OVERLAY NETWORKS AND COURSE REVIEW

George Porter
Week 10
Fall 2018

UC San Diego

1

ATTRIBUTION

UC San Diego

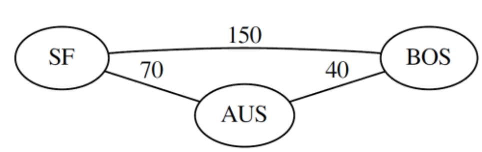
2

Outline

- Overlay networks
  - P&D Sec 9.4 if interested
- Final exam overview
- Open Q&A

3

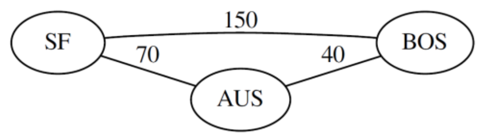
INEFFICIENT PATHS



- SF->BOS is one hop, 150ms
- But why not two-hop, 70+40=110ms??

4

STANDARD INTERNET ROUTING

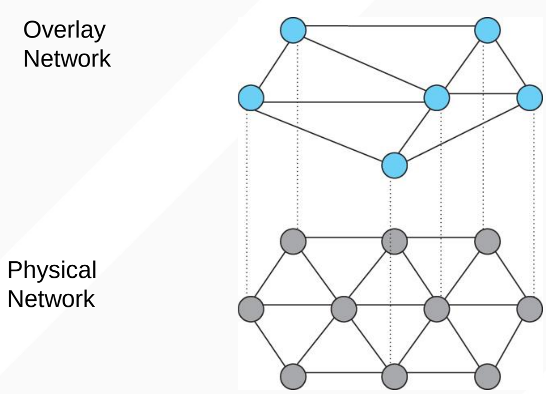


- Paths chosen via Border Gateway Protocol (BGP)
  - Based on business relationships, contracts, etc.
  - Not (necessarily) based on latency or bandwidth

5

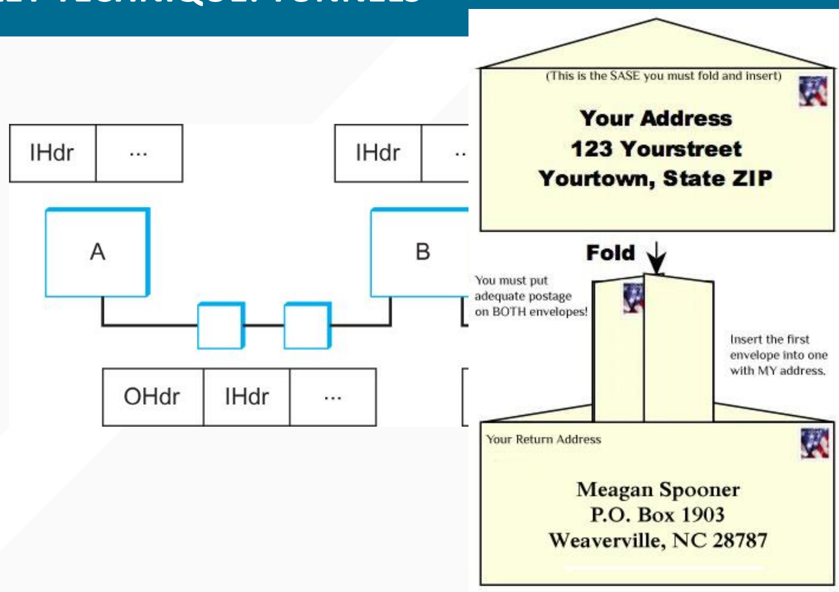
OVERLAY NETWORKS

- Control routing *at the application layer*



6

KEY TECHNIQUE: TUNNELS



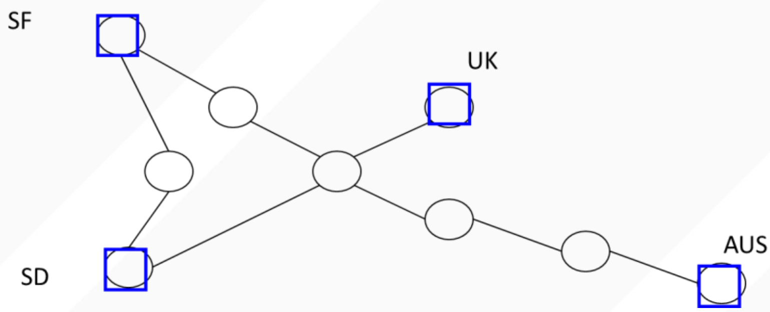
7

WHY OVERLAYS?

- For routing
  - Get better than best-effort service on the Internet
  - For reliability (what if a link on the internet drops p percent of packets?)
- To locate data
  - Remember Chord! It is a type of overlay network
- For security
  - What if you encrypt the overlay?
  - Also called a VPN (Virtual Private Network)

8

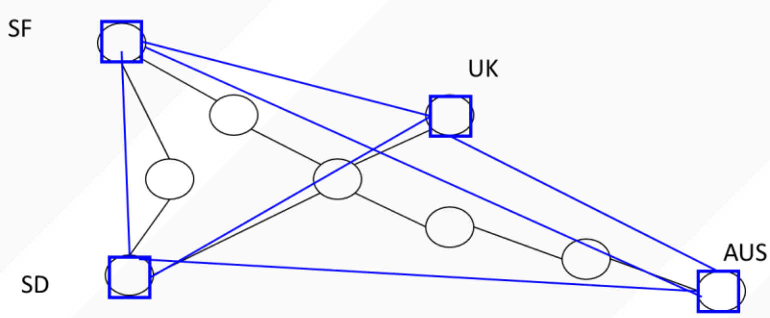
THE NEED FOR MULTICAST



- Imagine each office wants to send a message to every other office

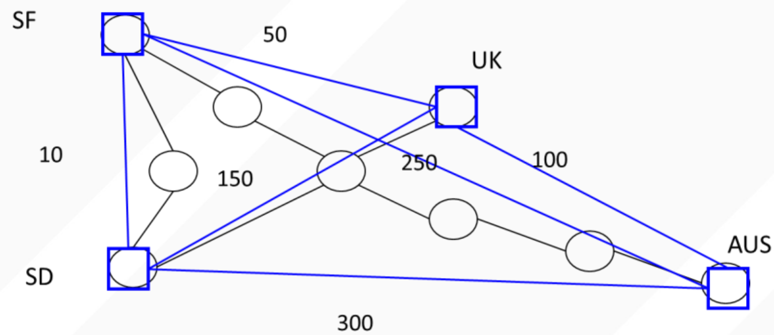
9

STRAWMAN SOLUTION: JUST SEND EVERYWHERE



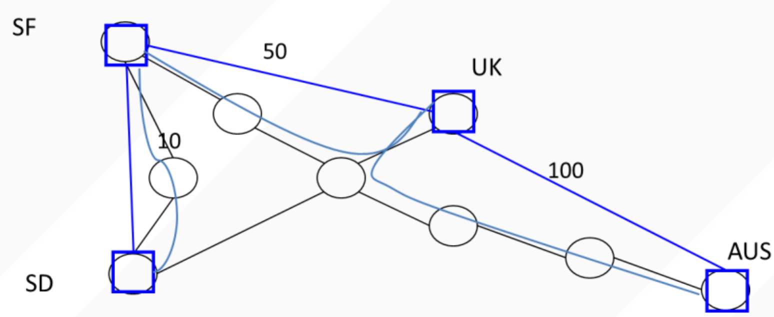
10

CONSTRUCTING AN OVERLAY MULTICAST TREE



11

CHOOSE MINIMUM COST SPANNING TREE



12

OVERLAY NETWORKS FOR PERFORMANCE

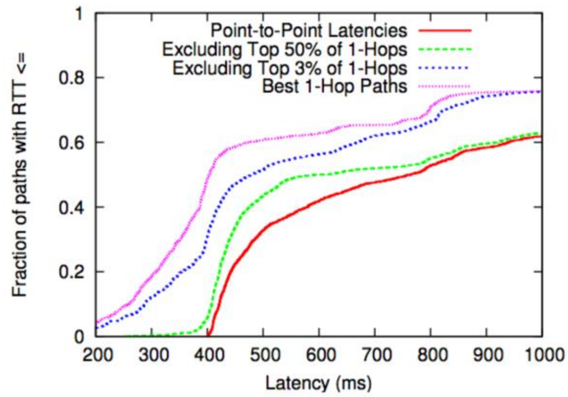
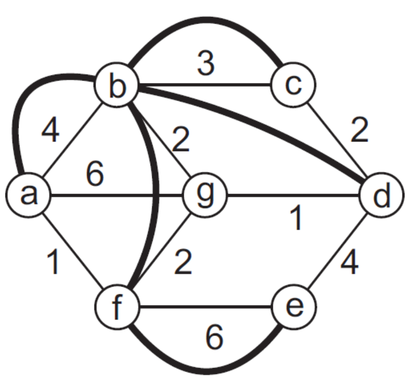


Figure 1: Comparison of RTT for pairs of PlanetLab hosts whose point-to-point latencies were larger than 400 ms (high-latency paths). For the "excluding top n%" graphs, we removed the top n% of one-hop alternatives for each high-latency path from consideration, then used the best remaining one-hop.

Sontag et al.,
CoNEXT 2009

13

APP-LAYER OVERLAY EXAMPLE



- "Tree" constructed using application-layer sockets
- Data flows along tree, not underlying network
- Why?
  - Can improve reliability
    - If link from B->G fails, can take few minutes for Internet to recover
      (meanwhile app can respond in milliseconds to create new path)
  - Disseminate data in a scalable way
  - Avoid censorship

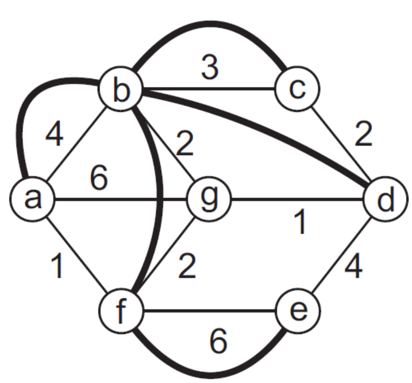
14

KEY CONCEPTS

- Link stress
  - How often a packet transits a given link
- Relative delay penalty (aka "Stretch")
  - Ratio of delay in overlay vs. underlying network

15

APP-LAYER OVERLAY EXAMPLE

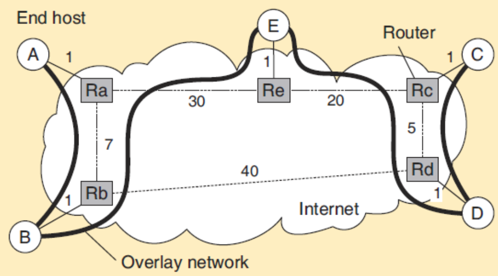


- Network cost A -> F
  - 1
- Overlay cost A -> F
  - 4 + 2 + 2 = 8
- Relative delay penalty A -> F
  - 8/1

16

ANOTHER EXAMPLE

Different metrics



- Link stress: How often does an ALM message cross the same physical link? Example: message from A to D needs to cross $\langle Ra, Rb \rangle$ twice.
- Stretch: Ratio in delay between ALM-level path and network-level path. Example: messages B to C follow path of length 73 at ALM, but 47 at network level $\Rightarrow$ stretch = 73/47.

17

Outline

- Overlay networks
  - P&D Sec 9.4 if interested
- Final exam overview
- Open Q&A

18

LOGISTICS

- Assigned seating
  - If you're left handed, and haven't let me know, please do so now!
- "Cheat sheet"
  - One 8.5x11" paper (both sides)
- No calculator needed
- Covers all material including papers, projects

19

NETWORK PROGRAMMING FUNDAMENTALS

- Network sockets API: open(), connect(), send(), recv(), etc
- How names are resolved to addresses in DNS
- End to end protocols
  - Move from host-to-host to process-to-process communication model
  - TCP – provide abstraction of reliable in-order byte stream on top of IP protocol, sliding window, ACKs
- IP addresses, netmasks, CIDR (e.g., /24)

20

NETWORKING FUNDAMENTALS

- Switching, forwarding
- Longest-prefix match
- Dijkstra, distributed Bellman-Ford
- Link state, Distance Vector
- Spanning Tree Algorithm
- ARP, Ethernet learning switches
- Performance (prop delay, latency, transmit time, serialization latency, queuing, …)

21

PROTOCOL DESIGN AND ANALYSIS

- Framing vs. parsing
- Delimiter vs. length-value
- Server-side protocol handling
- Request for comments documents (RFCs)
- Deep dive on HTTP

22

RPCS

- Explain concept of 'idempotent'
- Maybe vs at least once vs at most once semantics; how to implement each of these?
- Role of stub compiler, RPC runtimes
- Discuss whether the following operations are idempotent:
  - Pressing a lift (elevator) request button
  - Writing data to an offset in a file
  - Appending data to the end of a file (assuming there are no other writers in the system)

23

REPLICATION, CONSISTENCY AND FAULT TOLERANCE

- Lamport clocks, vector clocks, time synchronization, Total-ordered multicast
- Replicated state machines/logs
- Two-phase commit
- RAFT protocol
- Akamai CDN + DNS

24

DATA CENTERS AND CDNS

- Round-robin DNS vs load balancers. advantages and disadvantages of each
- Replication vs partitioning: advantages and disadvantages
- Terms: MTTR, MTBF, availability, yield, harvest, DQ principle
- Tail latency vs. average latency

25

OVERLAY NETWORKS, P2P, CHORD

- Compare and contrast aspects of flooding queries, supernodes, vs structured. Tradeoffs--which is better for joining, leaving, advertising content, querying for content.
- Assume you have a Chord system of 16 or 32 nodes with identifiers provided of the data and the servers
  - Draw the identifier circle and show which nodes the keys will be assigned to
  - Be able to perform "lookups" provided an example with finger tables provided
- Each chord node must maintain routing state.
  - Describe exactly what routing state must be maintained at each node to ensure correct function. Show what this state would be for a particular node. What is the expected lookup time of an object?
  - Describe what routing state must be maintained at each node to ensure fast lookup times. Show what this state would be for that node. what is the expected lookup time of an object?

26

SECURITY

- HTTPS security, very high level basics of TLS
- What are certificates?
- What is a CA?
- What is a cipher?

27

DURING THIS TERM… YOU

- Learned the basics of how the Internet works
- Wrote a web server
- Deployed and maintained code on 5 continents
- Measured the Internet empirically
- Built a distributed service that survives multiple failures while still operating
- …and hopefully had some fun too?

28

UC San Diego

29

Outline

- Overlay networks
  - P&D Sec 9.4 if interested
- Final exam overview
- Open Q&A

30

UC San Diego

31