CS 114:
Network Security

Lecture 1

Prof. Daniel Votipka
Spring 2023

(some slides courtesy of Prof. Micah Sherr)

COVID Logistics

- Lectures will streamed live on Zoom, recorded, and published to course Box folder
- Office hours will be available via Zoom

Video Recording and FERPA

- I am recording all the lectures for students who cannot attend due to whatever reason
  - Your name, voice, and other information may also be recorded as part of that
  - Per FERPA, you have a right to privacy in your student record

- Your consent to be video recorded is included as part of the syllabus:
  - *Course lectures may be audio/video recorded and made available to other students in this course. As part of your participation in this course, you may be recorded. If you do not wish to be recorded, please contact your instructor the first week of class (or as soon as you enroll in the course, whichever is latest) to discuss alternative arrangements*.

- Videos will not be posted publicly and only available to other students in this class!
  - Hosted on Box with password protection
  - Links to videos posted to Piazza only
  - **You may not reshare videos beyond the class otherwise you are violating FERPA**

Start Recording

The Plan

- Introductions
- Course Overview
- Logistics
- Intro to Networking (maybe)

Introductions

- Dan Votipka
  - Assistant Professor; Co-Director of TSP
    - 2021-Present
  - Previously:
    - National Security Agency from 2012-2016 (4 years)
  - Ph.D. University of Maryland
  - M.S. Carnegie Mellon University
  - B.S. Illinois Institute of Technology

https://tsp.cs.tufts.edu/

- Research Interests
  - Human factors in professional security work
  - Usable security and privacy, more generally
  - Usable security and privacy, more generally

professional security work
reverse engineering, network defense, etc.

Who are you?

- Name
- Year
- Program
- What's the most interesting topic in security? Why?
- Fun fact about yourself

Why does it matter?

- Enterprise Security
  - National Security
  - Financial Sector
  - Industrial Control Systems
- Personal Security
  - Identity Theft
  - Privacy

Some bad news

We're terrible at designing secure systems.

Designing secure systems is difficult.

Fundamental asymmetry between attacker and defender

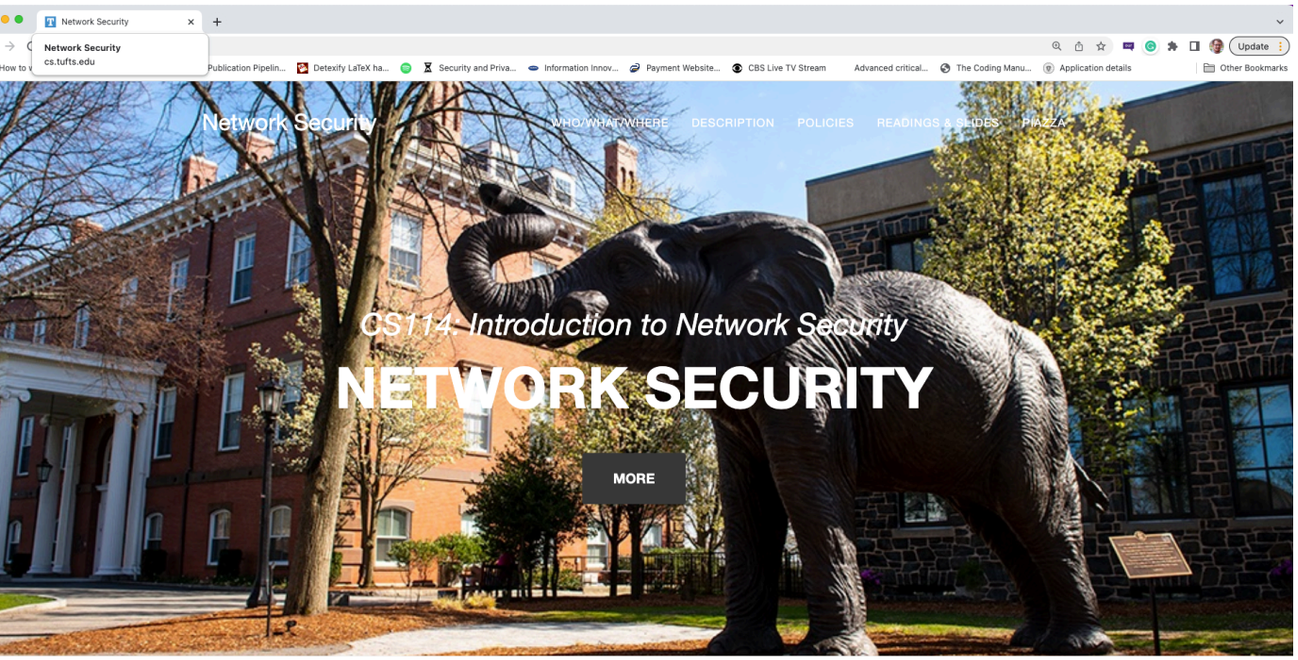
Functionality is easy to measure, but…

Airplane works

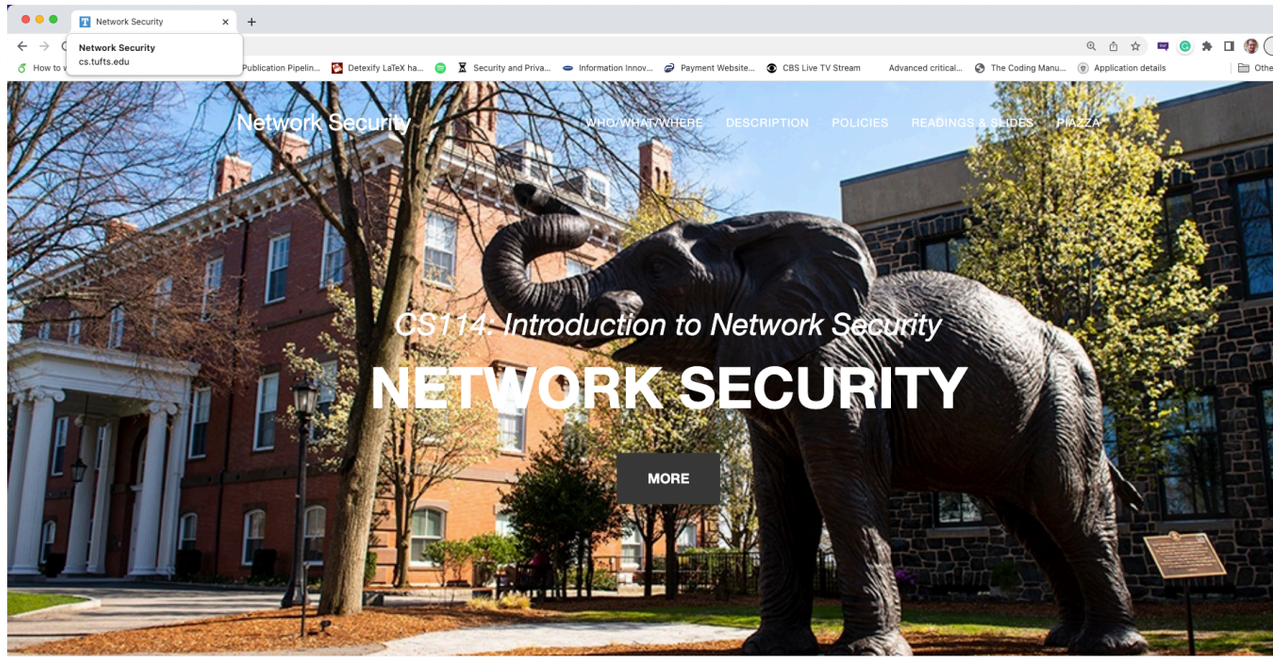
Airplane doesn't work

…*security* is almost impossible to measure

Web browser 0wned



Web browser not 0wned



Some good news

Computer security is a growth area.

Awesome

Course Goals

- Learn how to design more robust systems
- Learn how to protect against attacks
- Think like the bad actor, behave like the good actor

This introductory course will impart a broad understanding of the underpinnings of security techniques, security best practices, and computer security research. The course should help students to understand the mindsets of attackers (the bad guys and gals who do malicious things on the network) and system designers and defenders (the good guys and gals who try to stop the attackers). The course should prepare students to understand and assess security threats, become familiar with security engineering best practices, and write better software, protocols, and systems.

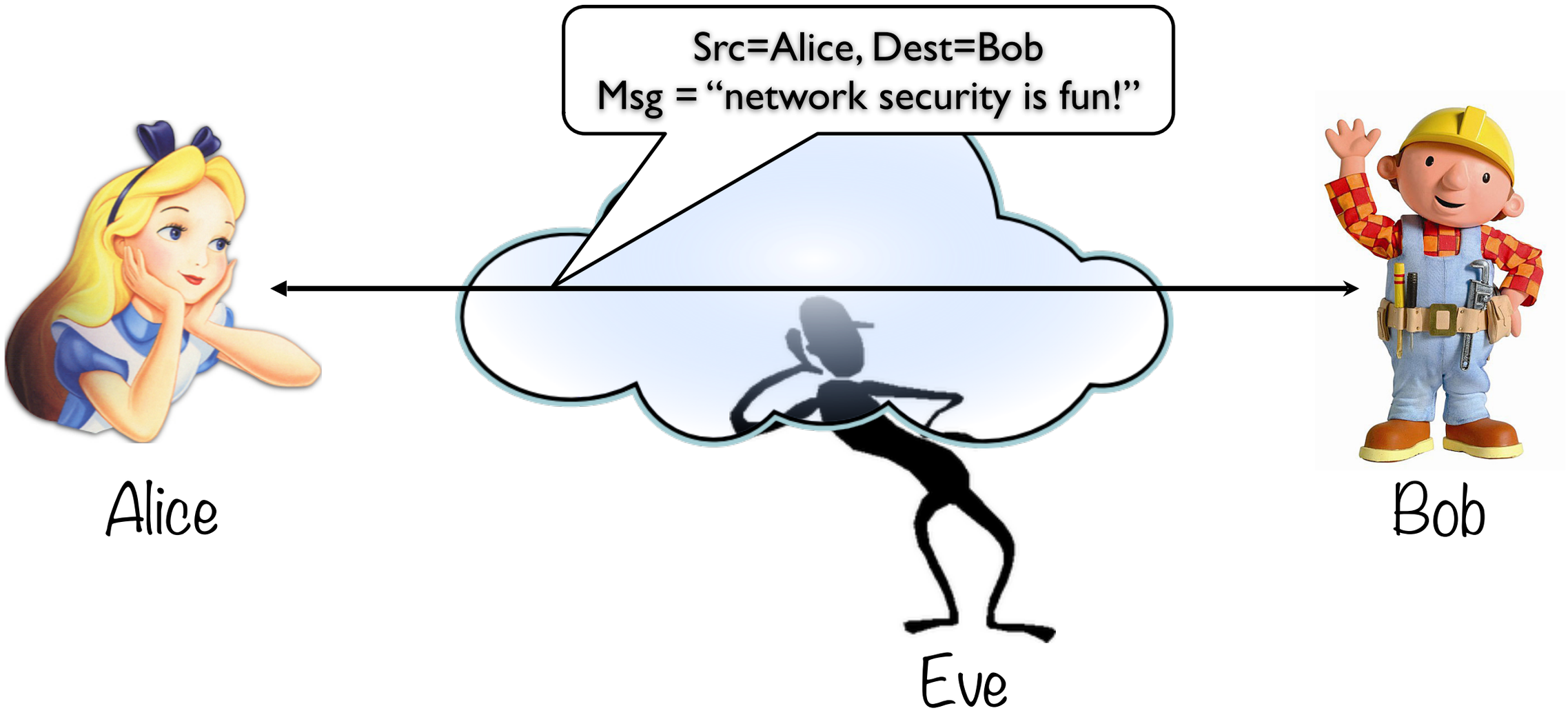
Non-goals

- Familiarization with latest tools
- Professional security certification

Course Topics

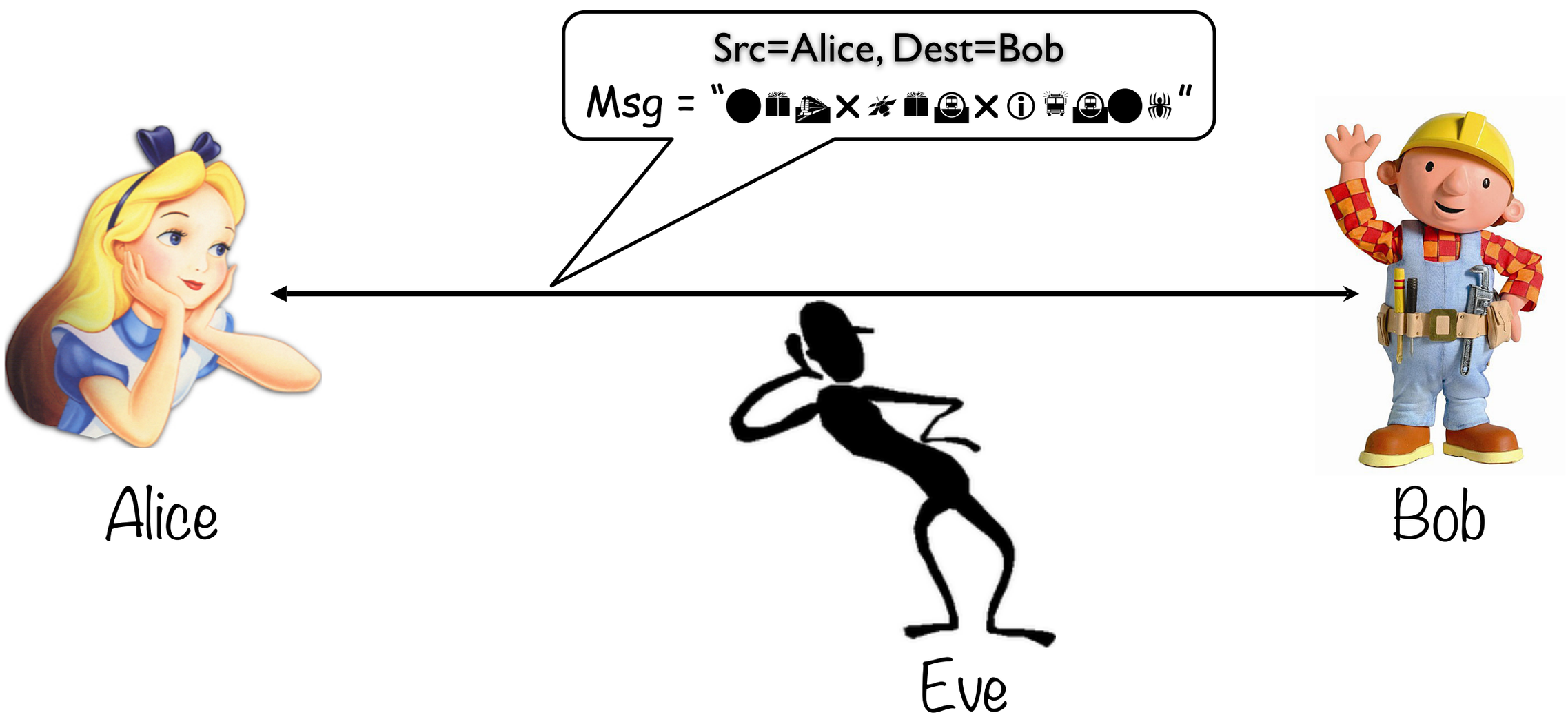
- High-level Topics:
  - Review of networking (this isn't a networking course)
  - Review of crypto (this isn't a crypto course either)
  - Network malware
  - Network defenses
  - Network privacy / anonymity
  - Web security
  - Social engineering
  - Usable Security
- Check the website!
- Assignments:
  - HW 0 - Autograding Intro
  - HW 1 - Secure IM
  - HW 2 - Port Scanning and Detection
  - HW 3 - CTF (Extra Credit)
- **Note:** I reserve the right to adapt the schedule throughout the semester; *I will give at least one week's notice of any changes*

Meet the players.



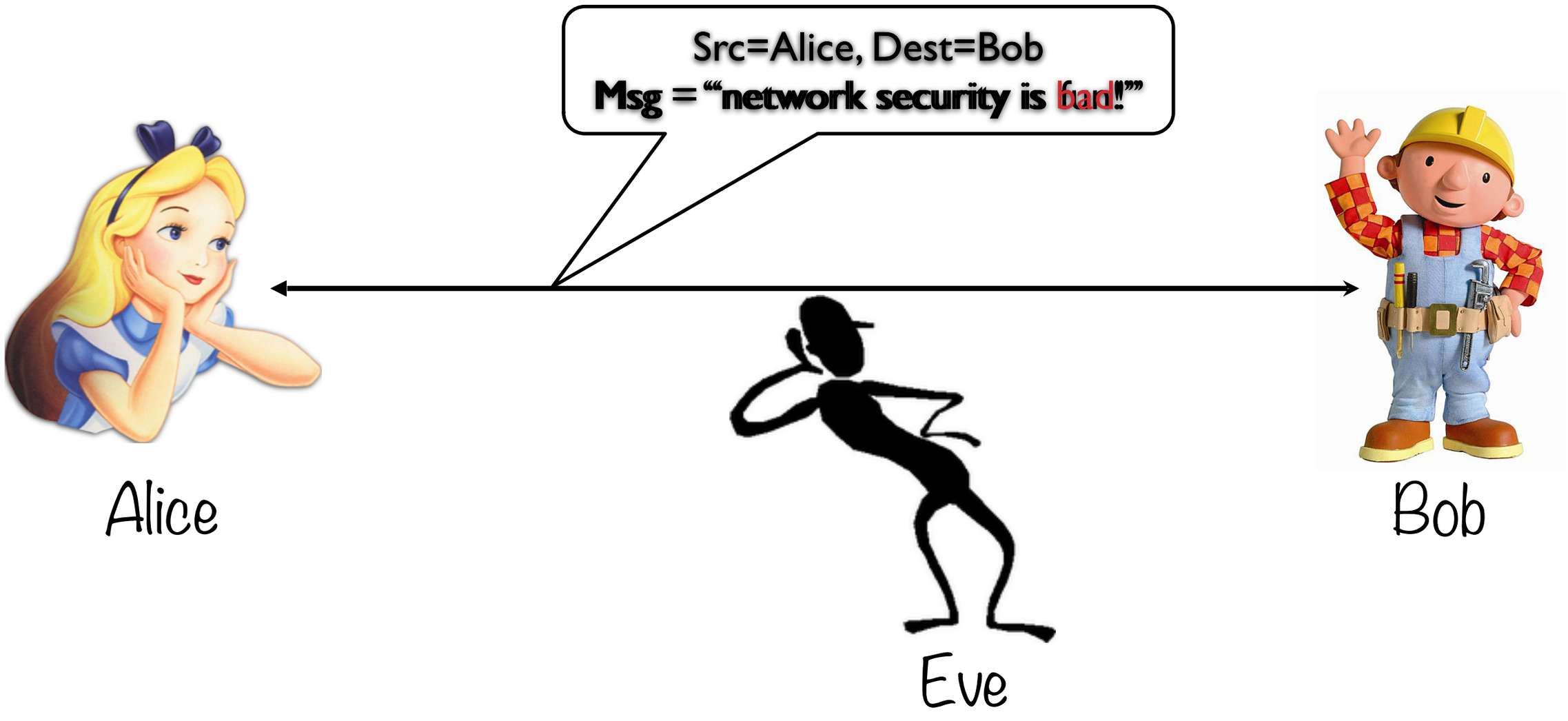
Let's look at some potentially desirable properties of a secure network system...

Confidentiality



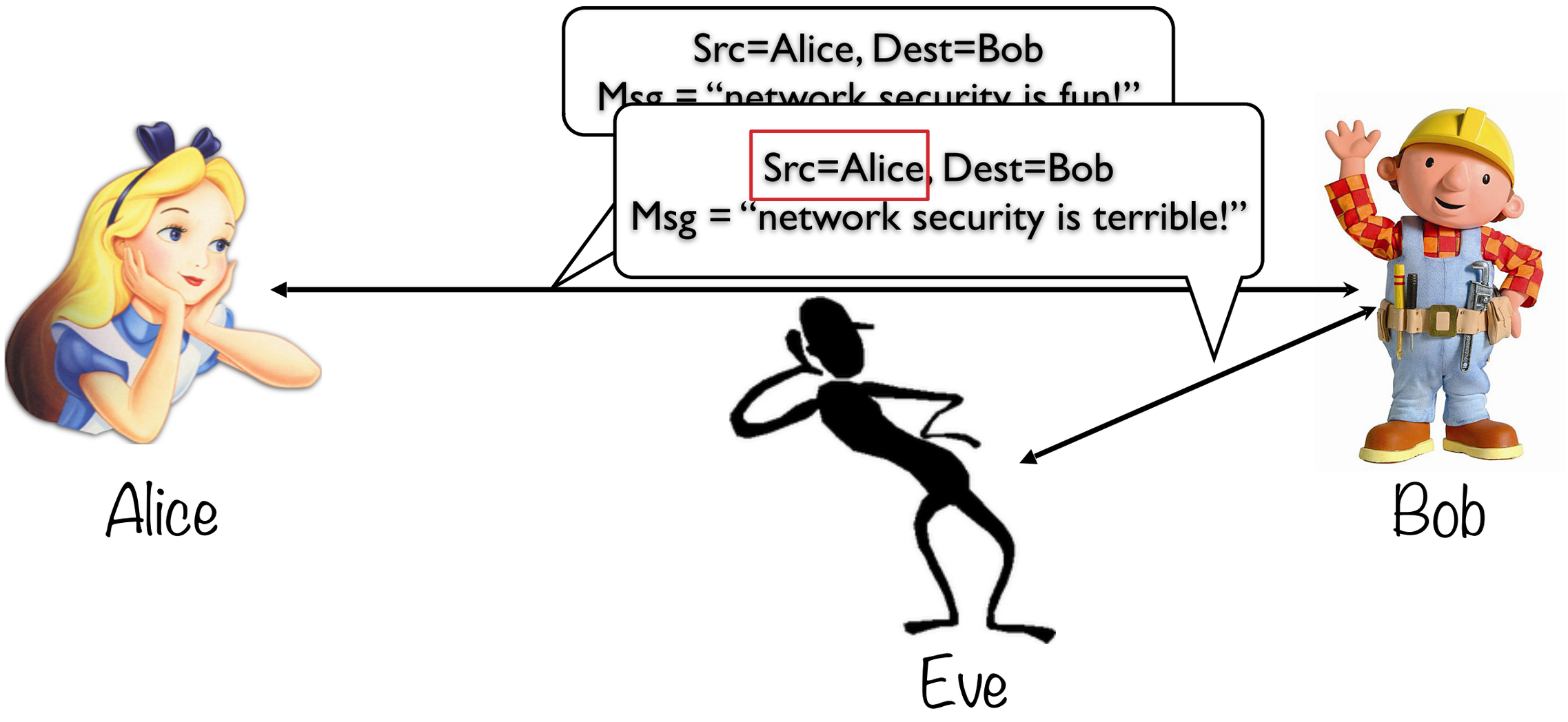
Alice and Bob want to communicate privately, preventing Eve from learning the contents of their communication

Integrity



Bob wants to verify that the message hasn't been altered in transit.

Authentication



Bob wants to verify that the message is actually from Alice.

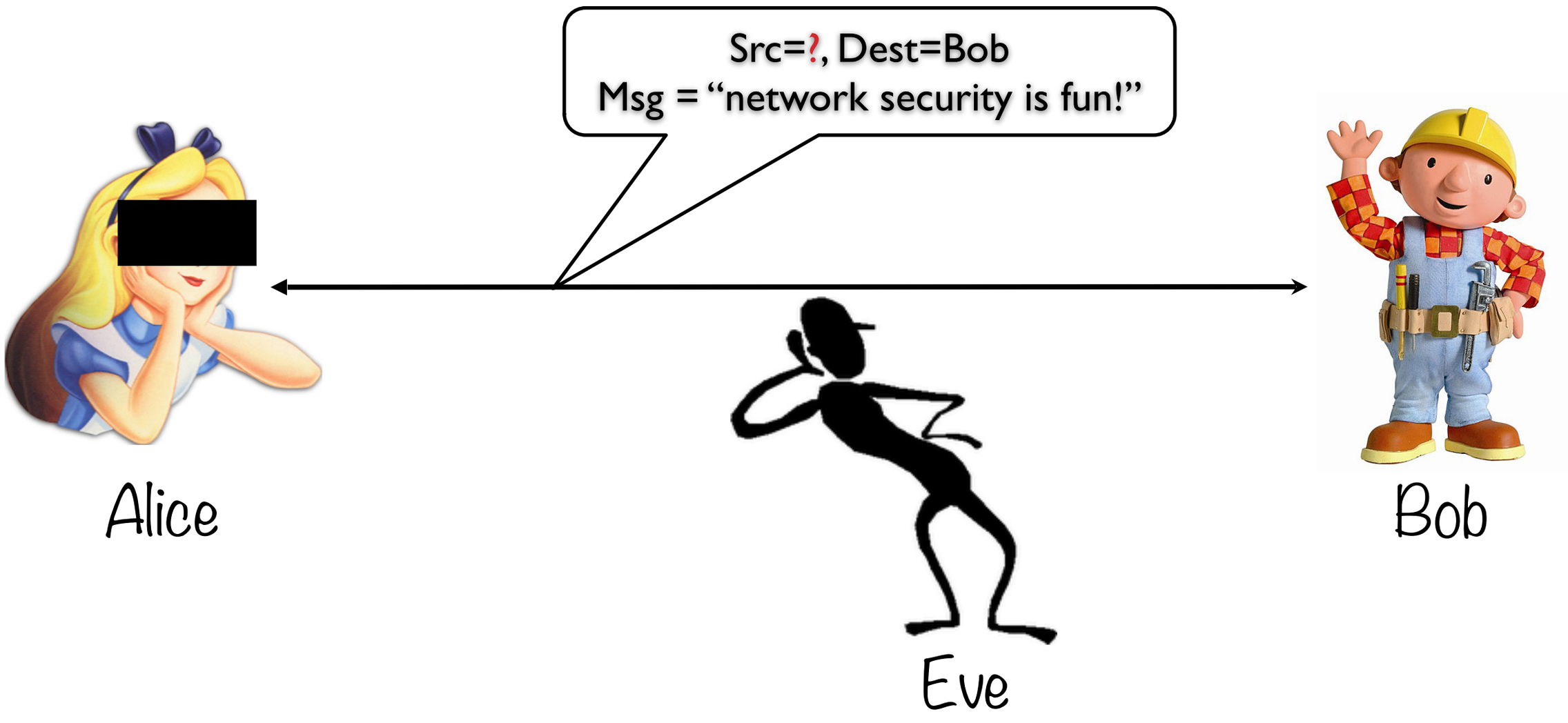
Client authentication

Alice wants to prove her identity to the service.

Server authentication

The service wants to prove its identity to Alice.

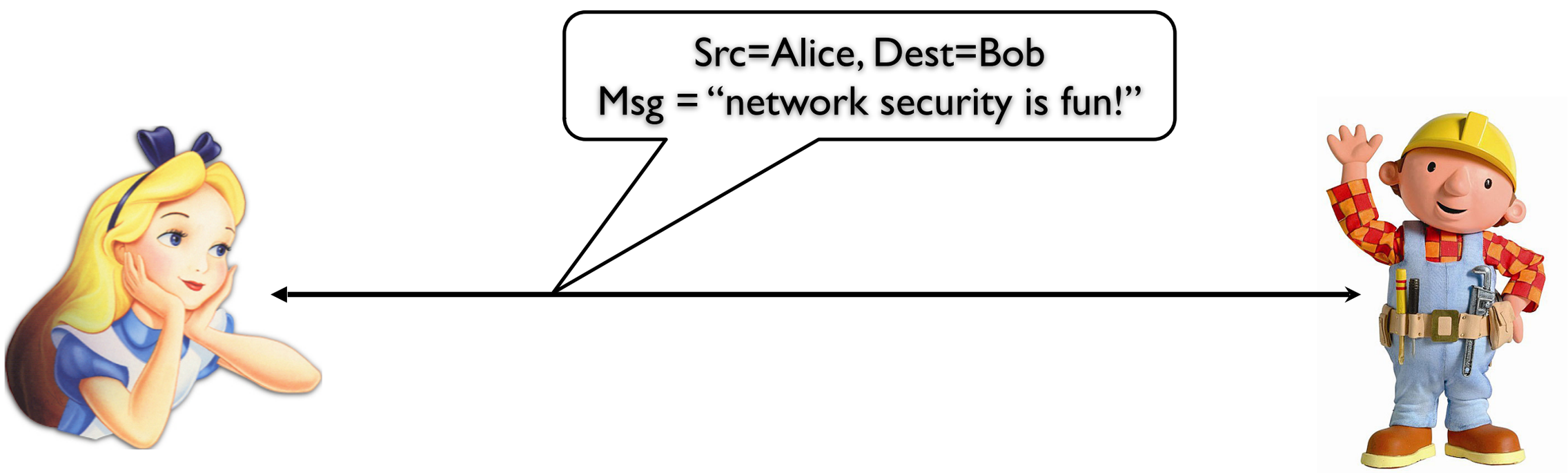
Anonymous communication



Alice wants to communicate anonymously to Bob (sender anonymity)

Warning:
crypto != security

Usability



Alice and Bob should be able to quickly learn how to use the system naturally with few errors.

Course policies, expectations, and other fun bureaucratic goodness

This is the most important slide in this deck!

Course website:

https://www.cs.tufts.edu/cs/114/2023S

Prerequisites

- CS 15 (or equivalent)
- You will build stuff. I expect you to:
  - know how to code
  - **learn** how to code in Python
    (you'll thank me later)
  - be(come) comfortable with Linux/UNIX

You have amazing TAs!

Carson Powers

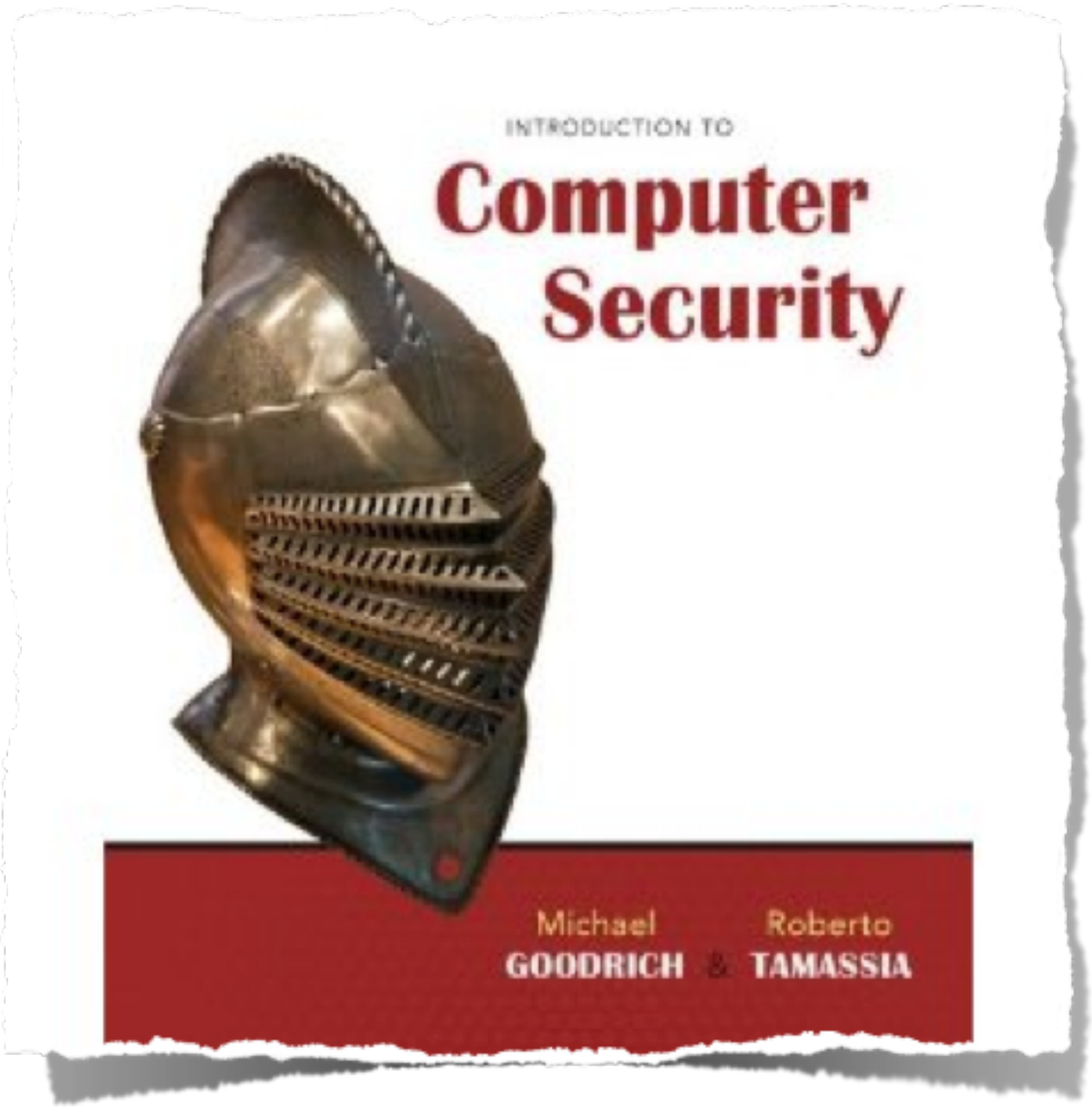
TBD

Office Hours

- Prof. Votipka
  - Tu/Th from 10:00am-11:00am and by appointment
  - Location: JCC, rm 361
- Carson (TA)
  - M/W from 11:00am-12:00pm
  - Location: JCC, rm 359

Office hours can also be joined via Zoom (see website for links)

Textbook



- This course has a strongly recommended textbook:
  - Introduction to Computer Security by Goodrich and Tamassia
- Also helpful:
  - Handbook of Applied Cryptography by Menezes, Oorschot, and Vanstone (available online)
  - Security Engineering by Ross Anderson (first edition available online)
  - Applied Cryptography by Bruce Schneier

Things that are not your textbook

WIKIPEDIA
The Free Encyclopedia

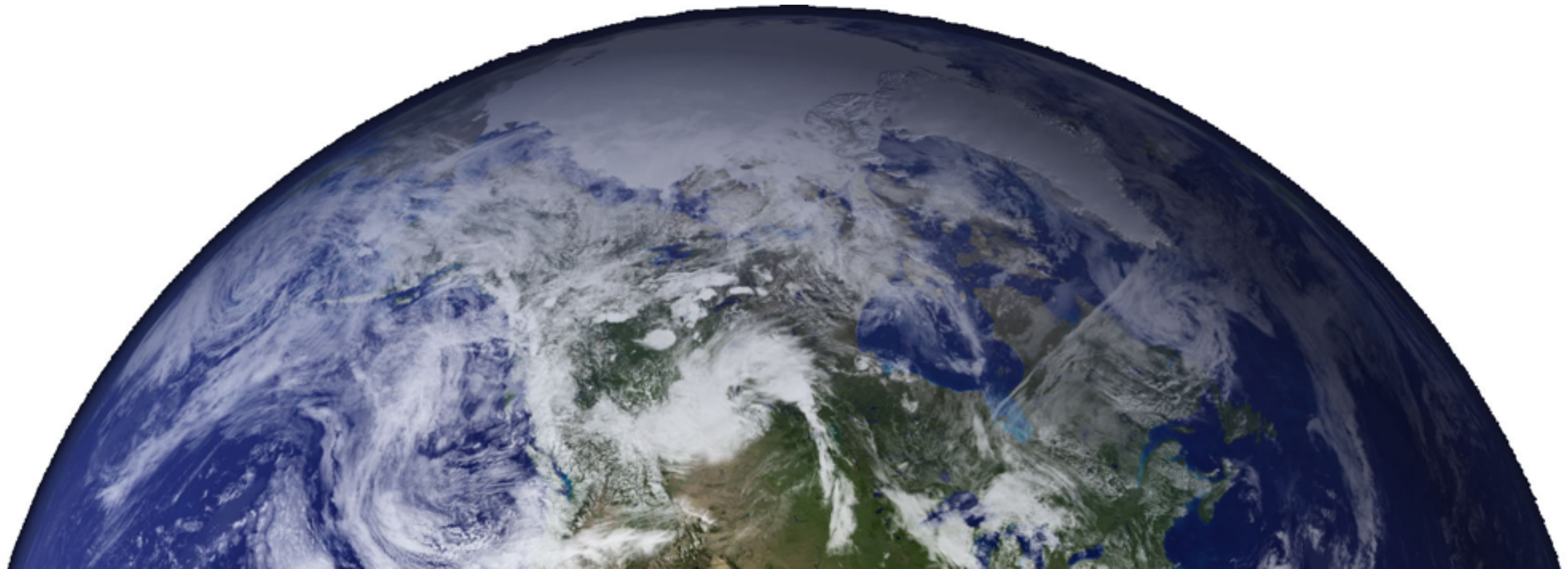
Google

Readings

- We'll be reading some seminal research papers (see syllabus)
- Do the readings.

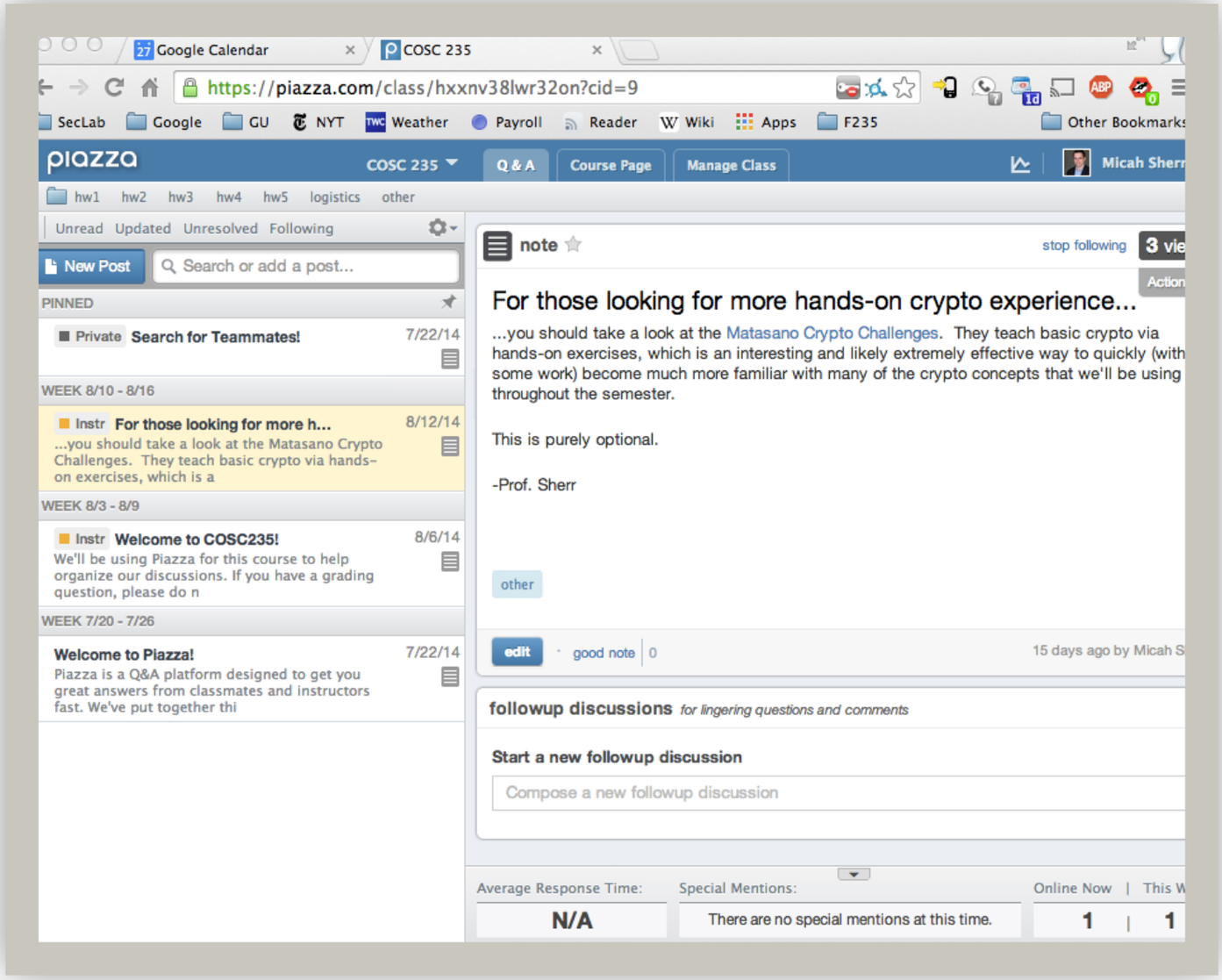
Lecture notes

- Slides will be released on the course web page after each class.
- I like trees.



Online Course Discussion

- **Extensive** class discussions and announcements via Piazza
  - Be prepared to receive many emails
- You are expected to read each and every posting
- You are expected to participate
- See course webpage for Piazza URL.



Online Course Discussion

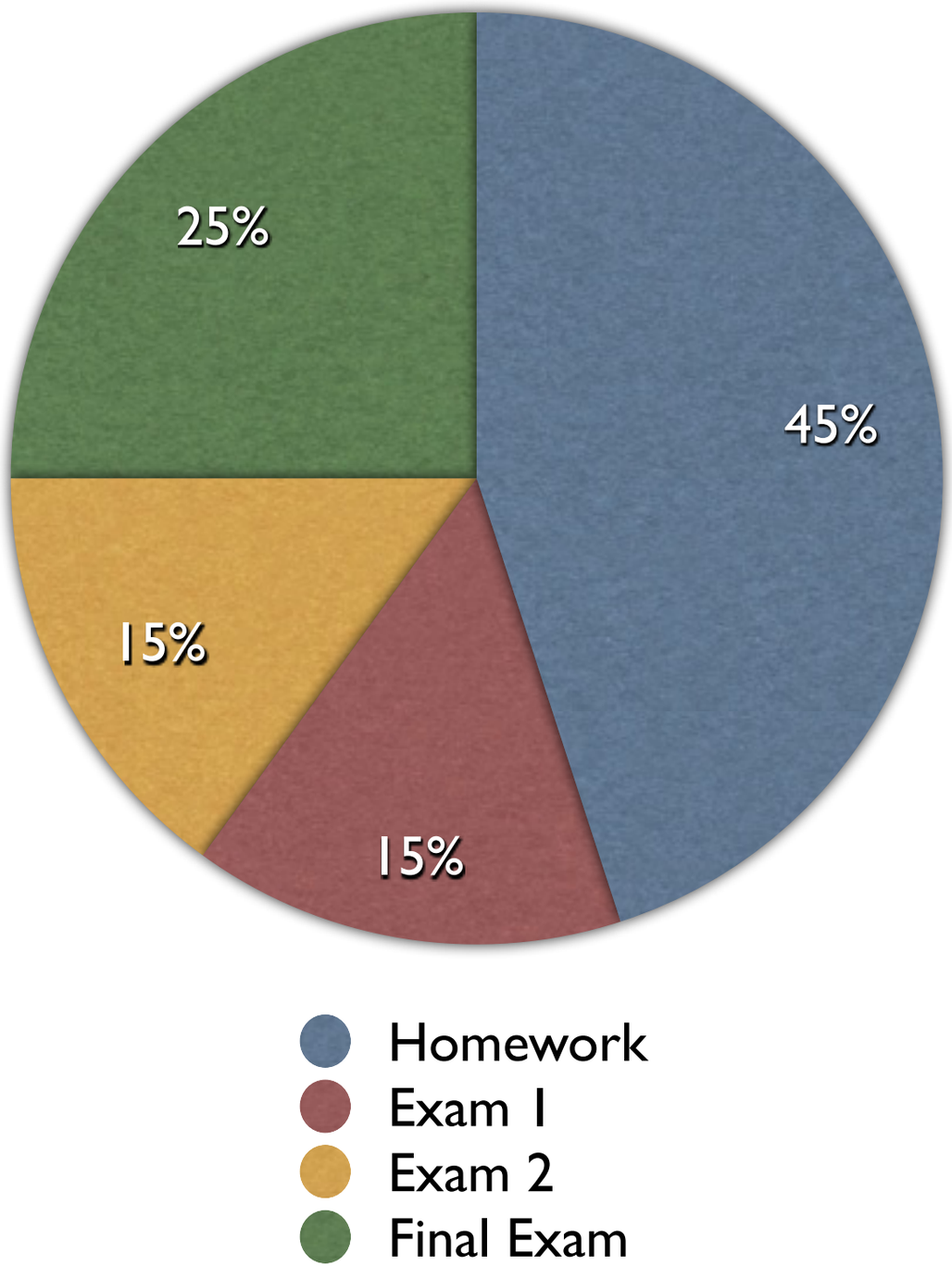
- Post to Piazza if…
  - …you have a question about the class subject matter (slides, lectures, etc.)
  - …you need a clarification on a homework
  - …you have a general question about network security
  - …you have a question regarding a class policy
- If you send any of the above to me directly, I'll ask you to post it on Piazza
- *Don't*:
  - Give away solutions to assignments
  - Start flamewars
- *Do* be respectful of others

Emailing

- It's really best not to email me. Emails get lost. Piazza posts stay there until I actually resolve them.
- Send a private Piazza post if…
  - …you have a grading issue
  - …you need to ask a question that would reveal a partial/ complete solution to a homework problem

- Email me directly (daniel.votipka@tufts.edu) or meet with me if…
  - …you have a personal issue that you don't want to share with the TAs

Grading



- There will likely be 3 homework assignments
  - Some are question-based
  - Most are programming-based
- 25% penalty for late homeworks within 24 hrs
- 100% penalty for late homeworks after 24 hrs
- 3 free late days for homeworks
- 1 extra credit assignment

Autograding

- We'll be using autograding for some (but not all) projects
- This isn't (just) because we are lazy.
- Autograding = automatic grading = immediate feedback!
  - You may submit your homework before it is due and get feedback.
  - You can resubmit a revised homework, and get feedback on that revision.
  - Rinse and repeat.
- Autograding should reduce grading surprises.

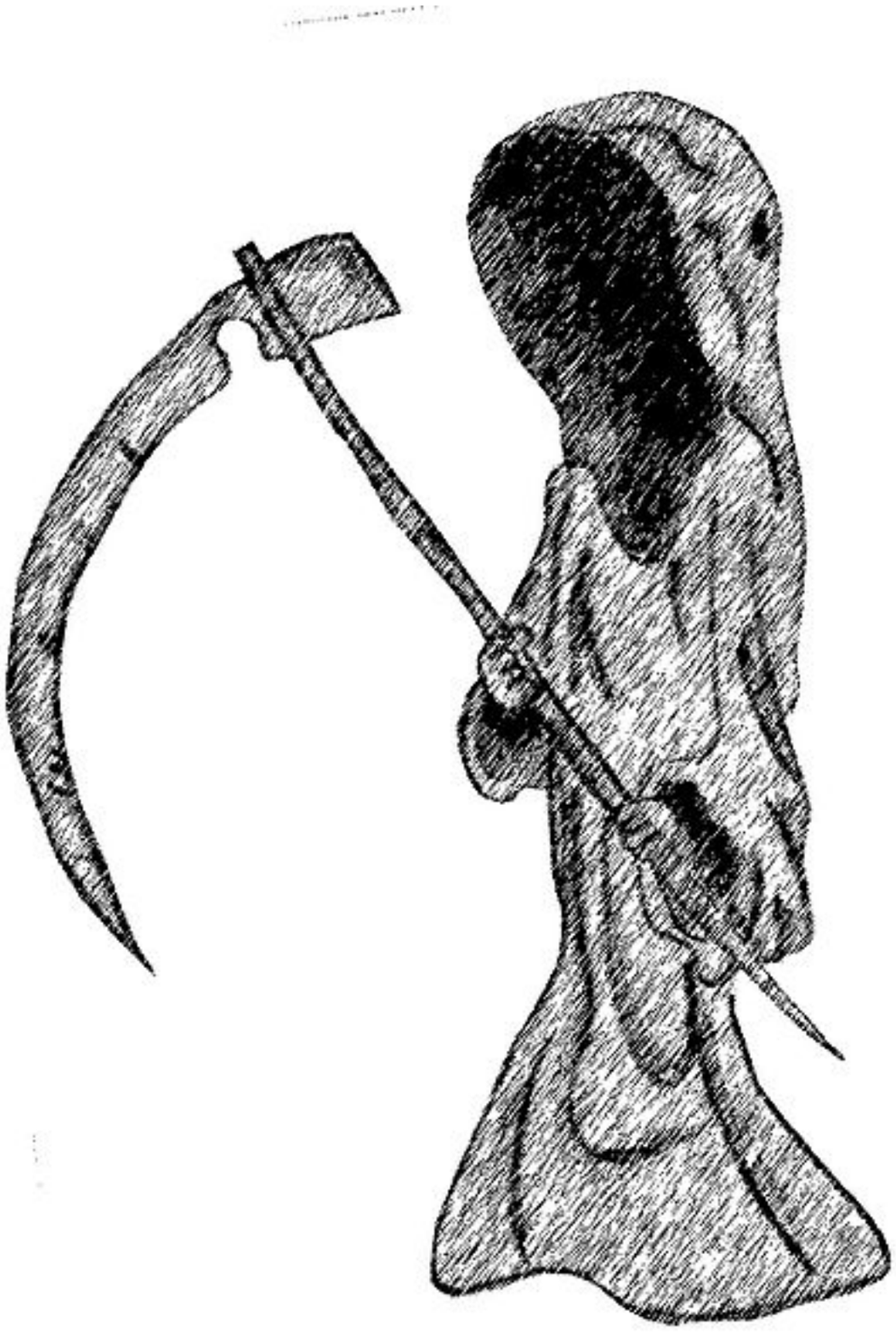
Homework 0

- You'll be writing hello world, in Python (version 3)
- This isn't a test of your coding skills. It's a test of following directions and using the autograder (Gradescope).
- Due Thursday, January 26th at 11:59pm.

Other Policies

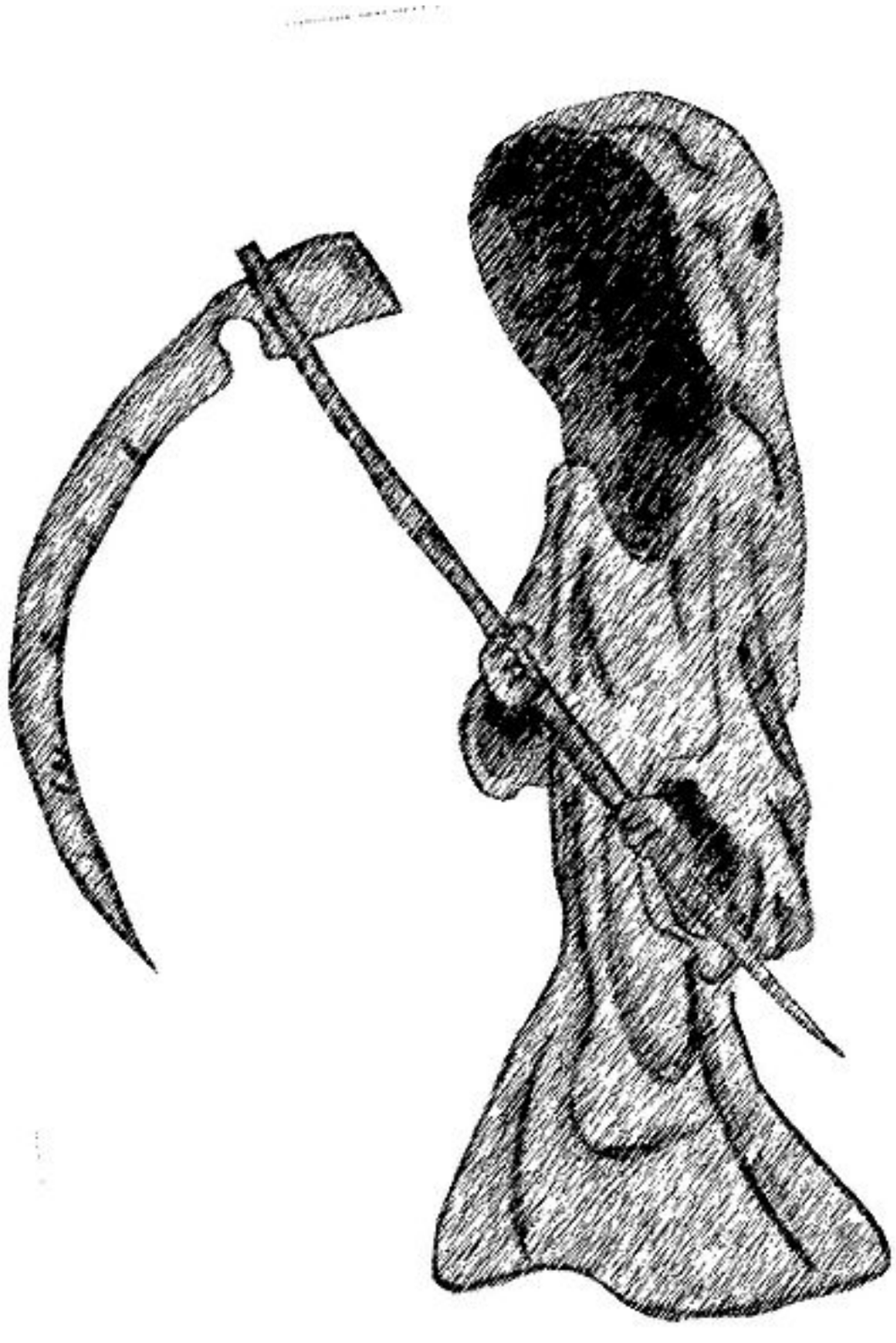
- Please turn off cell phones during class.
- I will do my best to respond to emails within 24 hours. You will receive faster answers if you post to Piazza
- Students may appeal to the instructor for reconsideration of a grade, but the appeal must be in writing (i.e., email), and must be sent within a week (or the close of the semester, whichever is sooner) of receiving the graded assignment.
- Behave civilly: don't be late for class; don't read newspapers/ blogs/etc. during class; don't solve Sudoku puzzles during class; don't struggle with crossword puzzles during class; respect others' opinions.
- Adhere to good scientific principles and practices, and uphold the Tufts Academic Integrity Policy

Cheating policy



- Cheating is not allowed
- We run tools
- If you cheat, you will probably get caught
- **I REFER ALL ACADEMIC DISHONESTY INCIDENTS WITHOUT EXCEPTION**
- If you are found to be in violation, you will almost certainly get an F on the course (not just for the parts you were caught cheating)
- If you don't cheat and **work hard**, you will always do better than if you cheated

Cheating policy



- Cheating is (but is not limited to):
  - Including source code in your homework submissions that you did not write
  - Basing any part of your source code off of someone else's without prior approval from a member of the teaching staff
  - Working together to solve homework problems
  - More generally, taking credit for something that you did not create

Diversity and Inclusion

- Diversity that students bring to this class is a resource, strength, and benefit. This is particularly true in security!
- I strive to create learning environments that support diversity of thought, perspective and experience, and that honor and respect your identity
- I expect everyone to contribute to a respectful and inclusive class environment
  - This doesn't mean we always have to agree, but instead that we consider other perspectives.
- We will not be perfect, so if something in this class makes you uncomfortable, please let me know
  - Alternatively, you can email the CS Dept. Chair (jeffrey.foster@tufts.edu) or the Office of the Dean of Student Affairs (deanofstudentaffairs@ase.tufts.edu) to provide anonymous feedback.

Course credo:

Think like an attacker,
but behave like a responsible adult.

Tufts' computer usage policies apply to this class.

Network security course !=
permission to disrupt or cause harm

Homework 1 (part 1) assigned today

- You'll be building a simple, unencrypted IM client
- Really, an introduction to network programming
- Due February 2nd at 11:59pm EDT
- Later parts of HW1 will add encryption

Questions?

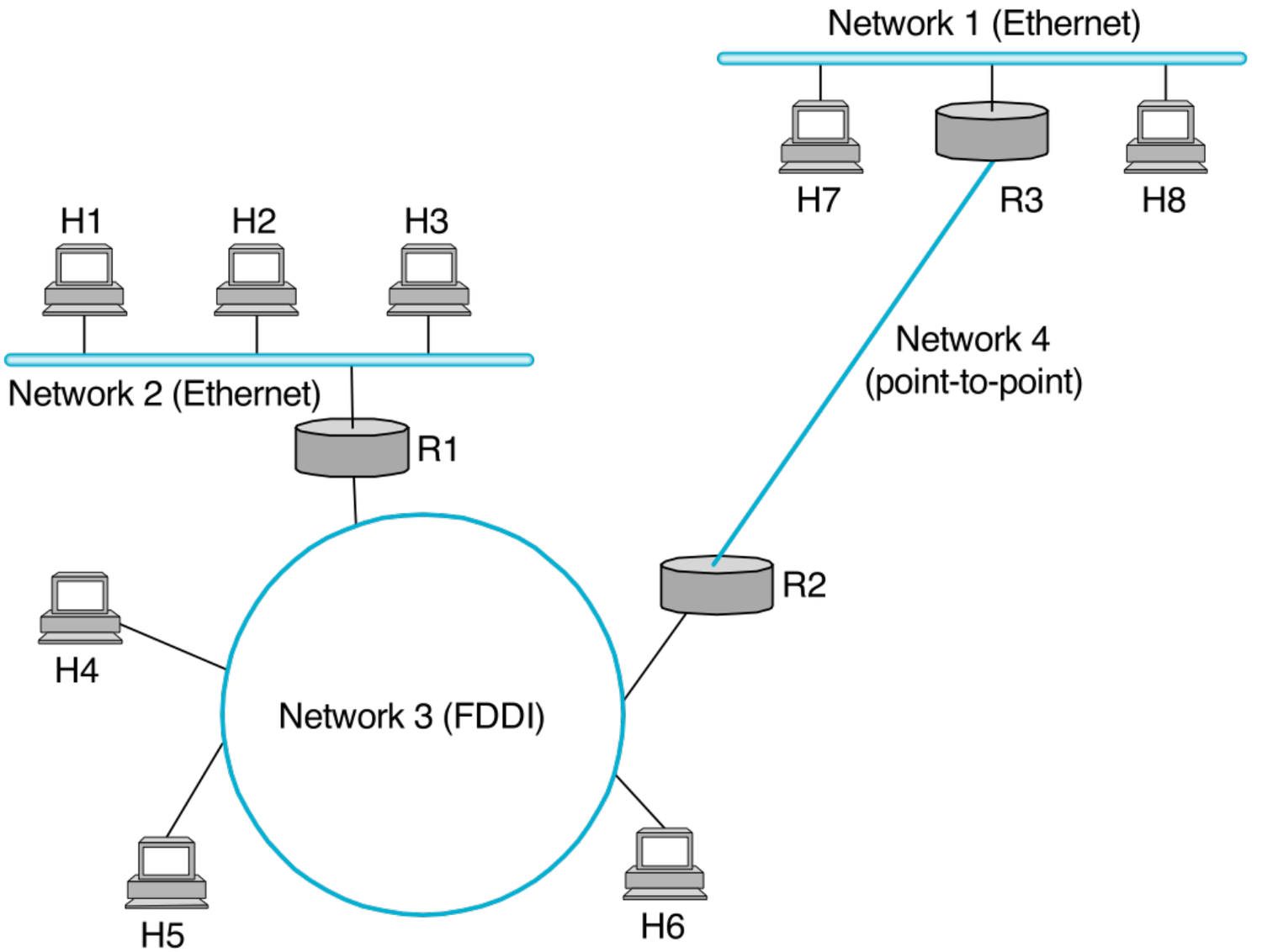
Intro to Networking, Part One

(Or, whatever we can get to before 2:45pm)

Internet History 101

- DARPA – Defense Advanced Research Projects Agency
- ARPANET:
  - World's first operational packet switching network
  - Predecessor of global Internet
  - Started in 1969 with 4 routers @ UCLA, Stanford, UC Santa Barbara, Utah
  - TCP/IP in 1983

Fundamental Goal: An Inter-network



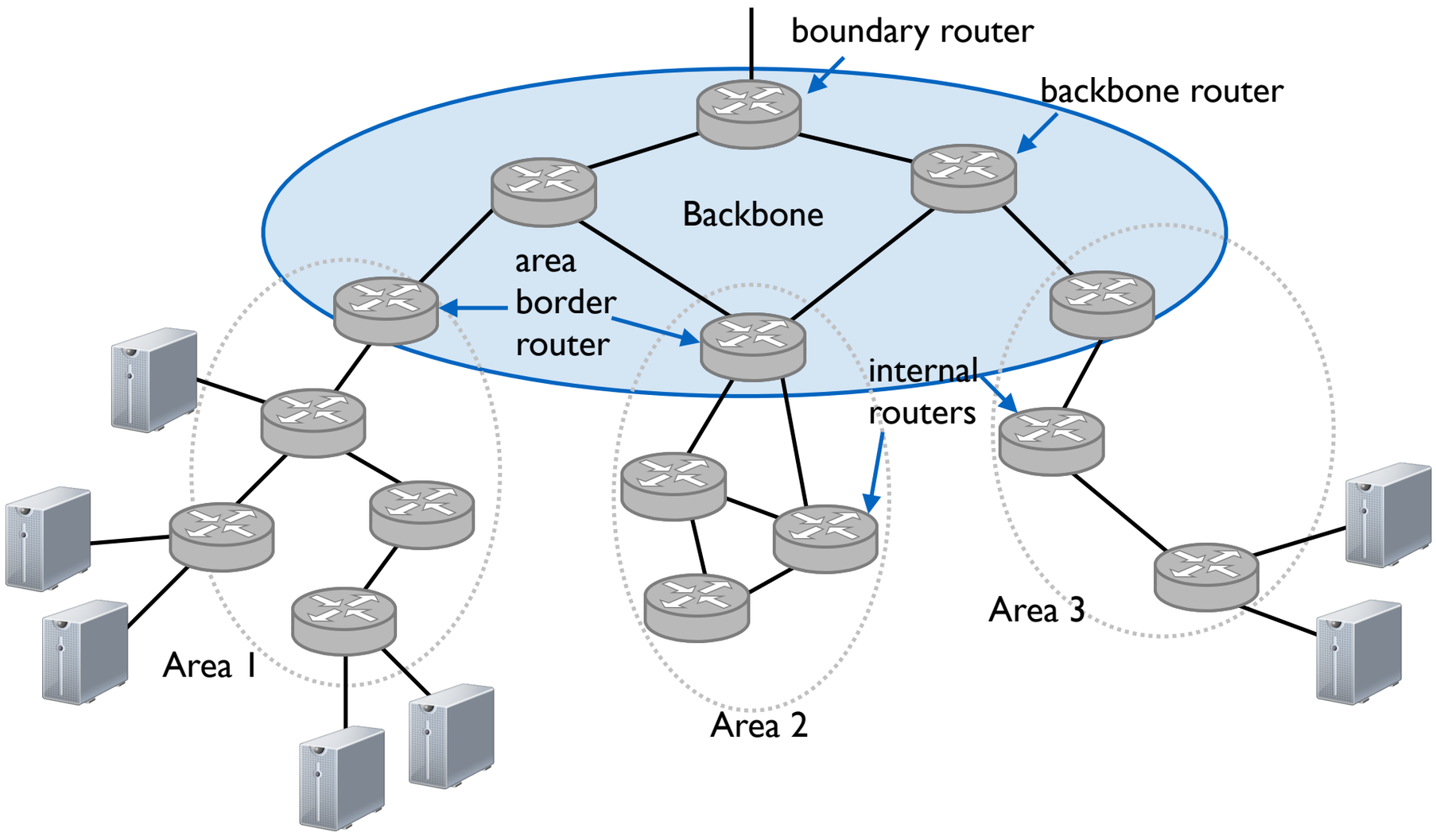
Hn = host, Rn = router/gateway

Goals of the Internet

- Fundamental goal: Inter-connect multiple networks of different types (wired and wireless) via store-and-forward gateways
- Second-level goals:
  - Robust in face of failures
  - Support multiple types of services
  - Support a variety of networks
  - Allow distributed management
  - Cost effective

What is the Internet?

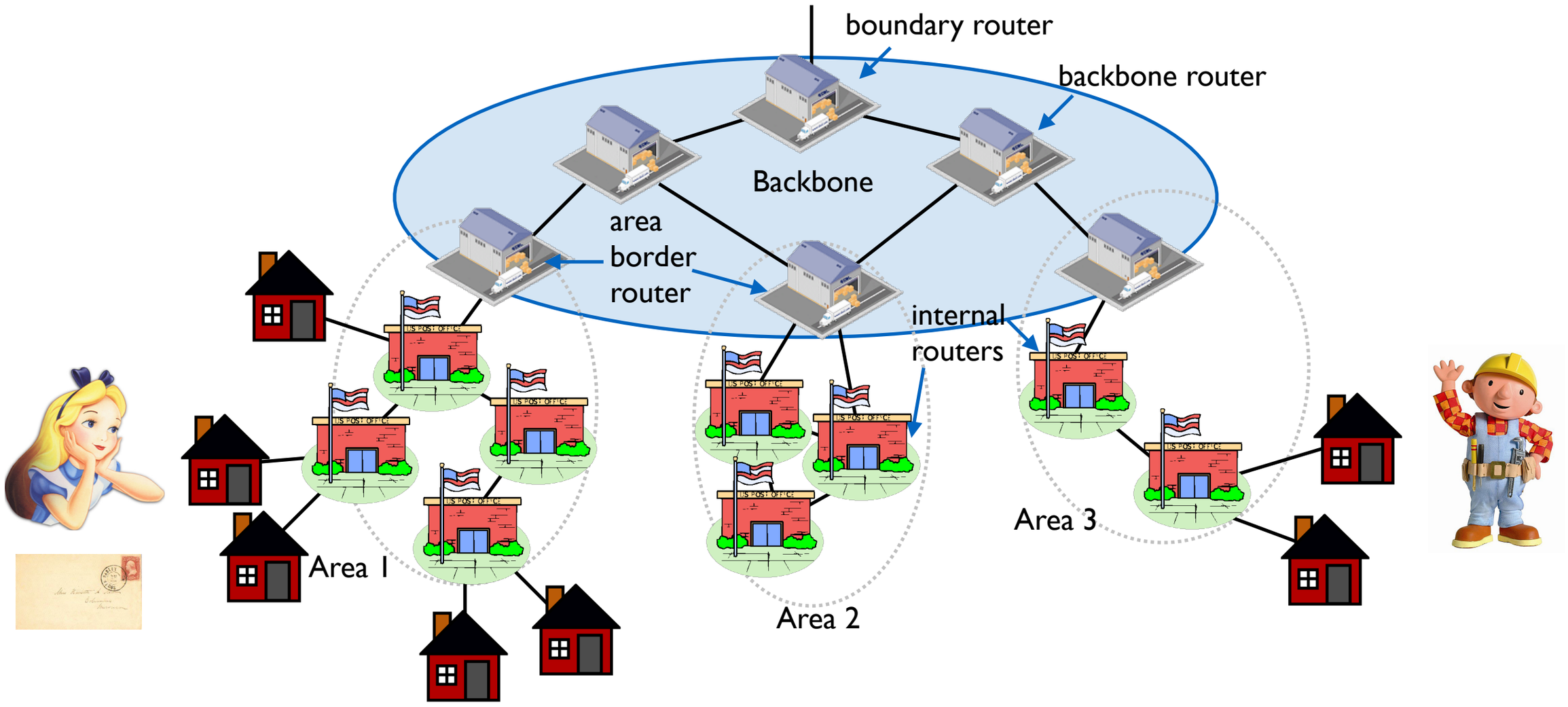
A collection of independently operated
autonomous systems (ASes)



What is the Internet?

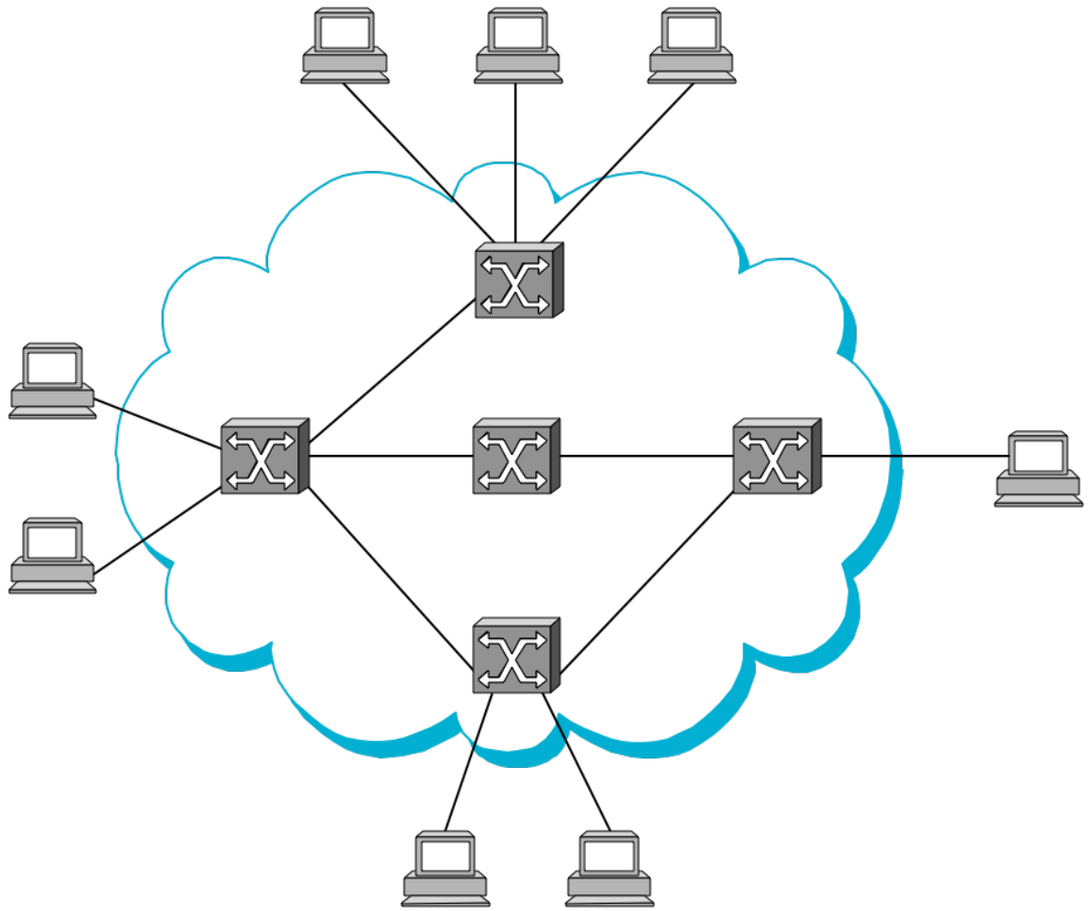
What is the Internet?

A collection of independently operated
autonomous systems (ASes)



Switched Network

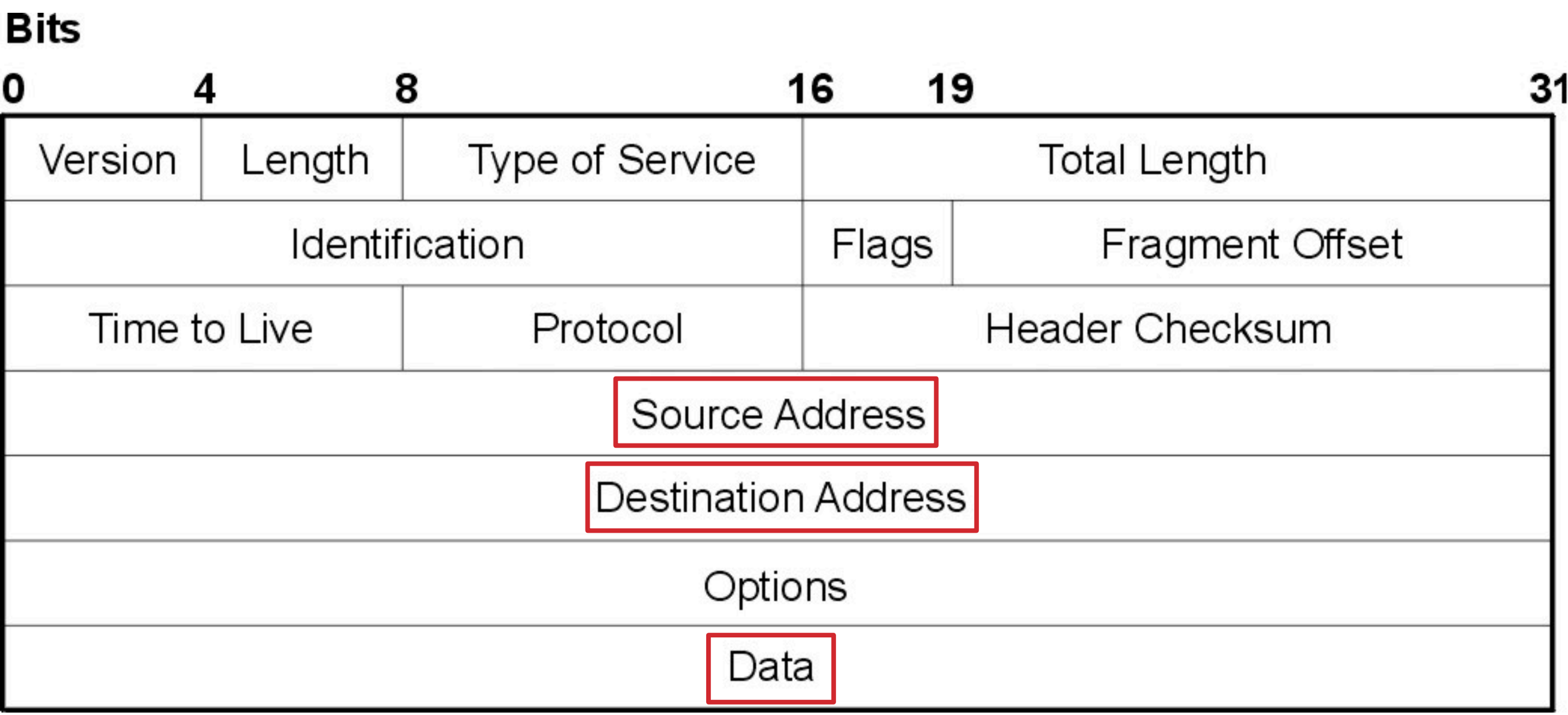
- End-hosts connected to switches
- **Switches:**
  - Forwarding nodes
  - At least two links
  - Also known as bridges or routers



Datagram Packet Switching

- **Packets** – discrete blocks of data
  - Each packet is independently switched
  - Each packet header contains destination address
    - Routing protocol is used to compute next hop
- Example: IP networks
- Advantages:
  - No connection state required
  - Easy to recover from errors
  - Minimal network assumptions

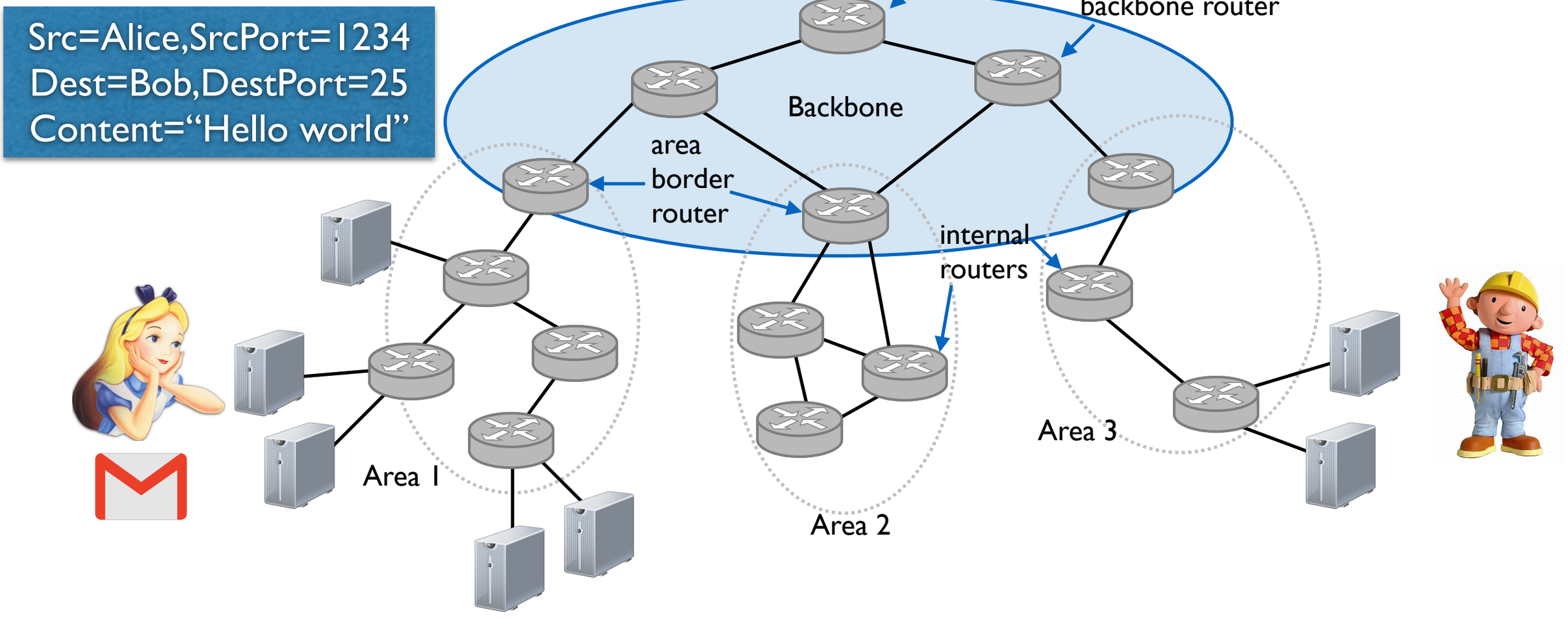
IP Packet Structure



https://tools.ietf.org/html/rfc791

What is the Internet?

A collection of independently operated
autonomous systems (ASes)



Network Programming

- The operating system provides an *interface* for sending/receiving network packets
- A ***socket*** is a descriptor for network communication
- As a client, you ***connect*** your socket to a remote host, and read/write to that socket as you would a file
- As a server, you ***listen*** and ***accept*** incoming connections, and read/write to that socket as you would a file

Internet communication *via sockets*

Src=HostA,SrcPort=1234
Dest=HostB,DestPort=1025
Content="Hello world"

HostA

HostB

running on HostA

```
import socket

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# connect to HostB on port 1025.
s.connect( ("HostB",1025) )
# Let the Internet worry about how my messages get there
s.send('hello world')
s.close()
```

Internet communication

HostA

Src=HostA,SrcPort=1234
Dest=HostB,DestPort=25
Content="Hello world"

Hello world

HostB

```
import socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(('', 1025))  # this socket is bound to my port 1025
s.listen(1)         # specify the "backlog" for this socket
conn, addr = s.accept()  # wait and accept the connection
data = conn.recv(1024)   # read the content of the message
print data;
conn.close()
```

running on HostB

The Seven Layers of OSI