CNT 5106C Computer Networks, Spring 2022

Instructor: Prof. Ahmed Helmy

Homework 1 – Part 2: Internet Architecture & Application Layer

Due Date: Feb 28, 2022 through Canvas

Instructions: Be precise and to the point. Many questions require answers using a sentence or two. Some questions will ask you to elaborate, use visual aids or graphs, or show traces/code. Use your own words and phrases, do not copy from any other source.

Q1 <6 points>. A. Application layer: What are the main network application architectures in the Internet? (mention 3)

Answer: This was discussed in the application layer context, and programming network applications, in the 1st 12 slides of ch2.

> The three models are: 1- client-server model, 2- peer-to-peer model, 3- hybrid (using both client-server and peer-to-peer)

B. What are the processes are needed to support all the above architectures? Explain which process is used in which architecture and how.

> Answer: only two processes are needed in all these models: the client process and the server model. [Extra: in client-server model the client process runs at the end-system (computer, laptop) and the server process runs on the server. In peer-to-peer model, each end-system runs both processes. They hybrid model encompasses both the other models.]

Q2. <13 points> What are the advantages and disadvantages of:

A. Hierarchical network architectures (mention 3 advantages and 2 disadvantages)
Answer: Ch1 slide 1-44

- Advantages
  - Isolates and scopes internal dynamics: dampens oscillations, providing stability to the overall network
  - Supports scalability: aggregation/summary per domain for smaller, more efficient routing tables
  - Allows for flexibility: domains deploy different protocols, and policies
- Disadvantages
  - Overhead of establishing and maintaining the hierarchy (esp. for mobile, dynamic networks)
  - Sub-optimality of routing

B. Protocol layering (mention at least 2 advantages and 2 disadvantages)
Answer: Ch1 slide 1-51 (two advantages and two disadvantages are enough)

Advantages:

- explicit structure allows identification, relationship of complex system's pieces

- layered reference model for discussion
- modularization eases maintenance and updating of the networking system
  - change of implementation of layer's service transparent to rest of system
  - change in one layer doesn't affect rest of system (in theory)

Disadvantages:

- overhead added to the traffic (in terms of packet headers at every layer)
- isolation between layers prevents cross-layer optimization if needed (particularly applicable in the context of mobile networking and the Internet of Things (IoT)
- change in one layer can sometimes affect upper layers unintentionally (think TCP over wireless links)

C. Stateless protocols (mention at least 2 advantages and 2 disadvantages)
Answer: Ch2 slide 2-22 (2 in each is enough)
Advantages

- past history (state) does not need to be maintained (as in stateful protocols)
- it is more resilient to crashes and failures (by contrast, in stateful protocols if server/client crashes, their views of "state" may be inconsistent, must be reconciled)
- It is simpler than stateful protocols (protocols that maintain "state" are complex!)

Disadvantages:

- because there is no requirement to maintain/remember previous state, the setup is needed with every new connection (which may mean re-entering credentials, passwords, etc).
- state can added via 'cookies', which brings up some issues of privacy and security
- since the state is not kept, more overhead is needed to 'carry' the state in the messages since it is not stored.

Q3. <6 points> A. How were the original Internet requirements met through its design? (mention 4)

Answer: Ch1 slide 1-66

- Scalability & economic access:
  - Resource sharing, reduce reservations, allow for higher utilization
  - Use of packet switching (statistical multiplexing) instead of circuit switching
- Robustness:
  - Re-routing around failures
  - Stateless connections, dynamic routing
- Reliablility:
  - Timed retransmission, based on acks, seq. #s

- Evolvablility:
  - ❖ Minimize complexity in the network and push functionality to the edges (*end-to-end* principles)

B. What are the two main requirements that you see missing from the original design that are much needed today?

Answer: Security and support for seamless mobility

Q4. <6 points> In slide 1-71 of chapter 1 discussed in class, explain the probability of '0.0004' when 35 users are active.

Answer: The system of packet switched (statistically multiplexed) network or link with capacity of 1Mbps, with N sources each with 100kbps and active 10% of the time. The capacity of this system will be exceeded when more than 10 users are active at the same time. If we express the number of users that are active at the same time by a random variable called *x*, and the probability that a source is active is *p*=10%, then we want to get the probability P(*x*>10). Assuming the sources are independent and identically distributed, then we can look at this system as having N=35 experiments, each could have an outcome of success '*on*' with 10% or failure '*off*' with 90%, then we can express the probability of having *x* of these experiments as success '*on*' using the binomial distribution, with $P(X = x) = \binom{N}{x} p^x (1-p)^{N-x}$, where $\binom{N}{x} = \frac{N!}{(N-x)!x!}$.

The probability that more than 10 sources are on at the same time is given by:

$P(X > 10) = \sum_{x=11}^{N} P(X = x)$

[using a binomial calculator or any other tool/program]

For *p*=10%=0.10, *N*=35, we get *P*(*X* > 10)= 0.0004243.

Q5. <4 points> A. What is a DDoS attack?

Answer: Distributed denial of service attack, where a large number/network of computers (botnet) can be compromised and used to launch a synchronized attack on a target machine/server. Usually each of the attacking computers is not injecting enough traffic to be detected, but the collective synchronized traffic can be quite harmful to limit the resources at the attacked server from performing its function efficiently (or at all).

B. why is it harder to control than a DoS attack?

Answer: DDoS is harder to control than DoS (which usually involves a single attacker), since a single attacker can be detected and prevented from injecting more traffic much more easily than a distributed system of many computers.

Q6. <4 points> What is the first Internet worm, and how did it harm the Internet? [hint: Watch video link posted on canvas]

Answer: The first worm in 1988 was the Morris worm developed by Robert Morris (for research/testing/experimentation reasons, not for malicious reasons). It was an experiment that went out of control unexpectedly. It consisted of code that copied itself from one computer and networked device to another passively (without the need for human intervention). It led to the

infection of a very large number of computers connected to the Internet at that time, sometimes preventing those machines from booting up or functioning correctly. [Note: Robert Morris is currently a well-respected researcher and Prof. of CS at the MIT CSAIL lab, working on wireless mobile networking among other projects.]

Q7. <5 points> A. Why is UDP preferred over TCP for IP-telephony/VoIP (like Skype)?

Answer: VoIP is an application that is sensitive to delays and delay jitters (variations), more than it is sensitive to loss (it can tolerate 10% or more loss in many cases and still be usable). UDP provides less delays than TCP, as it does not require connection (handshake), and does not perform reliability (re-transmission) or congestion control (regulation within packet windows). Hence UDP matches VoIP requirements better than TCP.

B. Why would Skype sometimes use TCP? Give two reasons.

Answer: Many firewalls block or rate-limit the UDP traffic, for security reasons. Since UDP traffic is not regulated (like TCP's), it can be used in many attacks. In those cases, if the VoIP over UDP is blocked then TCP may be used to get around the firewall restriction.

Skype uses several steps to setup the connection (using client-server model to contact the Skype server, then contacting the super-nodes to conduct search). TCP can be used in those phases.

Q8. <5 points> Would an application that needs congestion control ever use UDP? Give two examples to support your argument.

Answer: Yes.

First, an application that needs congestion control without the need for reliability and re-transmissions, may want to implement congestion control at the application layer over UDP.

Second, an application that needs congestion control that is different from that provided by TCP (which is window-based), may want to implement its own protocol (say rate-adaptation) at the application layer over UDP.

Q9. <6 points> How do web caches/proxy servers help Internet performance? Explain and list 3 of its benefits from the user and network perspectives. [hint: explain using your understanding of elementary queueing theory and delays, and use graphs as needed]

Answer: Ch2 slides 2-30 to 2-35

- 1. reduce response time for client request (from the user perspective)
- 2. reduce traffic on an institution's access link (from the network perspective)
- 3. Internet dense with caches: enables "poor" content providers to effectively deliver content (so too does P2P file sharing)

Points 1 and 2 above are achieved with the local proxy cache, which (assuming reasonable cache hit ratio) avoids sending most of the traffic over the access link. So instead of having link utilization (or traffic intensity) close to 80-90% leading to exponential queueing delay

(perhaps minutes of delay), the goal is to bring the traffic intensity with the help of the proxy cache close to 50-60% resulting in linear queueing delay (close to a few seconds or less in delay).

Q10. <7 points> Use '*traceroute'* and '*ping'* commands/tools to measure and analyze delays in the Internet:

A. Use *traceroute* to measure delays between your location and an overseas location (e.g., www.eurecom.fr). Show the trace and annotate it showing the transoceanic link.
B. Identify machines/routers along the way with:

1. less than1ms delay, 2. 2–10ms delay, 3. 11–100ms delay, more than 100ms delay

then *ping* those machines for 15seconds each and analyze their delays

C. Identify the locations of the machines and reason about the differences in delays

[hints: look at the traceroute example in the lecture/book and perform something similar. *traceroute* is called *tracert* on windows. On some machines you need to be super user (sudo) to run traceroute. You may run the commands from your machine or from a UF machine (e.g., storm.cise.ufl.edu), so try and see what works for you.]

Answer:

A- storm:16% traceroute www.eurecom.fr
traceroute to www.eurecom.fr (193.55.113.240), 30 hops max, 60 byte packets
1 _gateway (128.227.205.193) 0.503 ms 0.441 ms 0.481 ms
2 * * *
3 ssrb230a-nexus-msfc-1-v21-1.ns.ufl.edu (128.227.236.9) 0.456 ms 0.407 ms 0.314 ms
4 ssrb230a-pel-asr9001-1-v15-1.ns.ufl.edu (128.227.236.203) 1.173 ms 1.301 ms 1.426 ms
5 * * *
6 ssrb230a-ewan-msfc-1-v16-1.ns.ufl.edu (128.227.236.205) 1.054 ms 1.135 ms 1.096 ms
7 ctx36-pel-msfc-1-te15-1.ns.ufl.edu (128.227.236.175) 1.162 ms 1.819 ms 1.798 ms
8 renet-flrcore-108-59-26-114.rtr.net.flrnet.org (108.59.26.114) 4.318 ms 4.178 ms 4.111 ms
9 jax-flrcore-asr9010-1-hu0701-1.net.flrnet.org (108.59.31.150) 6.863 ms 8.081 ms 7.786 ms
10 prov-i2-atla-renet-1070.net.flrnet.org (108.59.25.21) 6.615 ms 6.565 ms 6.308 ms
11 et-3-3-0.4079.rtsw.atla.net.internet2.edu (162.252.70.42) 13.900 ms 13.550 ms 13.493 ms
12 ae-4.4079.rtsw.wash.net.internet2.edu (198.71.45.7) 25.044 ms 25.176 ms 24.953 ms
13 internet2-gw.mx1.lon.uk.geant.net (62.40.124.44) 99.824 ms 99.569 ms 99.501 ms
14 ae6.mx1.lon2.uk.geant.net (62.40.98.37) 100.731 ms 100.586 ms 100.488 ms
15 ae5.mx1.par.fr.geant.net (62.40.98.179) 110.408 ms 107.075 ms 107.223 ms
16 renater-lb1-gw.mx1.par.fr.geant.net (62.40.124.70) 107.144 ms 107.065 ms 107.047 ms
17 te0-6-0-4-lyon1-rtr-001.noc.renater.fr (193.51.177.219) 122.085 ms 193.51.180.61 (193.51.180.61) 124.681 ms te0-3-1-0-lyon1-rtr-001.noc.renater.fr (193.51.177.65) 124.692 ms
18 xe0-0-1-marseille1-rtr-131.noc.renater.fr (193.51.177.17) 119.891 ms xe1-0-1-marseille1-rtr-131.noc.renater.fr (193.51.177.222) 119.798 ms 119.786 ms

19 te1-2-sophia-rtr-021.noc.renater.fr (193.51.177.21) 121.780 ms 121.727 ms 138.205 ms
20 eurocom-valbonne-gi9-7-sophia-rtr-021.noc.renater.fr (193.51.187.17) 121.580 ms 121.568 ms 121.538 ms
21 * * *
22 * * *
23 * * *
24 * * *
25 * * *
26 * * *
27 * * *
28 * * *
29 * * *
30 * * *

in #13 above the delay jumps from ~25ms to ~100ms, going from the US to the UK across a transoceanic link

B- 1. less than1ms delay, 2. 2–10ms delay, 3. 11–100ms delay, more than 100ms delay

#1 above is the gateway (the first hop router) with less than 1ms delay

#3 is a router at uf with less than 1ms delay

#8 to #10 routers are outside ufl.edu but are in the florida net (i.e., in-state) with delays from 2ms-10ms

#11 and 12 are routers outside of Florida but within the US (before crossing the ocean) with delays between 11-99ms delays

routers 13 and beyond are routers in Europe (UK, France) with delays larger than 99ms

ping to #3 above results in:

storm:17% ping 128.227.236.9
PING 128.227.236.9 (128.227.236.9) 56(84) bytes of data.
64 bytes from 128.227.236.9: icmp_seq=1 ttl=253 time=0.549 ms
64 bytes from 128.227.236.9: icmp_seq=2 ttl=253 time=0.687 ms
64 bytes from 128.227.236.9: icmp_seq=3 ttl=253 time=0.623 ms
64 bytes from 128.227.236.9: icmp_seq=4 ttl=253 time=0.694 ms
64 bytes from 128.227.236.9: icmp_seq=5 ttl=253 time=46.4 ms
64 bytes from 128.227.236.9: icmp_seq=6 ttl=253 time=0.564 ms
64 bytes from 128.227.236.9: icmp_seq=7 ttl=253 time=320 ms
64 bytes from 128.227.236.9: icmp_seq=8 ttl=253 time=0.592 ms
64 bytes from 128.227.236.9: icmp_seq=9 ttl=253 time=0.592 ms
64 bytes from 128.227.236.9: icmp_seq=10 ttl=253 time=0.601 ms
64 bytes from 128.227.236.9: icmp_seq=11 ttl=253 time=276 ms
64 bytes from 128.227.236.9: icmp_seq=12 ttl=253 time=0.546 ms

64 bytes from 128.227.236.9: icmp_seq=13 ttl=253 time=0.603 ms
64 bytes from 128.227.236.9: icmp_seq=14 ttl=253 time=0.572 ms
64 bytes from 128.227.236.9: icmp_seq=15 ttl=253 time=96.2 ms
64 bytes from 128.227.236.9: icmp_seq=16 ttl=253 time=0.596 ms
^C
--- 128.227.236.9 ping statistics ---
16 packets transmitted, 16 received, 0% packet loss, time 15226ms
rtt min/avg/max/mdev = 0.546/46.651/320.319/98.591 ms

as can be observed the delay varies widely from 0.55-320ms

ping to # 10 above

storm:18% ping 108.59.25.21
PING 108.59.25.21 (108.59.25.21) 56(84) bytes of data.
64 bytes from 108.59.25.21: icmp_seq=1 ttl=56 time=6.86 ms
64 bytes from 108.59.25.21: icmp_seq=2 ttl=56 time=7.83 ms
64 bytes from 108.59.25.21: icmp_seq=3 ttl=56 time=6.83 ms
64 bytes from 108.59.25.21: icmp_seq=4 ttl=56 time=8.90 ms
64 bytes from 108.59.25.21: icmp_seq=5 ttl=56 time=9.85 ms
64 bytes from 108.59.25.21: icmp_seq=6 ttl=56 time=7.11 ms
64 bytes from 108.59.25.21: icmp_seq=7 ttl=56 time=7.74 ms
64 bytes from 108.59.25.21: icmp_seq=8 ttl=56 time=7.62 ms
64 bytes from 108.59.25.21: icmp_seq=9 ttl=56 time=6.88 ms
64 bytes from 108.59.25.21: icmp_seq=10 ttl=56 time=7.14 ms
64 bytes from 108.59.25.21: icmp_seq=11 ttl=56 time=6.84 ms
64 bytes from 108.59.25.21: icmp_seq=12 ttl=56 time=6.86 ms
64 bytes from 108.59.25.21: icmp_seq=13 ttl=56 time=6.74 ms
64 bytes from 108.59.25.21: icmp_seq=14 ttl=56 time=6.74 ms
64 bytes from 108.59.25.21: icmp_seq=15 ttl=56 time=6.82 ms
64 bytes from 108.59.25.21: icmp_seq=16 ttl=56 time=6.90 ms
^C
--- 108.59.25.21 ping statistics ---
16 packets transmitted, 16 received, 0% packet loss, time 15082ms
rtt min/avg/max/mdev = 6.744/7.358/9.857/0.856 ms

the delay varies to this router but not as much as to #1 above, with much less deviation

ping to #12 above

storm:19% ping 198.71.45.7
PING 198.71.45.7 (198.71.45.7) 56(84) bytes of data.
64 bytes from 198.71.45.7: icmp_seq=1 ttl=54 time=25.6 ms
64 bytes from 198.71.45.7: icmp_seq=2 ttl=54 time=25.4 ms

64 bytes from 198.71.45.7: icmp_seq=3 ttl=54 time=25.6 ms
64 bytes from 198.71.45.7: icmp_seq=4 ttl=54 time=25.6 ms
64 bytes from 198.71.45.7: icmp_seq=5 ttl=54 time=25.8 ms
64 bytes from 198.71.45.7: icmp_seq=6 ttl=54 time=25.4 ms
64 bytes from 198.71.45.7: icmp_seq=7 ttl=54 time=25.4 ms
64 bytes from 198.71.45.7: icmp_seq=8 ttl=54 time=25.5 ms
64 bytes from 198.71.45.7: icmp_seq=9 ttl=54 time=25.6 ms
64 bytes from 198.71.45.7: icmp_seq=10 ttl=54 time=25.5 ms
64 bytes from 198.71.45.7: icmp_seq=11 ttl=54 time=25.5 ms
64 bytes from 198.71.45.7: icmp_seq=12 ttl=54 time=25.3 ms
64 bytes from 198.71.45.7: icmp_seq=13 ttl=54 time=25.7 ms
64 bytes from 198.71.45.7: icmp_seq=14 ttl=54 time=35.3 ms
64 bytes from 198.71.45.7: icmp_seq=15 ttl=54 time=25.6 ms
64 bytes from 198.71.45.7: icmp_seq=16 ttl=54 time=25.5 ms
64 bytes from 198.71.45.7: icmp_seq=17 ttl=54 time=25.8 ms
^C
--- 198.71.45.7 ping statistics ---
17 packets transmitted, 17 received, 0% packet loss, time 16062ms
rtt min/avg/max/mdev = 25.349/26.171/35.301/2.287 ms

the standard deviation is bigger than that of #10 but smaller than that of #3

ping to #20 above:

storm:20% ping 193.51.187.17
PING 193.51.187.17 (193.51.187.17) 56(84) bytes of data.
64 bytes from 193.51.187.17: icmp_seq=1 ttl=237 time=120 ms
64 bytes from 193.51.187.17: icmp_seq=2 ttl=237 time=120 ms
64 bytes from 193.51.187.17: icmp_seq=3 ttl=237 time=120 ms
64 bytes from 193.51.187.17: icmp_seq=4 ttl=237 time=120 ms
64 bytes from 193.51.187.17: icmp_seq=5 ttl=237 time=120 ms
64 bytes from 193.51.187.17: icmp_seq=6 ttl=237 time=122 ms
64 bytes from 193.51.187.17: icmp_seq=7 ttl=237 time=120 ms
64 bytes from 193.51.187.17: icmp_seq=8 ttl=237 time=120 ms
64 bytes from 193.51.187.17: icmp_seq=9 ttl=237 time=121 ms
64 bytes from 193.51.187.17: icmp_seq=10 ttl=237 time=120 ms
64 bytes from 193.51.187.17: icmp_seq=11 ttl=237 time=120 ms
64 bytes from 193.51.187.17: icmp_seq=12 ttl=237 time=120 ms
64 bytes from 193.51.187.17: icmp_seq=13 ttl=237 time=121 ms
64 bytes from 193.51.187.17: icmp_seq=14 ttl=237 time=121 ms
64 bytes from 193.51.187.17: icmp_seq=15 ttl=237 time=127 ms
64 bytes from 193.51.187.17: icmp_seq=16 ttl=237 time=128 ms
64 bytes from 193.51.187.17: icmp_seq=17 ttl=237 time=125 ms
64 bytes from 193.51.187.17: icmp_seq=18 ttl=237 time=123 ms

64 bytes from 193.51.187.17: icmp_seq=19 ttl=237 time=123 ms
^C
--- 193.51.187.17 ping statistics ---
19 packets transmitted, 19 received, 0% packet loss, time 18072ms
rtt min/avg/max/mdev = 120.499/122.155/128.948/2.448 ms

all the delay is 120ms and above, mostly attributed to propagation delay across the atlantic (and back) which occurs at ~2x10^8m/s (electromagnetic wave speed). Variance is relatively low.

C- It is clear that those machines with min delay (less than 1ms) are either in the local network (gateway) or are inside of the UF campus (ufl.edu domain). Those with slightly higher delays are outside of ufl.edu but part of Florida flrnet. Then delays increase slightly outside of Florida but inside the US when we cross internet2. Finally, delay increases sharply when crossing the Atlantic to the UK then France, with delays exceeding 99ms.

The locations can be easily identified by looking at the router/machine address name, particularly the domain name.

Q11. <7 points> Visit the wireshark website at wireshark.org, read the user's manual (https://www.wireshark.org/docs/wsug_html_chunked/), then answer these questions:

A. What is wireshark?

Answer: Wireshark is a network packet sniffer and analyzer. A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible.

[Extra, detailed:

Wireshark is a packet sniffing tool for observing the messages exchanged between executing protocol entities.

A packet sniffer captures (obtains copies of) messages being sent/received from/by computers on the local network;

it stores and displays the contents of the various protocol fields in these captured messages.

It is a passive tool, as it observes messages being sent and received by applications and protocols running on the computer, but never sends packets itself. Similarly, received packets are never explicitly addressed to wireshark. Instead, a the tool

receives a copy of packets that are sent/received from/by application and protocols executing on the machine on which it runs.

The second component of a packet sniffer is the packet analyzer, which displays the contents of all fields within a protocol message.]

B. What are some of intended purposes? (mention four)

Answer: Here are some reasons people use Wireshark:

Network administrators use it to troubleshoot network problems

Network security engineers use it to examine security problems

QA (quality assurance) engineers use it to verify network applications

Developers use it to debug protocol implementations

People use it to learn network protocol internals

C. What are two unintended purposes?
[hints: install wireshark and start using it to prepare for future hwks. Read intro posted on canvas.]

Answer: Here are some things Wireshark does not provide:

Wireshark isn't an intrusion detection system. It will not warn you when someone does strange things on your network that he/she isn't allowed to do. However, if strange things happen, Wireshark might help you figure out what is really going on.

Wireshark will not manipulate things on the network, it will only "measure" things from it. Wireshark doesn't send packets on the network or do other active things (except domain name resolution, but that can be disabled).