Network Measurements: Passive, Active and Hybrid approaches

Stefano Traverso
stefano.traverso@polito.it

Why running measurements?

“The Internet is the first thing that humanity has built that humanity doesn't understand, the largest experiment in anarchy that we have ever had.”

Eric Schmidt, former Google CEO.

• Fundamental cornerstone to build an “idea” of what is going on in the network
  – Network monitoring
  – Troubleshooting
  – Security
  – Characterization of users/devices/services
Network Measurements

– **Active Measurements**
  • Study cause/effect relationships by injecting extra traffic into the network and observe reactions
  • Full control on the generated traffic
  • Lack of generality

– **Passive Measurements**
  • Analysis of traffic traces captured by sniffers
  • Study traffic properties without interfering with it
  • Study traffic generated from actual Internet users
  • In general, require a large effort

– **Hybrid Measurements**
  • Take the best of the passive and active approaches
The Active Measurements Scenario

Active Measurements

- The basic Swiss knife:
  - Ping
    - Host reachability tests
  - Traceroute
    - Detects the routers to reach a given hosts
  - Nmap
    - Detects open ports and hosts
  - Iperf
    - Measures the throughput to a host
  - ...

May be deployed in several vantage Points e.g., PlanetLab
Ping

- Based on ICMPs echo messages
- Measures RTT (min, max, avg, std), TTL, packet loss, ...

$ ping www.google.com

PING www.google.com (173.194.35.20): 56 data bytes
64 bytes from 173.194.35.20: icmp_seq=0 ttl=54 time=2.715ms
64 bytes from 173.194.35.20: icmp_seq=1 ttl=54 time=2.740ms
64 bytes from 173.194.35.20: icmp_seq=2 ttl=54 time=3.218ms
64 bytes from 173.194.35.20: icmp_seq=3 ttl=54 time=2.750ms
64 bytes from 173.194.35.20: icmp_seq=4 ttl=54 time=2.932ms
^C

--- www.google.com ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 2.715/2.871/3.218/0.190ms

Stefano Traverso - TNG group - Politecnico di Torino

Ping – advanced use

- Flood the host

$ sudo ping -f 130.192.9.61
PING 130.192.9.61 (130.192.9.61) 56(84) bytes of data.
.^C

--- 130.192.9.61 ping statistics ---
30734 packets transmitted, 30733 received, 0% packet loss, time 8514ms
rtt min/avg/max/mdev = 0.187/0.243/0.828/0.039ms, ipg/ewma
0.277/0.224ms

- Get the list of traversed hops

$ ping -R 130.192.2.90
PING 130.192.2.90 (130.192.2.90) 56(124) bytes of data.
64 bytes from 130.192.2.90: icmp_req=1 ttl ms NOP
RR: 130.192.91.74
  130.192.2.105
  130.192.2.90
  130.192.2.90
  130.192.91.65
  130.192.91.74
64 bytes from 130.192.2.90: icmp_req=2 ttl=254 time=6.15ms NOP (same route)

Stefano Traverso - TNG group - Politecnico di Torino
Ping – advanced use (cont’d)

- **Measure the buffer size** of your ADSL modem

  ```
  $ num=11 ; sudo ping -l $num -c $num -a 1472 130.192.9.61
  PING 130.192.9.61 (130.192.9.61): 1472 data bytes
  1480 bytes from 130.192.9.61: icmp_seq=0 ttl=50 time=55.363 ms
  1480 bytes from 130.192.9.61: icmp_seq=1 ttl=50 time=68.527 ms
  1480 bytes from 130.192.9.61: icmp_seq=2 ttl=50 time=79.940 ms
  1480 bytes from 130.192.9.61: icmp_seq=3 ttl=50 time=92.876 ms
  1480 bytes from 130.192.9.61: icmp_seq=4 ttl=50 time=107.227 ms
  1480 bytes from 130.192.9.61: icmp_seq=5 ttl=50 time=120.439 ms
  1480 bytes from 130.192.9.61: icmp_seq=6 ttl=50 time=133.647 ms
  1480 bytes from 130.192.9.61: icmp_seq=7 ttl=50 time=147.563 ms
  1480 bytes from 130.192.9.61: icmp_seq=8 ttl=50 time=160.042 ms
  1480 bytes from 130.192.9.61: icmp_seq=9 ttl=50 time=177.829 ms
  --- 130.192.9.61 ping statistics ---
  11 packets transmitted, 10 packets received, 9.1% packet loss
  round-trip min/avg/max/stddev = 55.363/114.345/177.829/38.607 ms
  ```

Geolocation of IP addresses using multi-lateration techniques
Ping – advanced use (cont’d)

- Ping’s reliability is impaired by flow-based load balancing
- It can not reliably represent the performance experienced by applications

- In-flow RTT is more reliable
- Use tokyo-ping

Ping – advanced use (cont’d)

- Often blocked by routers (they do not reply to ICMP probes)
- Use other protocols than ICMP
  - UDP (as traceroute does)
  - TCP SYN/ACK/FIN/RST messages
  - nping
Traceroute

• Print the route packets travel to network host

$ traceroute www.google.it
traceroute to www.google.it (74.125.232.151), 30 hops max, 60 byte packets
 1 130.192.91.65 (130.192.91.65) 2.021ms 2.203ms 2.427ms
 2 18-c6500-vss.polito.it (130.192.2.65) 0.377 0.424ms 0.485ms
 3 mz-c-polifi.polito.it (130.192.232.60) 1.432ms * 1.727ms
 4 13-garr.polito.it (130.192.232.254) 6.472ms 6.704ms 7.211ms
 5 ru-polito-rx1-tol.tol.garr.net (193.206.132.33) 7.520ms 7.907ms 8.205ms
 6 rx1-tol-rx1-mi2.mi2.garr.net (90.147.80.217) 11.472ms 11.148ms 11.088ms
 7 rx1-mi2-r-mi2.mi2.garr.net (90.147.80.73) 9.654ms 8.722ms 8.831ms
 8 r-mi2-google.mi2.garr.net (193.206.129.130) 4.044ms 6.412ms 6.398ms
 9 * 209.85.249.54 (209.85.249.54) 8.203ms 20.474ms
10 64.233.174.243 (64.233.174.243) 8.294ms 8.311ms 8.745ms
11 mil02s05-in-f23.1e100.net (74.125.232.151) 8.617ms 8.709ms 9.336ms

Traceroute (cont’d)

• Sends UDP/IP packets with short TTLs to induce routers to reply with ICMP “Time to Live exceeded in Transit”

$ nping --ttl=2 --udp 130.192.232.60
Starting Nping 0.6.00 (http://nmap.org/nping) at 2014-12-16 10:05 CET
SENT (0.0047s) UDP 130.192.91.74:53 > 130.192.232.60:40125
  ttl=2 id=17368 iplen=28
RCVD (0.0182s) ICMP 130.192.2.65 > 130.192.91.74 TTL=0 during transit (type=11/code=0) ttl=254 id=62922 iplen=56
Traceroute (cont’d)

- Fails in presence of routers employing load balancing based on the packet header
- Use `paris-traceroute`

Nmap

- Tool for network discovery and security auditing
  - Given a host
    - It identifies open ports
    - It identifies the OS
  - Given a network
    - It identifies hosts that are up
Nmap (cont’d)

• **SYN stealth** port scan
  
  – Sends TCP SYN message
  
  • if ACK received, port = **open**
    
    – Immediately sends RST message to close the connection
  
  • Else, port = **closed**

```
$ sudo nmap -sS 130.192.9.61
```

```
Starting Nmap 6.00 ( http://nmap.org ) at 2014-12-16 17:28 CET
Nmap scan report for 130.192.9.61
Host is up (0.00069s latency).
Not shown: 992 closed ports
PORT     STATE SERVICE
22/tcp open   ssh
80/tcp open   http
3306/tcp open  mysql
6000/tcp open  X11
9418/tcp open  git
```

Nmap (cont’d)

• **A network scan**

```
$ nmap -sP 130.192.91.0/25
```

```
Starting Nmap 6.00 ( http://nmap.org ) at 2014-12-16 17:21 CET
Nmap scan report for aslan.polito.it (130.192.91.1)
Host is up (0.00025s latency).
Nmap scan report for l3-areait-91.polito.it (130.192.91.17)
Host is up (0.0048s latency).
Nmap scan report for amedeon12.polito.it (130.192.91.101)
Host is up (0.0014s latency).
...
Nmap scan report for 130.192.91.102
Host is up (0.00059s latency).
Nmap scan report for printern11.polito.it (130.192.91.126)
Host is up (0.00072s latency).
Nmap done: 128 IP addresses (14 hosts up) scanned in 1.71 seconds
```
Iperf

- It measures TCP and UDP bandwidth performance
  - Useful to
    1. Run speed-tests
    2. Create artificial traffic
  - Requires the user to have control of both end points

Iperf (cont’d)

Server
$ iperf -s

Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)

[  4] local 130.192.91.74 port 5001 connected with 2.230.134.65 port 51957
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0-11.4 sec   768 KBytes   550 Kbits/sec

Client
$ iperf -c 130.192.91.74

Client connecting to 130.192.91.74, TCP port 5001
TCP window size: 129 KByte (default)

[  4] local 192.168.1.76 port 58733 connected with 130.192.91.74 port 5001
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0-10.3 sec   768 KBytes   611 Kbits/sec
DNS-level active measurements

- Useful to dissect the infrastructure of Internet services
  - E.g., Content Delivery Networks, Cloud-based services
- Different DNS servers provide often different results, based on client location

NSlookup

$ nslookup eu.sc-proxy.samsungosp.com 130.192.3.21
Server: 130.192.3.21
Address: 130.192.3.21#53

Non-authoritative answer:
Name: scloud-pew1-prx-493782659.eu-west-1.elb.amazonaws.com
Address: 54.246.187.126
Name: scloud-pew1-prx-493782659.eu-west-1.elb.amazonaws.com
Address: 54.194.18.205
HTTP-level active measurements

- Web servers provide information about the services they host
- Probe the server with artificial HTTP requests
  - **HTTP-Knocking**
  - Using *automatic (headless) browsers*

---

**HTTP-Knocking**

- Probe a web server with a simple HTTP
  
  ```
  $ curl --head 54.231.128.184
  HTTP/1.1 405 Method Not Allowed
  x-amz-request-id: ACF953EAEC5F234B
  x-amz-id-2:
  eUqWzyXGRJUDhfpnH00f5X2L84hTCB9mI8dzD2jdCSSNQHJBTg0Iztr403RLbR9m
  Allow: GET
  Content-Type: application/xml
  Transfer-Encoding: chunked
  Date: Wed, 17 Dec 2014 16:21:48 GMT
  Server: AmazonS3
  ```
Automatic Browsing

- Selenium WebDriver [http://www.seleniumhq.org]
  - Can be combined with
    - Standard browsers (Firefox, Chrome, etc.)
    - Headless browsers (Phantomjs [http://phantomjs.org])
  - Emulates users interacting with a webpage:
    - Important for web developers to test their designs
    - Useful for researchers to build crawlers
      - E.g., infer Twitter’s social graph
The Passive Measurement Scenario

Passive measurements

• The basic Swiss knife:
  – **TCPdump**
    • Full packet-level captures
  – **Tstat, Netflow**
    • Flow-level captures, aggregated statistics
  – **Netstat**
    • Host-level, aggregated statistics
  – **Wireshark**
    • Full packet-level captures with nice GUI

May be deployed in several vantage points, e.g., probes within a large network

Works on a single machine
Netstat (cont’d)

- Summarizes the connections, interface statistics, etc. for a given host
- Useful to monitor server machines

$ netstat -s
Tcp:
  132861 active connections openings
  39 passive connection openings
  131967 failed connection attempts
  139 connection resets received
  6 connections established
  185494 segments received
  202549 segments send out
  39519 segments retransmited
  4 bad segments received.
  270 resets sent

Stefano Traverso - TNG group - Politecnico di Torino

$ netstat
Active Internet connections (w/o servers)
Proto  Recv-Q  Send-Q  Local Address        Foreign Address  State
tcp    0         0  localhost:50398      localhost:23414        ESTABLISHED
tcp    0         0  localhost:25001      localhost:41428        ESTABLISHED
tcp    0         0  localhost:41428      localhost:25001        ESTABLISHED
tcp    0         0  localhost:23414      localhost:50398        ESTABLISHED
tcp    0         0  grigio.local:53723  snt-re4-10c.sjc.dr:http ESTABLISHED
tcp    0       48  grigio.local:ssh   2.230.134.65:56181        ESTABLISHED
tcp    0         1  grigio.local:42397  stefanonli.polito:24800 SYN_SENT

Stefano Traverso - TNG group - Politecnico di Torino
Tcpdump

- It “dumps” the traffic it observes flowing on a network from its vantage point
- It inspects packets at network level
  - E.g., It does not rebuilds TCP flows
- It can generate large amounts of data
  - In pcap and txt formats

```
$ sudo tcpdump -nvvXS src 130.192.91.73
```

11:31:10.863681 IP (tos 0x0, ttl 64, id 62735, offset 0, flags [DF], proto TCP (6), length 40)
```
130.192.91.73.62266 > 17.254.32.16.80: Flags [.] cksum 0xb3f6
```

Tcpdump (cont’d)

It provides very detailed view on the packets

```bash
$ sudo tcpdump -nvvXS src 130.192.91.73
```

11:31:10.863681 IP (tos 0x0, ttl 64, id 62735, offset 0, flags [DF], proto TCP (6), length 40)
```
130.192.91.73.62266 > 17.254.32.16.80: Flags [.] cksum 0xb3f6
```

Stefano Traverso - TNG group - Politecnico di Torino
Wireshark

- Tcpdump with a nice GUI 😊
- Same very detailed view on the traffic
- It rebuilds TCP/HTTP/... connections
- Let the user run specific analysis in a few clicks
  - Best tool for quick troubleshooting
- Load the traffic into the memory 😞
  - Not suitable to load/capture GBs of traces

Wireshark (cont’d)
Tstat is a **passive sniffer developed by Polito**

- It works at flow-level
  - Monitors at network level, and re-builds flows at transport-level
- Computes several L3/L4 metrics (e.g., #pkts, #bytes, RTT, TTL, etc.)
  - Performs traffic classification
    - Deep Packet Inspection (DPI)
    - Statistical methods (Skype, obfuscated P2P)
- Different output formats (logs, RRDs, histograms, pcap)
- Run on off-the-shelf HW
  - Up to 2Gb/s with standard network interfaces
- Currently adopted in real network scenarios (campus network and ISPs)

---

### Tstat – deployment scenario

![Tstat deployment scenario diagram](image-url)
How Tstat works?

TCP logs (100+ stats)

Classification
- HTTP Transaction
- P2P
- Skype

HTTP logs

Tstat + DN-Hunter

DNS conversation
- The worst enemy of a passive sniffer is traffic encryption (HTTPS)
- DN-Hunter is a plugin which inspects DNS conversations (which are not encrypted, yet)
- DNS-level information is attached to TCP logs
Tstat – Use case:
Understanding Amazon AWS

Focus:
1. Evolution over time of AWS: the number of datacenters, their locations, and performance as perceived by the ISP customers
2. Analysis of popular Cloud/CDN-based web-services: show their dynamics, how they perform, etc.

Analysis of traffic generated by Clouds and CDNs
Tstat – Use case: NetCurator

Web app based on Tstat which extracts interesting links browsed by users from HTTP traffic
http://tstat.polito.it/netcurator/

Hybrid Measurements
Network Measurements...

- **Active measurements**
  - Let us exactly define the workload the network is subject to
  - Allow to measure cause/effect relationships easily
- **Passive measurements**
  - Do not change the working point of the network
  - More detailed view of the network status

...with some drawbacks

- **Active measurements**
  - Difficult to obtain detailed information about Transport and Network Layers
- **Passive measurements**
  - Generate lots of data to process
  - Fail when no (useful) traffic is generated
Hybrid Measurements

- **Combine** active and passive approaches:
  1. **Augment** measurements from active tests with **detailed IP/TCP-level** information obtained passively
  2. Active measurements generate the desired amount of traffic → **less logs to analyze**

---

Hybrid Measurements for **Network QoS Troubleshooting**
Use case: Speed Tests in ISP network

- **Active probe:** IQM (by Fastweb)
- **Passive probe:** Tstat

![Diagram showing FTP Client, Traceroute, Ping, Headless Browser, FTP Server, Traceroute, IQM Client, and IQM Server connections.]

**Testbed & Dataset**

- **30+ IQM client probes uniformly distributed within Fastweb edge network**
- **Tstat installed on the IQM server probe**
- **Network configurations**
  1. **ADSL:**
     - U-1Mbps/D-16Mbps
     - U-1Mbps/D-12Mbps
     - U-0.5Mbps/D-8Mbps
  2. **FTTH**
     - U-10Mbps/D-10Mbps
- **10-sec long speed-test every 4min**
  - FTP transfers in both Upload and Download
- **3 months of tests (Feb 1st to Apr 30th 2014)**
- **1.2M+ speed-test reports**
Congestion in the Virtual Leased Network

ADSL U1Mbps/D12Mbps

Evident day/night periodicity
+ Large std(RTT) and RTX rate

Passive measurement

Congestion in the Virtual Leased Line

Stefano Traverso - TNG group - Politecnico di Torino

Congestion at the Uplink

FTTH U10Mbps/D10Mbps

FTTH (no low SNR issues)
+ No day/night periodicity
+ large RTX segment rate

Active Test

Passive measurement

Small buffer at the uplink (thanks ping 😊)

Stefano Traverso - TNG group - Politecnico di Torino
Thanks!