CSC6730 – ADVANCED NETWORKING

END-TO-END ARGUMENT

Susmit Shannigrahi
Computer Science
sshannigrahi@tntech.edu

Tennessee
TECH

Discussions from previous class - logistics

- Ilearn, Class website, and Slack
  - iLearn – Only for assignments and announcements, nothing else!
  - Class website – Video recording, survey links, papers, and slides
  - Slack – I am (almost) always available on Slack.

- Use of technology
  - Let’s blame the chip shortage for everything
  - Nothing I can do at this moment
  - Department and ITS knows about this and is working on fixing it

- What is the main goal of the survey questions?
  - Understanding what needs to be discussed more
  - Clarify any problems
  - **Fill out after every class, reusable link!**

- Can not download papers
  - Has this been fixed?

Discussions from your feedback – class direction

- Why should I read these papers and how do they relate to the class
  - Course objectives:
    - The design rationale of the Internet
    - how the Internet supports contemporary applications
    - the shortcomings of the current design, and
    - the future research directions that can address those challenges.

    This should enable you to:
    - understanding of the networking research landscape
    - present your work to a larger group, both orally and in writing
    - Identify and present design choices, tradeoffs, and research gaps
    - design new network protocols and apply them to your applications ← Network should not be a black box!

Discussions from your feedback – class direction

- How do I read and understand the papers?
  - Read the papers linked from the first lecture
  - Meet with me – slack or email to set up a time

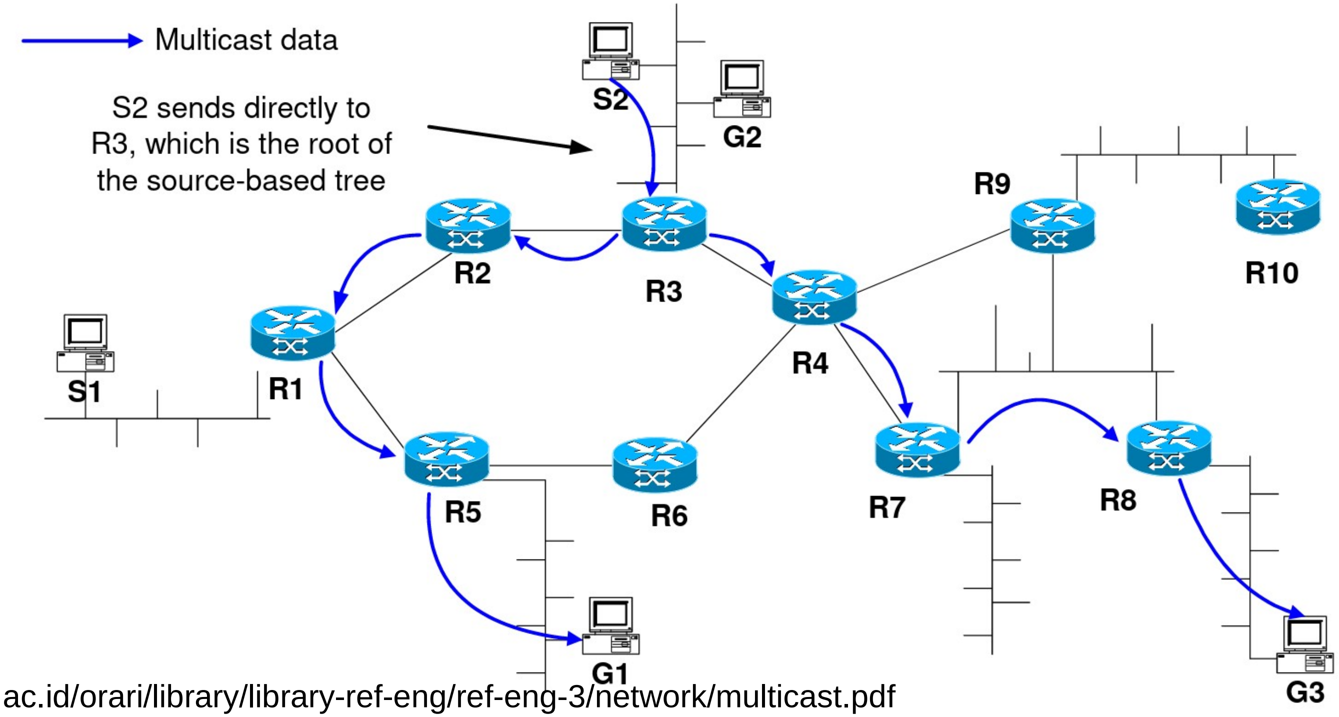
- More time for discussion
  - From next week, we have dedicated one day/week for discussion

Discussions from your feedback – technical questions

- Security ramification of broadcast signals ← Anthony
  - Not all traffic need to be secure ← Superbowl
  - Use encryption if needed ← End-to-end argument

Discussions from your feedback – technical questions

Had trouble understanding the issue with broadcast and multicast. I understand that TCP/IP is point-to-point, but it seems like switches are pretty good at broadcasting and multicasting despite this. ← Jordan



Multicast data
S2 sends directly to R3, which is the root of the source-based tree
S2
G2
R9
R10
R2
R3
R4
R1
S1
R5
R6
R7
R8
G1
G3

https://ftp.unpad.ac.id/orari/library/library-ref-eng/ref-eng-3/network/multicast.pdf

Switches are at Layer2, IP is at Layer 3

Technically you don't need an IP address in your LAN

Beyond your LAN:
Multicast works by creating a multicast tree, state Explosion

Broadcast – traffic explosion

Multicast TCP vs UDP

- Mentioned broadcasting on TCP a challenge. Is it just as much of a challenge on UDP? Are there good set ways to do this?
  - Problems with IP remain
  - UDP is unreliable
  - Still does not scale over the Internet

Other problems with Multicast

- State explosion
- Access controls, including group creation and membership
- Security. Potential for DDoS, no authentication
- Address allocation – anyone can send anything anywhere
- Congestion control and more

- These are excellent questions for this class!
- Suggested read:
  C. Diot, B. N. Levine, B. Lyles, H. Kassem and D. Balensiefen, "Deployment issues for the IP multicast service and architecture," in IEEE Network, vol. 14, no. 1, pp. 78-88, Jan.-Feb. 2000, doi: 10.1109/65.81917

End-to-End Argument – Saltzer et. al.

Why should anyone care?

When designing a computer system (software, network, hybrid), you need to make a conscious decision where to include certain functionality.

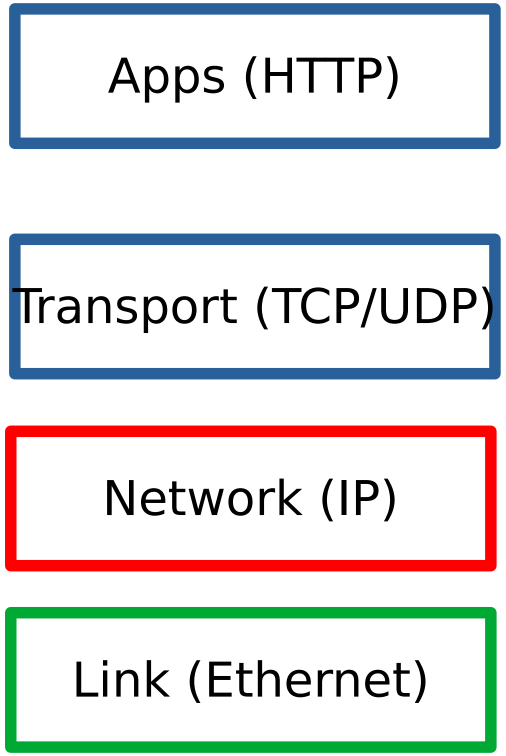
First sentence of the paper “Choosing the proper boundaries between functions is perhaps the primary activity of the computer system designer.”

"This paper discusses one class of function placement argument that has been used for many years with neither explicit recognition nor much conviction."

Writing tip - Mention your goals early and concisely

What is this paper about?

- We learned about network layers.
- When you are developing networked applications, at which layer would you put various functions ?
- This paper provides some guideline about that.



Apps (HTTP)

Transport (TCP/UDP)

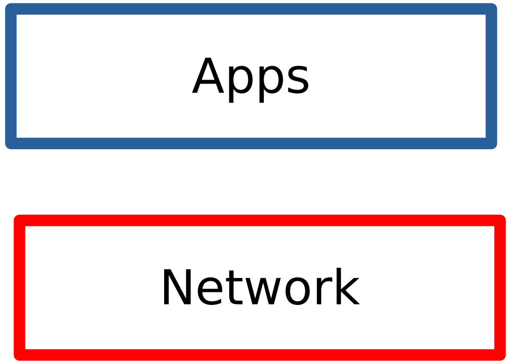
Network (IP)

Link (Ethernet)

Should you treat the network as a Black-Box?

- You have a bunch of requirements for your application
  - Security, error checking, reliable delivery, in-order message delivery

- Should you implement them in the network? Should you implement them in the application? Doing it in both layers? Is more better?
  - What are the trade-offs?



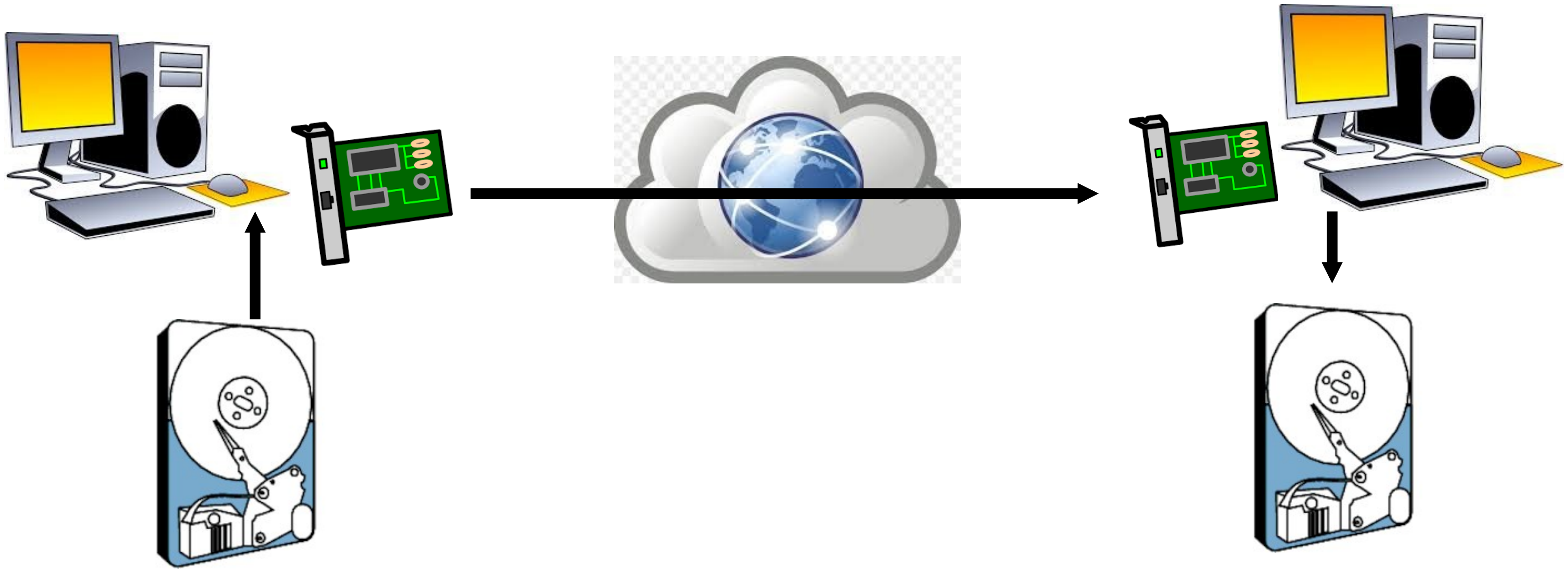
Apps

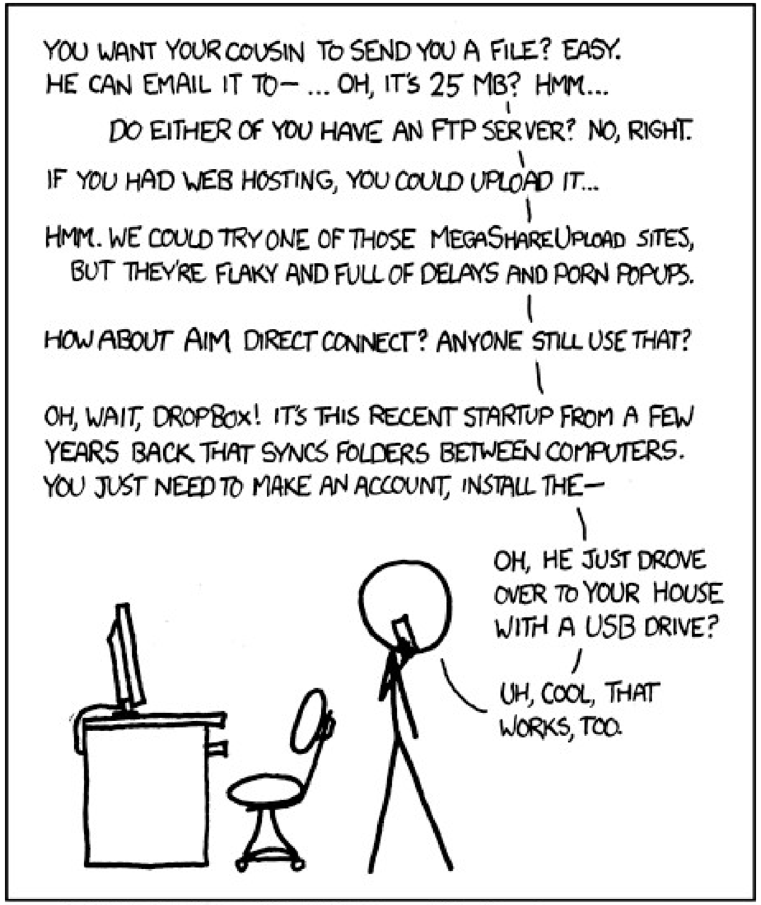
Network

File transfer

Oldest use case on the Internet – still happening

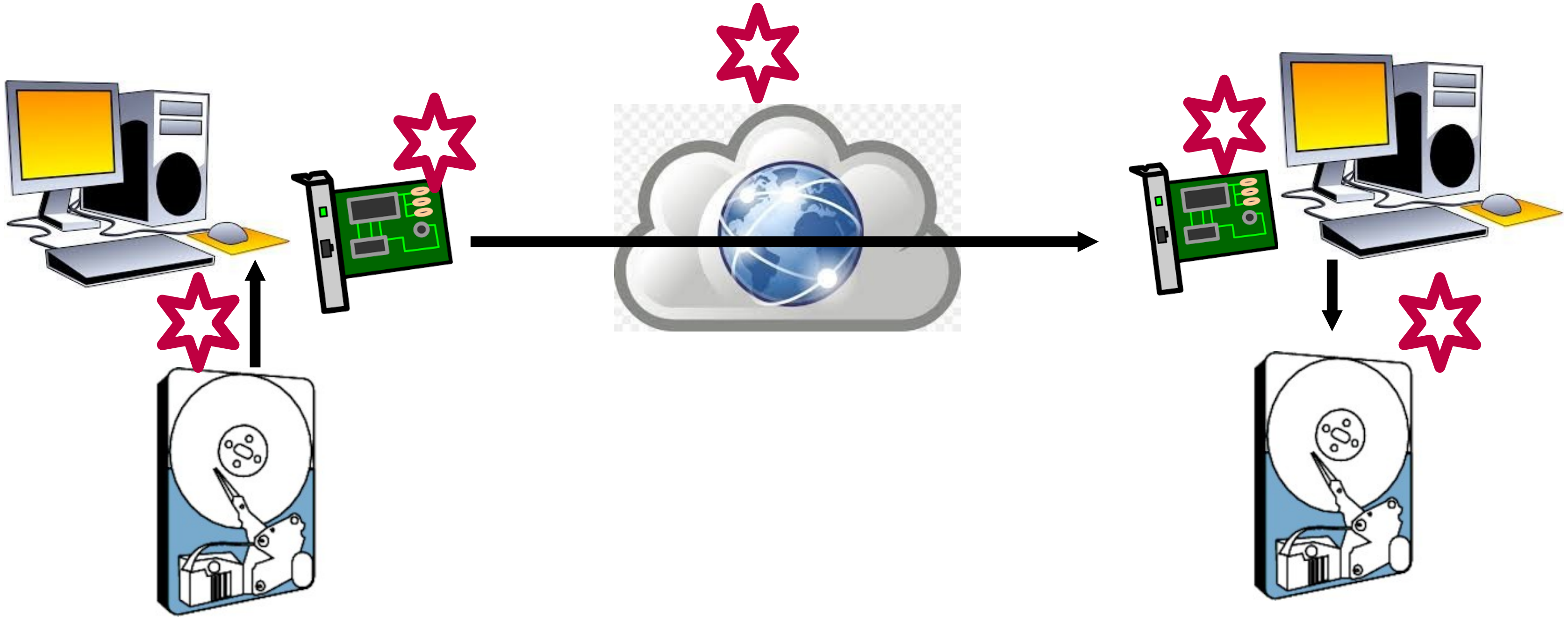
Goal – send a file from Host A to Host B over the network.




YOU WANT YOUR COUSIN TO SEND YOU A FILE? EASY. HE CAN EMAIL IT TO– ... OH, IT'S 25 MB? HMM...
DO EITHER OF YOU HAVE AN FTP SERVER? NO, RIGHT.
IF YOU HAD WEB HOSTING, YOU COULD UPLOAD IT...
HMM. WE COULD TRY ONE OF THOSE MEGASHAREUPLOAD SITES, BUT THEY'RE FLAKY AND FULL OF DELAYS AND PORN POPUPS.
HOW ABOUT AIM DIRECT CONNECT? ANYONE STILL USE THAT?
OH, WAIT, DROPBOX! IT'S THIS RECENT STARTUP FROM A FEW YEARS BACK THAT SYNCS FOLDERS BETWEEN COMPUTERS. YOU JUST NEED TO MAKE AN ACCOUNT, INSTALL THE–
OH, HE JUST DROVE OVER TO YOUR HOUSE WITH A USB DRIVE?
UH, COOL, THAT WORKS, TOO.

I LIKE HOW WE'VE HAD THE INTERNET FOR DECADES, YET "SENDING FILES" IS SOMETHING EARLY ADOPTERS ARE STILL FIGURING OUT HOW TO DO.

File transfer – Possible failures



Disk

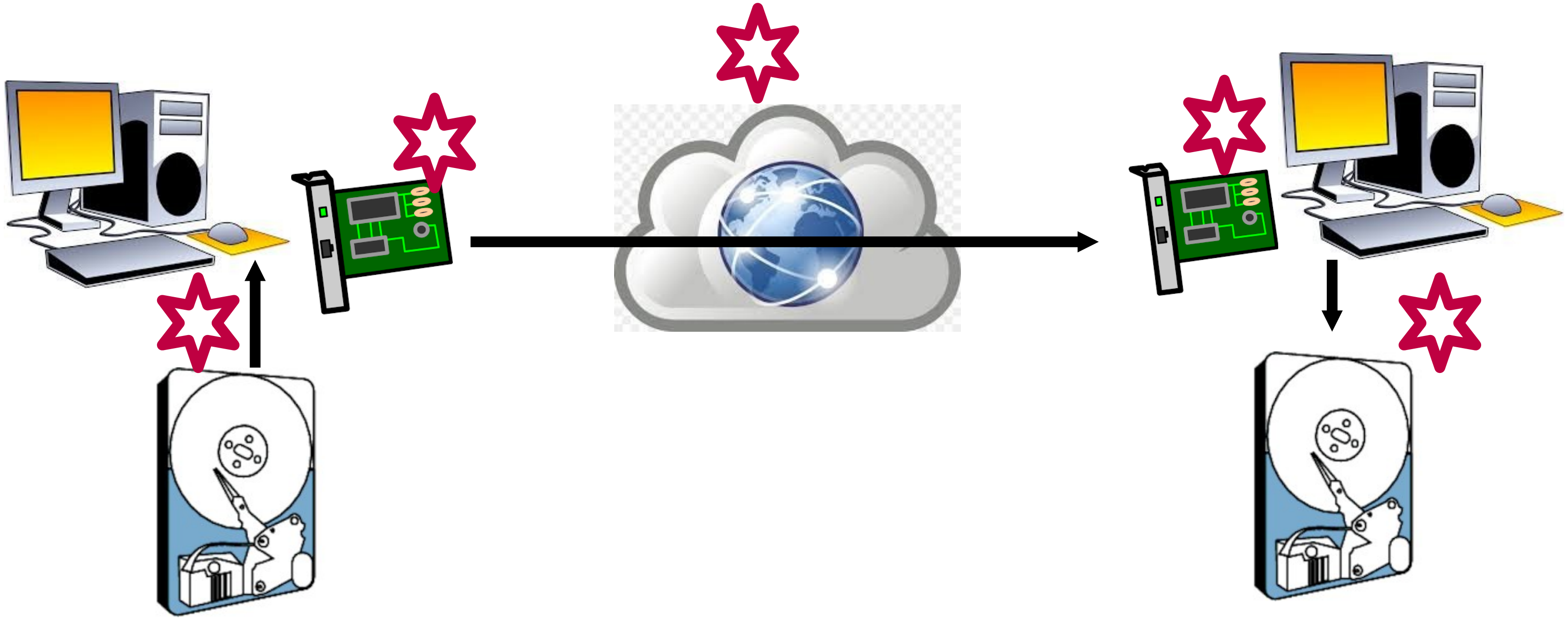
Software buffer

Memory

Communication system

Hosts may crash

File transfer – Where to fix?

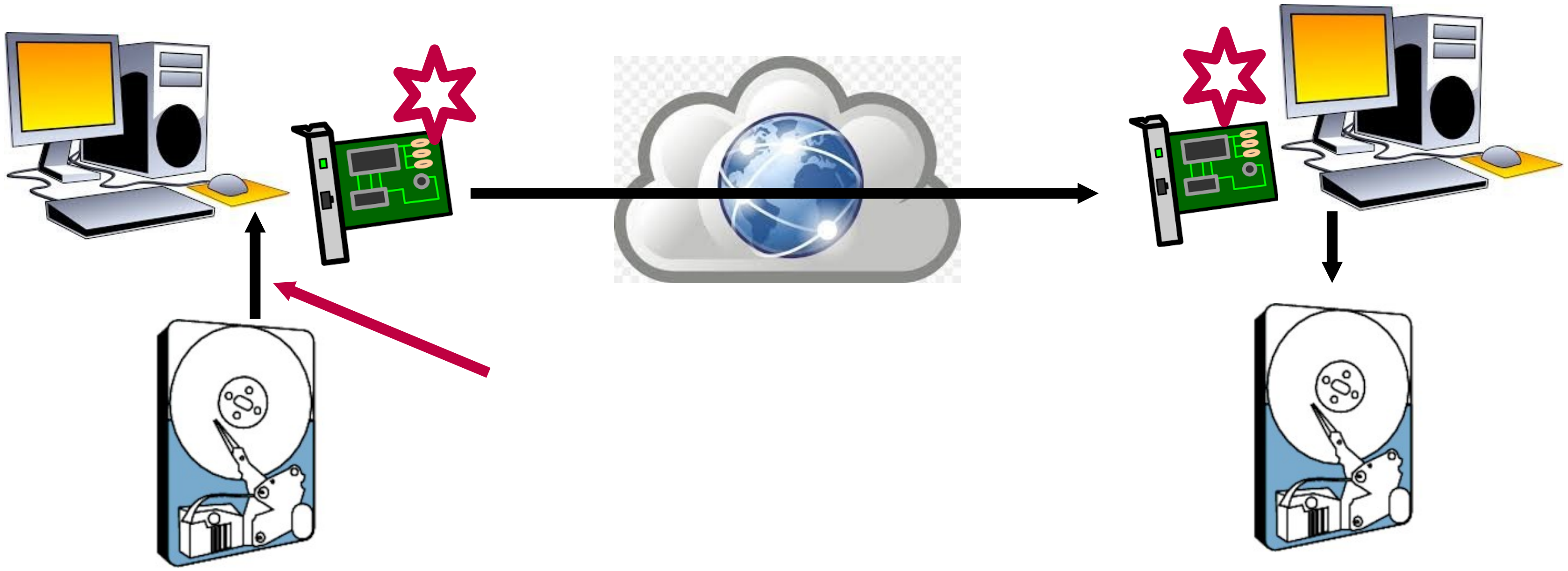


At each step
+ More checks
- Correctness is hard to achive
- Uneconomical
- Errors might still happen

End-to-end check and retry
+ Applications’ control
- Might not work well for some Applications
-example?

Why not do both?
- Does not add value
- Does add complexity

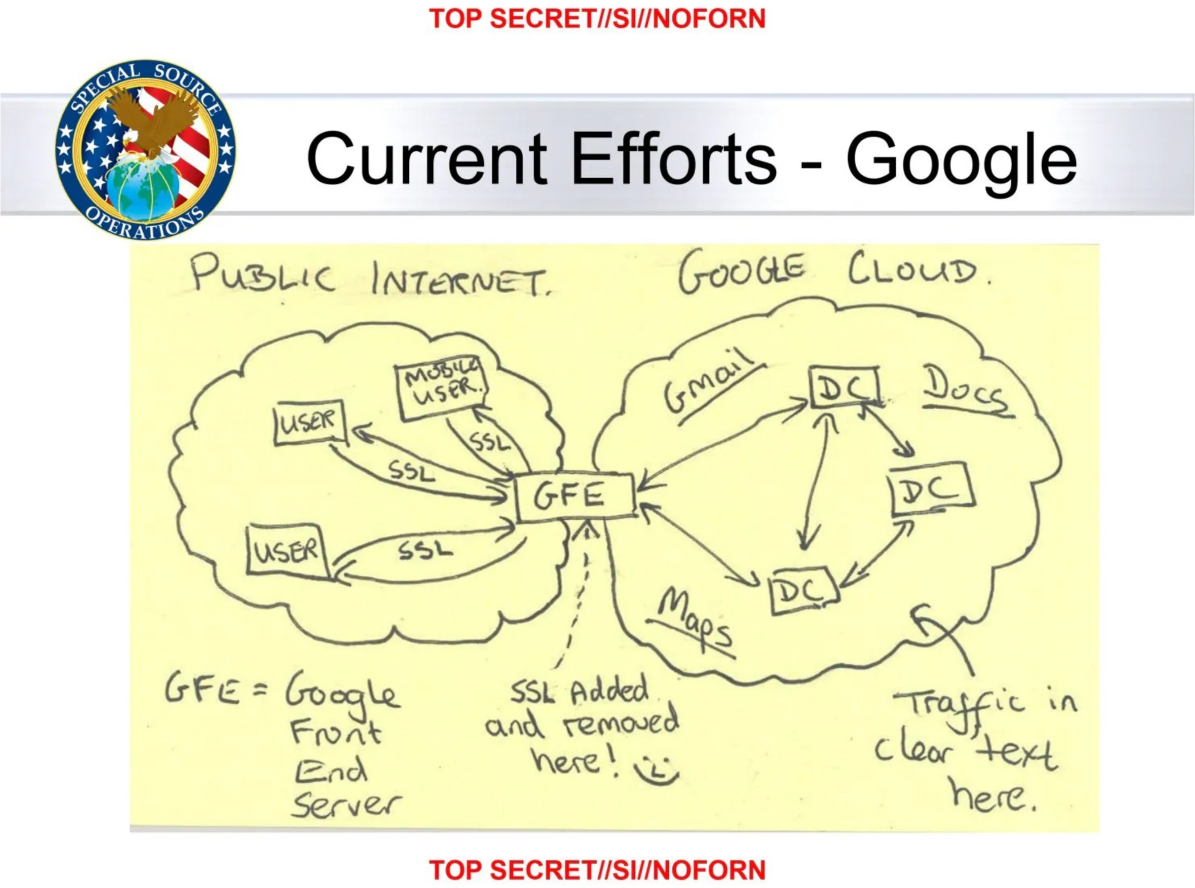
Another example - Security



Secure host to host communication

- A malware on your laptop can read everything
- The applications must still check the authenticity of the data

Another example – NSA breaches Google



TOP SECRET//SI//NOFORN

Current Efforts - Google

PUBLIC INTERNET.
GOOGLE CLOUD.
MOBILE USER.
USER
USER
SSL
SSL
SSL
GFE
Gmail
DC
Docs
DC
DC
Maps
GFE = Google Front End Server
SSL Added and removed here! :)
Traffic in clear text here.

TOP SECRET//SI//NOFORN

Secure host to host communication

- No SSL on data center links
- Has since been fixed
- Good argument for end-to-end encryption

Alternative – the application performs encryption

- Application performs all encryption and decryption
  - Works beautifuly!

- No need for the network to perform any additional tasks

- BUT, if you are concerted about data exfiltration by a rouge application
  - You might want to have some sort of network encryption
  - Can be cheap – reuse keys between end-points, use same keys for all traffic and so on.
  - Exactly what SSL/TLS does!

End-to-End Argument

Thus the argument:

In order to achieve careful file transfer, the application program that performs the transfer must supply a file-transfer-specific, end-to-end reliability guarantee – in this case, a checksum to detect failures and a retry/commit plan.

For the data communication system to go out of its way to be extraordinarily reliable does not reduce the burden on the application program to ensure reliability.

But the communication system should not be a black box

Inherent dangers of a black box -

- Example – file transfer protocol at MIT
  - Assumed the network is reliable
  - Ended up matching files manually due to a corrupt gateway

- Some low level assurance is needed
  - Big file transfers – petabyte files
  - Correcting all errors are the application are very expensive!
  - Allow the network to put “some” efforts
    - But don’t go out of the way to solve every problem

  - Engineering trade-off rather than correctness

Don’t try to anticipate application’s needs

Example – ARPANET ACK messages - "Request For Next Message" (RFNM)

Every time the network delivers a message, the other end point returns an RFNM message.

Useful for the network – what if your application crashed?

You need an acknowledgment from the application itself, not the network!

Occam’s Razor

- Proposed by the British philosopher – William of Ockham
- Also called the “law of parsimony” -
  `"entities should not be multiplied beyond necessity"`
- `The razor cuts away unnecessary things`
- Do not duplicate work at multiple layers unless it benefits you.
- Useful whenever you are designing a system

How do I identify the end points?

- Good question - depends on your application!

- If you are on a phone call or watching a video, you can rely on the listener or viewer to act as the end
  - Ask to repeat
  - Refresh the video or ignore momentary disruptions

- If you are archiving data for long term
  - Your software is your end-point, you must ensure correctness!

Conclusions

- End-to-end argument is a principle that helps the design of networked systems

- Keep Occam’s razor in mind – start simple, and think where functionality is best implemented

- When designing a network, do not try to help the users too much
  - You don’t know what they want
  - They don’t know what they want

- When designing an application, do not treat the network as a black box

- Some functions are obvious function of a particular layer
  - Err

Deliverable before the next class

- Fill out the survey - https://tntech.co1.qualtrics.com/jfe/preview/SV_bjT8BpnRABUxZ7o

- Read the paper “The Design Philosophy of the DARPA Internet Protocols”.
  - We will discuss in the next class.

- Set up a time to meet with me about your presentation
  - Jordan
  - Luke

- If you want to meet during my office hours:
  - Please email me or ping me on Slack.
  - David