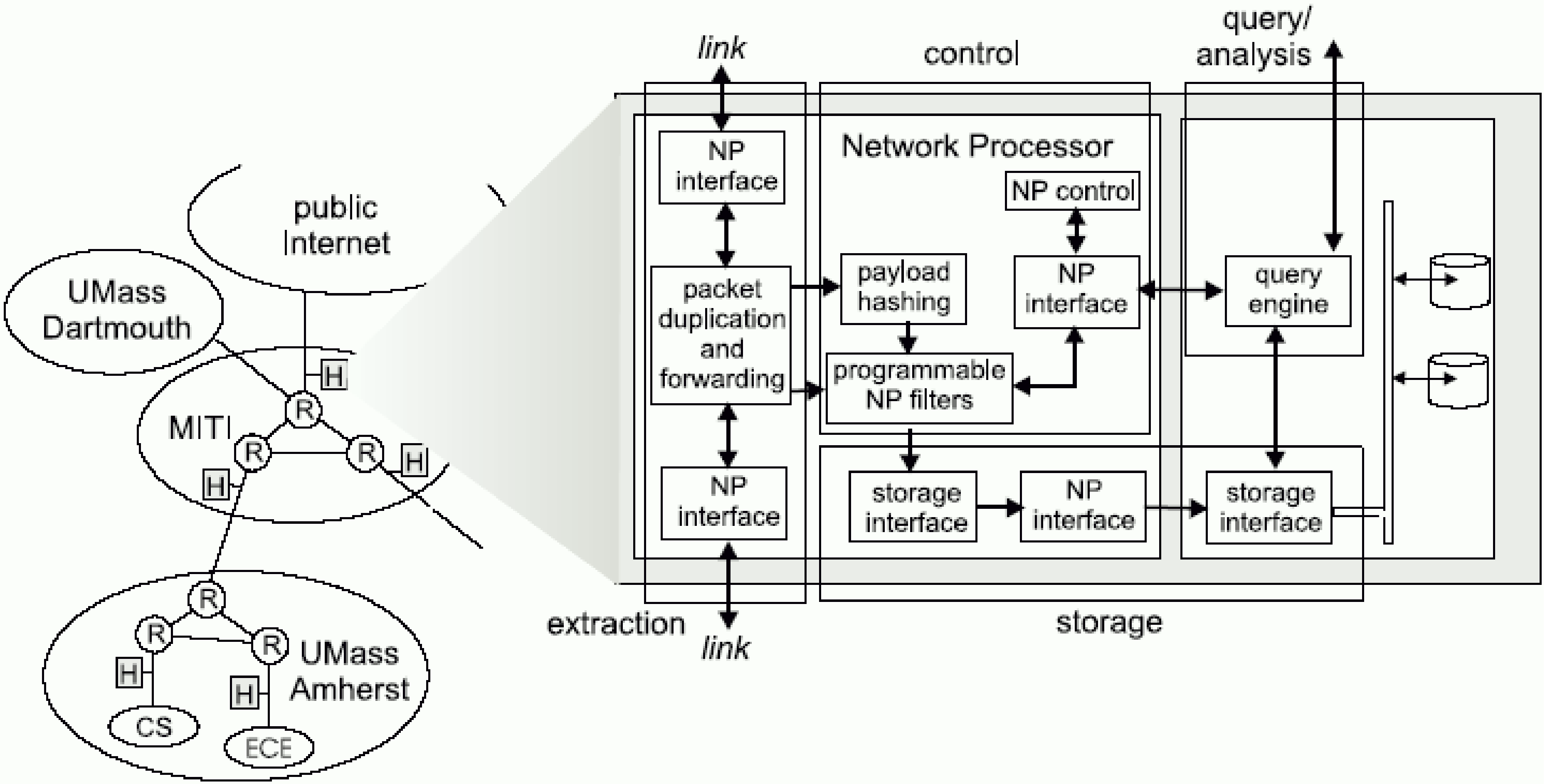
Hyperion Project

- Distributed passive measurement platform
  - Multiple measurement node in network
  - Coordinated traffic collection and storage
- Performance challenge:

System	link speed	avg. pkt. size	pkt. rate to storage	data/hour	window size for 4 TB
small	100M Ethernet (100 Mb/s)	100 bytes	125 kp/s	25 GB/h	160 hours
medium	Gigabit Ethernet	100 bytes	1.25 Mp/s	300 GB/h	13 hours
large	OC-192 (10 Gb/s)	100 bytes	12.5 Mp/s	2.5 TB/h	1.5 hours

  - Extraction, storage, and retrieval requires high performance
- Network processors can be used for extraction and pre-processing

Hyperion Node Architecture



Privacy Issues

- Passive measurements observe all traffic in network
  - Users have rights to privacy
  - Measurement data can reveal lots of personal information
- Examples of personal information
  - Web pages visited
  - Usernames and passwords (if not encrypted)
  - Emails, IM, etc.
  - Even encrypted traffic reveals information
- One possible solution: anonymization of traces
  - "Scramble" IP addresses
  - Prefix-preserving hashing is preferable over random hashing
  - Computationally expensive

Lab 3

- Use of IXP1200 Hardware in Lab
  - Thursday (12/4): 4:00pm-5:30pm
  - Friday (12/5) 1:00pm-2:00pm
  - Monday (12/8) 1:00pm-2:00pm
- No programming, just measurement
- Measurement of forwarding performance
  - Direct wire
  - wwwbump (see book Chapter 26)
  - IPv4 forwarding
- Use iperf and tcpdump tool to collect data
- Due 12/9/03

Next Class

- Course Summary
  - Any topics you want to cover?
- Help for final projects
- Course Evaluation