IOWA

CS3640

Overview (5): Network Security & History

Prof. Supreeth Shastri
Computer Science
The University of Iowa

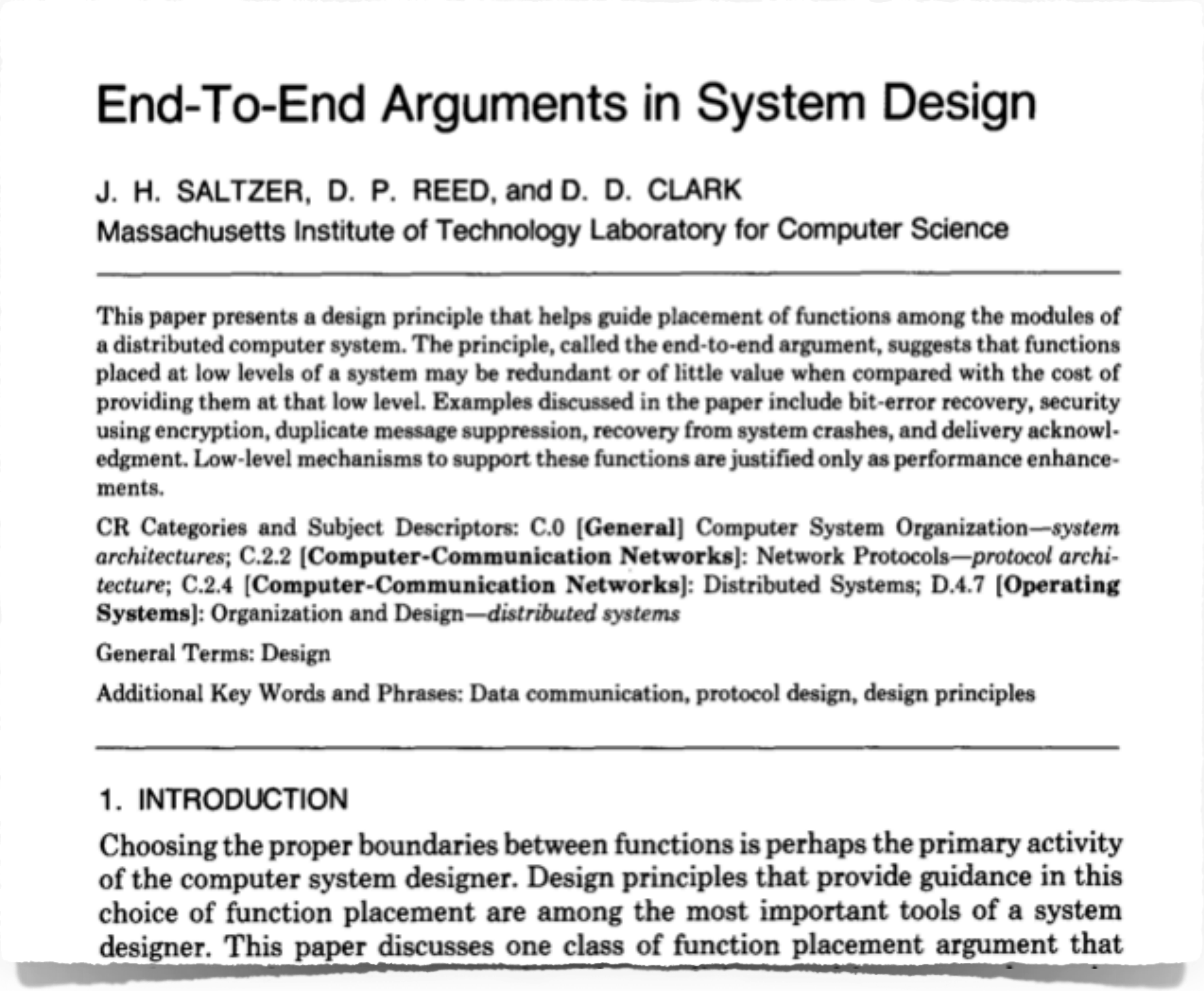
Announcements

Updates to spot quizzes

- *Will be open until 11:59pm*
- *Reach out to me if something doesn't make sense; I'm happy many of you already did so*

Written assignment 1

- *Open now; submission deadline: Feb-7*
- *Part-1 requires reading a research paper on end-to-end principle*
- *Part-2 covers the overview lectures*



End-To-End Arguments in System Design

J. H. SALTZER, D. P. REED, and D. D. CLARK
Massachusetts Institute of Technology Laboratory for Computer Science

This paper presents a design principle that helps guide placement of functions among the modules of a distributed computer system. The principle, called the end-to-end argument, suggests that functions placed at low levels of a system may be redundant or of little value when compared with the cost of providing them at that low level. Examples discussed in the paper include bit-error recovery, security using encryption, duplicate message suppression, recovery from system crashes, and delivery acknowledgment. Low-level mechanisms to support these functions are justified only as performance enhancements.

CR Categories and Subject Descriptors: C.0 [**General**] Computer System Organization—*system architectures*; C.2.2 [**Computer-Communication Networks**]: Network Protocols—*protocol architecture*; C.2.4 [**Computer-Communication Networks**]: Distributed Systems; D.4.7 [**Operating Systems**]: Organization and Design—*distributed systems*

General Terms: Design

Additional Key Words and Phrases: Data communication, protocol design, design principles

1. INTRODUCTION

Choosing the proper boundaries between functions is perhaps the primary activity of the computer system designer. Design principles that provide guidance in this choice of function placement are among the most important tools of a system designer. This paper discusses one class of function placement argument that

TL;DR:

> *If a function can be* ***completely and correctly*** *implemented* ***only with the knowledge of*** *the application at the endpoints of the communication system,*
>
> *then providing that function as a feature of the communication system is* ***not possible*** *(and could be* ***harmful****)*

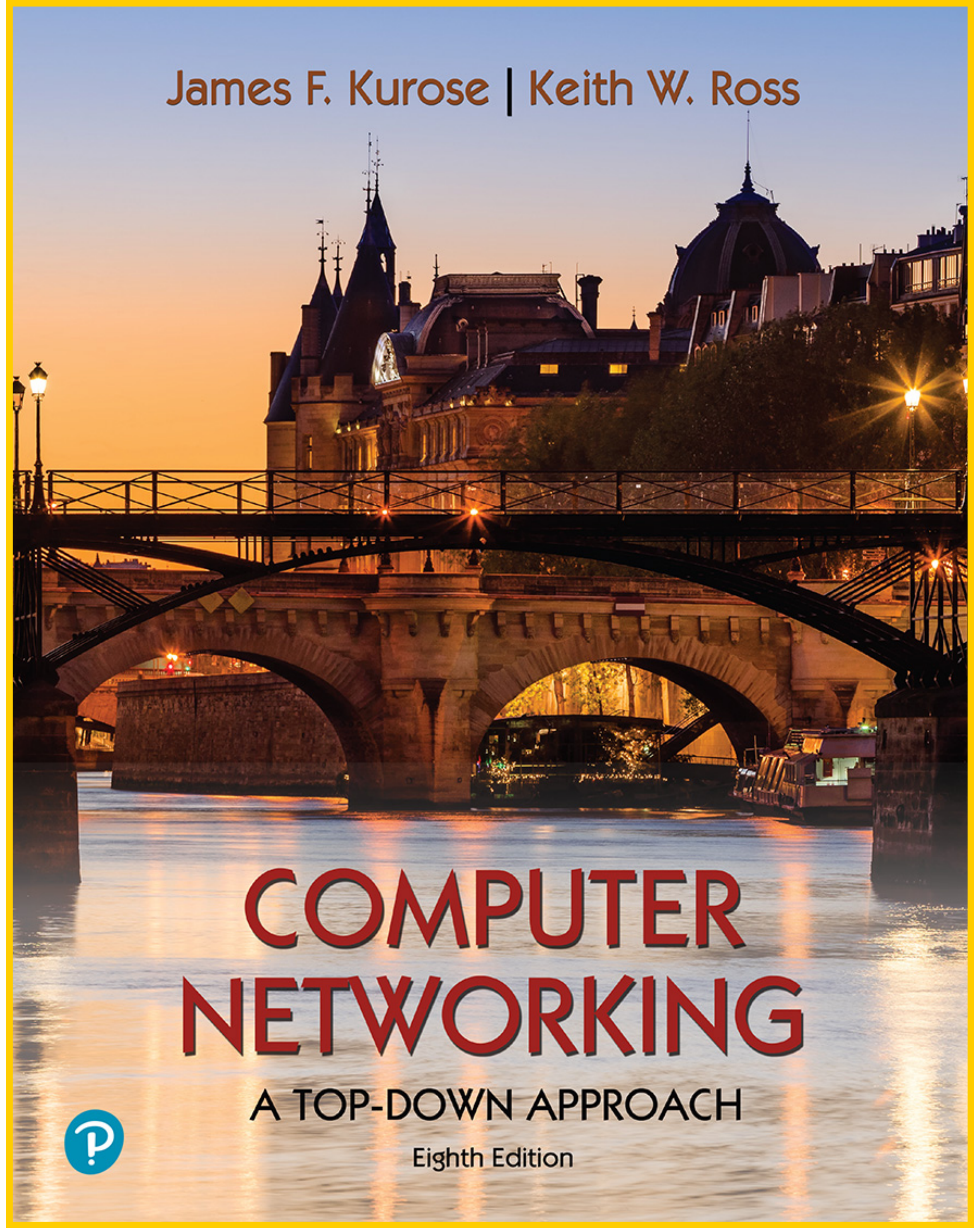
I've never read a research paper before. How do I go about it?

- This paper is easy to follow and fun to read (unlike most research papers)!
- Prof. Sherry has a talk on this: https://www.youtube.com/watch?v=aR_UOSGEizE

Lecture goals

Continuing our in-depth exploration into the structure and functionality of the Internet

- *Internet history and evolution*
- *Network security*



Chapter 1.6 - 1.7

Evolution of the Internet

Internet timeline

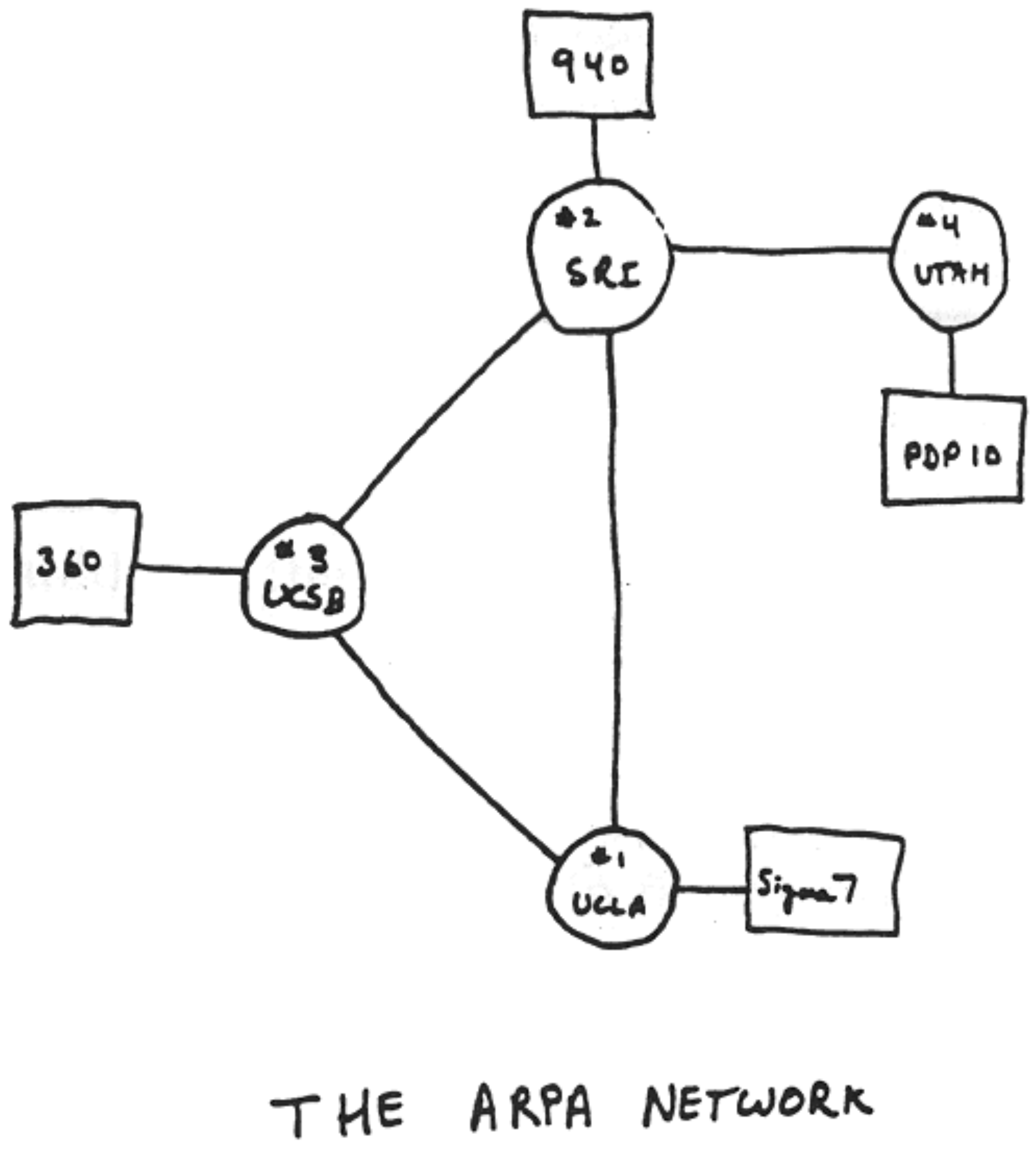
Early development of packet switching

1961: Leonard Kleinrock develops a ***queueing theoretical foundation*** for packet-switching

1964: Paul Baran designs a packet-switching for voice communications in military networks

1969: Advanced Research Projects Agency creates the ***first packet switched*** computer network, ARPAnet

1972: ***First public demonstration*** of ARPAnet by Robert Kahn. ARPAnet has its own host-to-host protocol called Network Control Protocol (NCP) and 15 connected nodes.



Internet timeline

Rise of new, proprietary computer networks

1970s: Multiple proprietary computer networks started emerging. E.g., ALOHAnet, GE ISN, IBM SNA

1974: Cerf and Kahn propose ***internetting***, an architecture for interconnecting autonomous networks

1976: Metcalfe develops the protocol and technology for ***Ethernet***, a wire-connected broadcast network

1980: ARPAnet connects more than 200 hosts

Vinton Cerf and Robert Kahn’s **internetting** principles:

- minimalism
- best-effort service model
- stateless routing
- decentralized control

Internet timeline

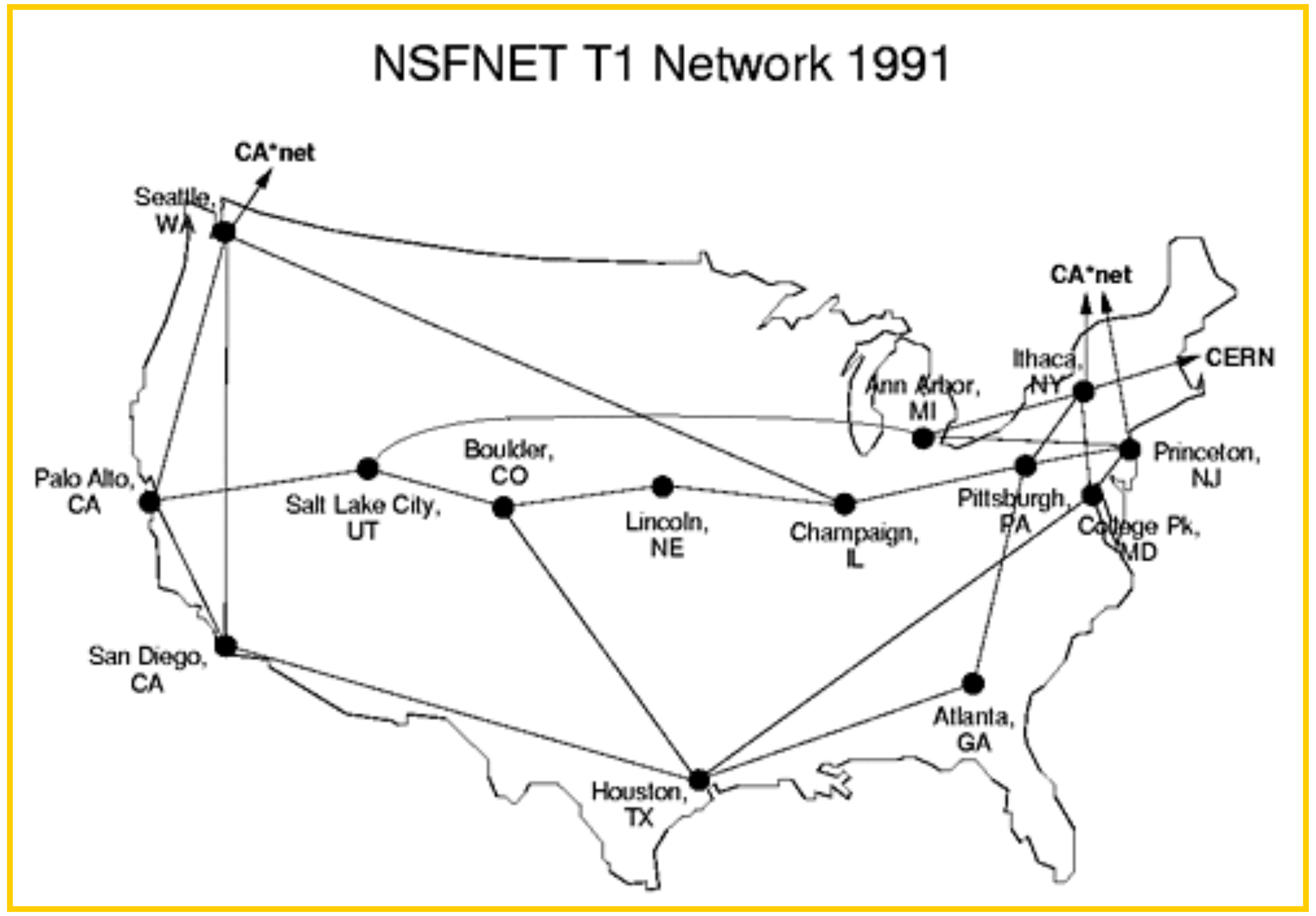
Proliferation of the Internet/protocols

1983: ***TCP/IP*** deployed as the standard network protocol on ARPAnet

1980s: Protocols are designed and deployed for name resolution (DNS), file transfer (FTP), emails (SMTP), etc.,

1986: New national ***backbone networks*** emerged. For example, the NSFnet

1990: the network of networks reaches 100K connected hosts


NSFNET T1 Network 1991

Internet timeline

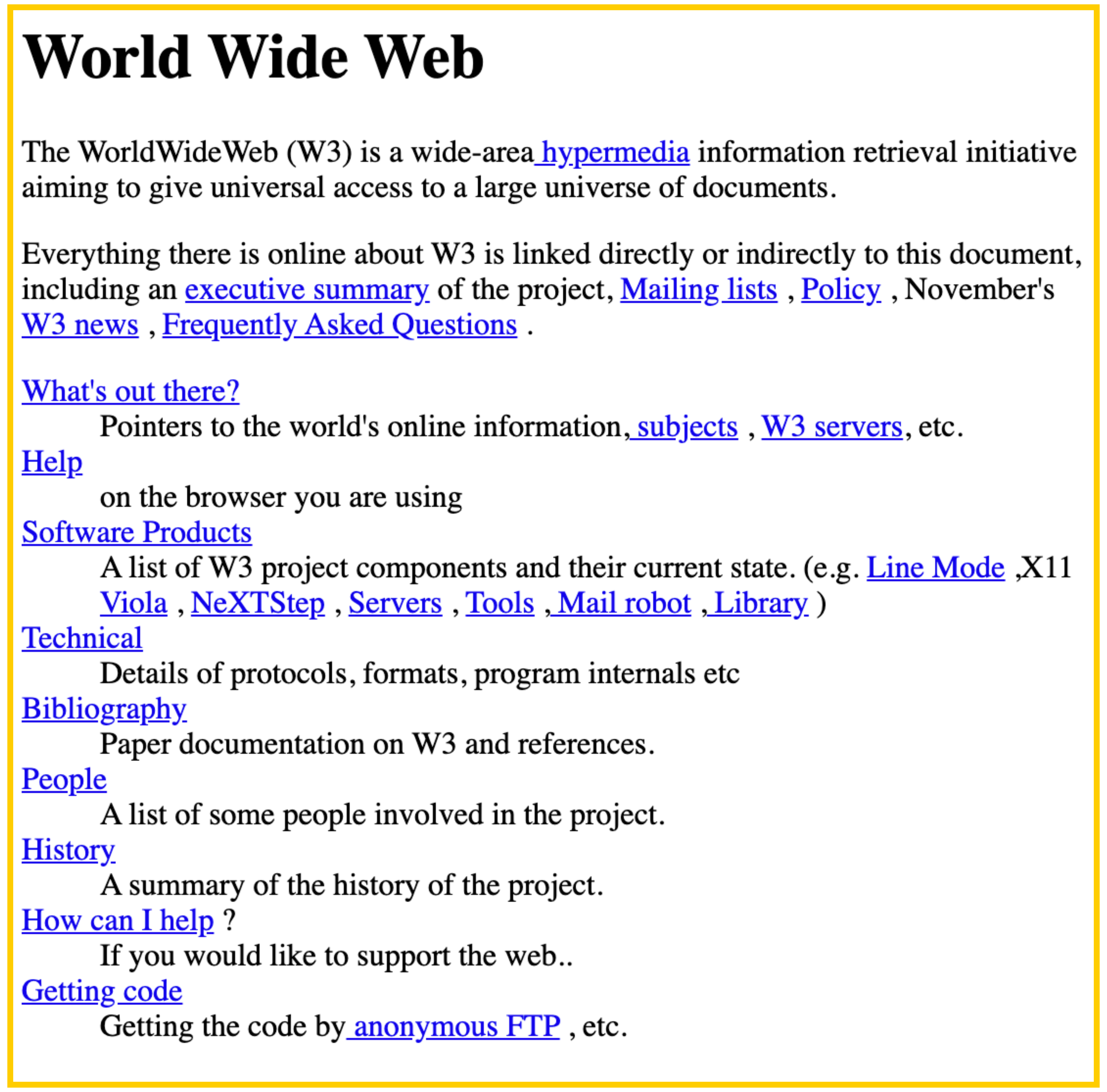
Commercialization and the Internet explosion

1991: ARPAnet decommissioned, and NSFnet lifted its ***restrictions*** on its use for commercial purposes

1991: Tim Berners-Lee builds and demonstrates the ***world wide web (www)*** and its four key components: HTML, HTTP, web server, and web browser

1995: ***Commercial ISPs*** emerge after NSFnet is decommissioned

1998 - 2000: the browser war, the dot-com bubble, and four killer apps (email, www, IM, p2p file share)



World Wide Web

The WorldWideWeb (W3) is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an executive summary of the project, Mailing lists , Policy , November's W3 news , Frequently Asked Questions .

What's out there?
    Pointers to the world's online information, subjects , W3 servers, etc.
Help
    on the browser you are using
Software Products
    A list of W3 project components and their current state. (e.g. Line Mode ,X11 Viola , NeXTStep , Servers , Tools , Mail robot , Library)
Technical
    Details of protocols, formats, program internals etc
Bibliography
    Paper documentation on W3 and references.
People
    A list of some people involved in the project.
History
    A summary of the history of the project.
How can I help ?
    If you would like to support the web..
Getting code
    Getting the code by anonymous FTP , etc.

World's first website.

Courtesy: http://info.cern.ch/hypertext/WWW/TheProject.html

Internet timeline

Hyper connectivity and innovation

2000s: High-speed connectivity in access networks: broadband, 3G/4G, and WiFi technologies

2005 - 2010: Cloud computing, Social networks, Software Defined Networking (SDN)

2010 onwards: New end devices (smart phones) and new traffic (video) overtake the traditional fixed devices and text-based traffic

2017: The Internet has more than 18B devices connected

Network **Security**

Network security (or lack thereof)

The Internet was not originally designed with security in mind

- ***Why?*** *The original operating setup of the Internet: a group of **mutually trusting** users attached to a **transparent** network*
- ***What changed?*** *Growth of the Internet, and commercialization — both of which invalidated the original working conditions/assumptions*
- ***How does it impact?*** *All the layers of networking stack have vulnerabilities. The networking community has been playing catch up.*
- ***So, why not stop-drop-and-learn network security?*** *Sure, but the first step is to develop expertise in networking and protocols*

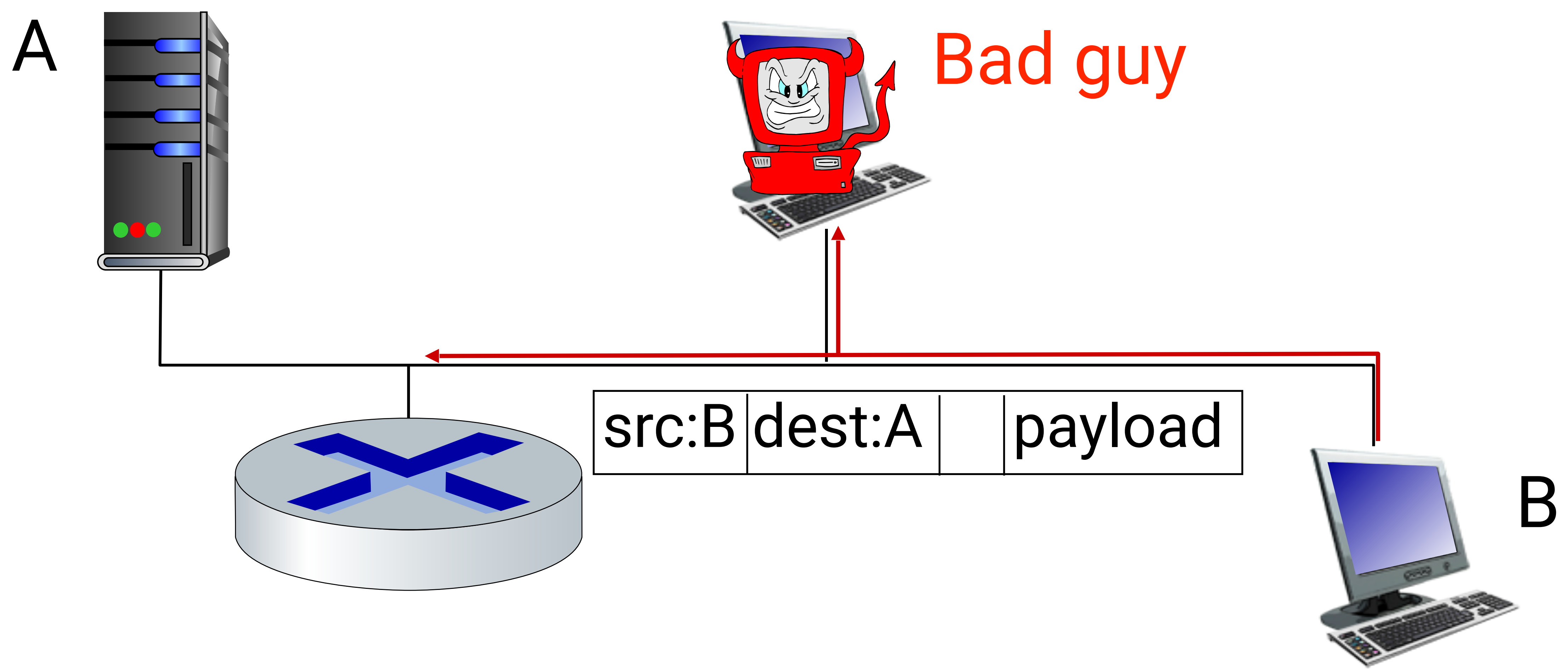
1. The bad guys can **sniff your packets**
2. The bad guys can **masquerade as someone** you trust
3. The bad guys can **break** into your host
4. The bad guys can **attack network infrastructure**

1

The bad guys can **sniff your packets**

Packet sniffer: *a passive receiver that records a copy of every packet that flies by in the network*

- *Could be deployed in any type of network (wired, wireless) and any portion of the network (broadcast LANs, outside of an access network, in the backbone etc)*
- *they capture packets in promiscuous mode, and their presence is difficult to detect*

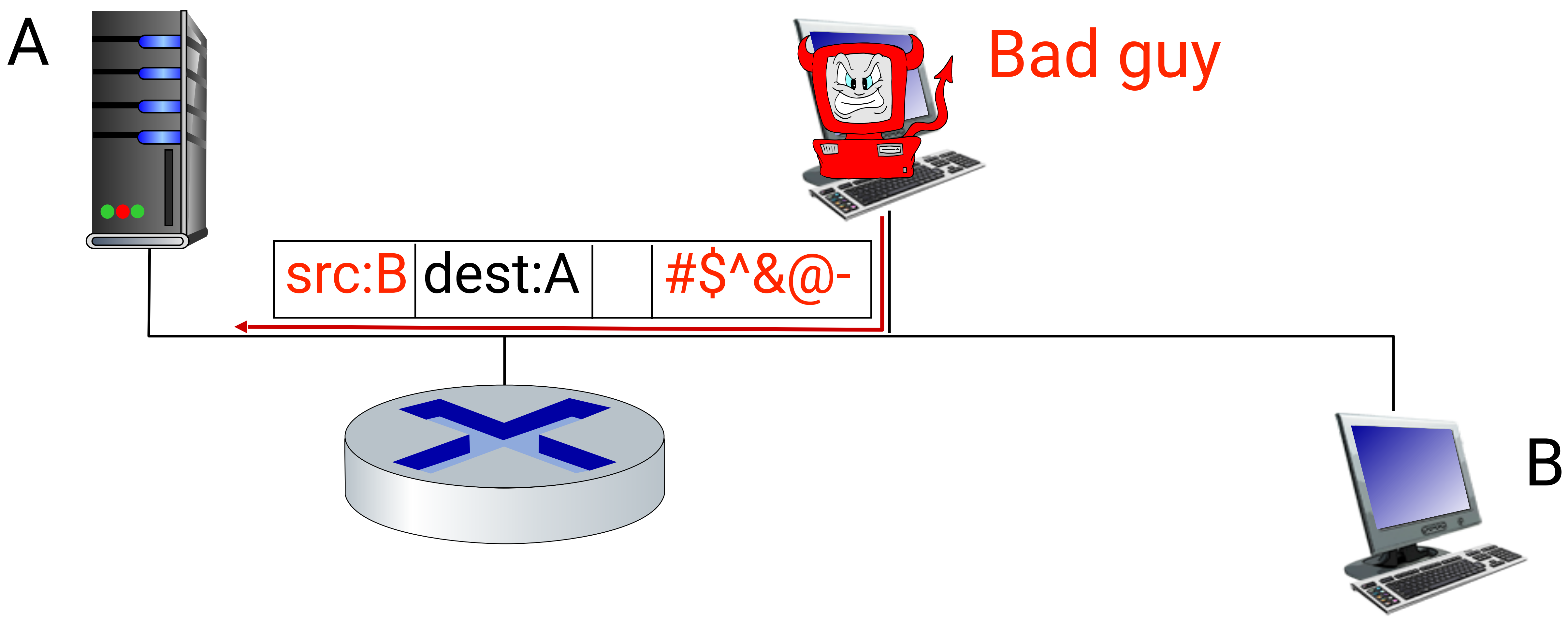


Wireshark: an open-source software for packet-sniffing

2 The bad guys can **masquerade as someone** you trust

IP spoofing: *ability to inject packet into the Internet with a false source address*

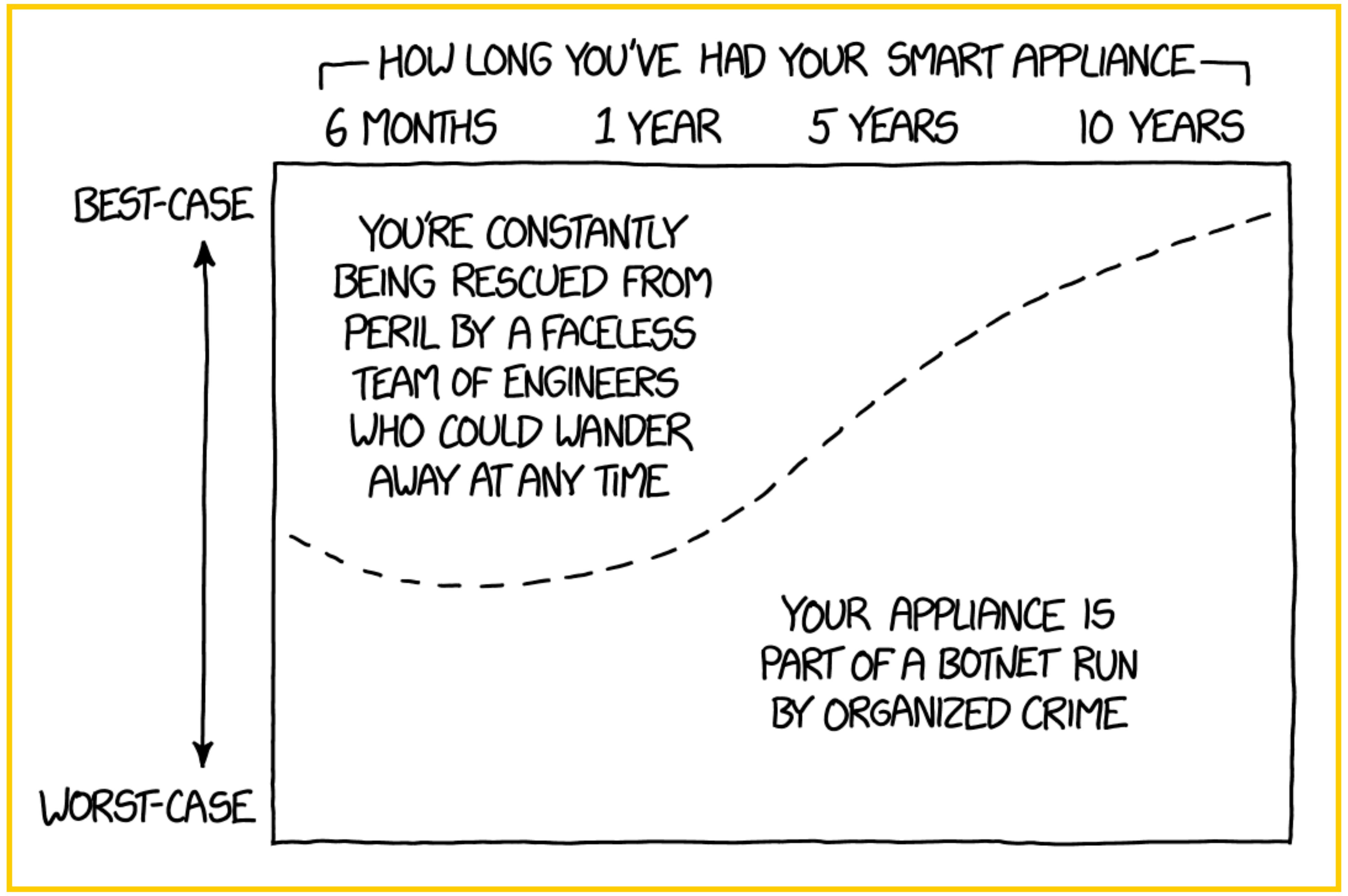
- *It is trivial to create and inject handcrafted packets into the network!*
- *This circles back to the assumptions of the original Internet*
  - *Anyone can send packets to anyone on the Internet (contrast that w/ telephone network)*
  - *User identity is taken at declared face value rather than authenticated by default*



3 The bad guys can **break** into your host

Malware: *malicious software installed on a host system without permission*

- *Examples include viruses, spyware, ransomware, wipers and so on*
- ***Self-replicating** i.e., once they infect a host, they extract contact information and spread themselves to other hosts*
- ***Botnets**. A collection of compromised hosts that could be directed to participate in network attacks orchestrated by bad guys*


HOW LONG YOU'VE HAD YOUR SMART APPLIANCE
6 MONTHS
1 YEAR
5 YEARS
10 YEARS
BEST-CASE
WORST-CASE
YOU'RE CONSTANTLY BEING RESCUED FROM PERIL BY A FACELESS TEAM OF ENGINEERS WHO COULD WANDER AWAY AT ANY TIME
YOUR APPLIANCE IS PART OF A BOTNET RUN BY ORGANIZED CRIME

Courtesy: XKCD

4 The bad guys can **attack network infrastructure**

Denial of Service (DoS): *a class of network attacks, where a network server, host, router, or software is rendered unusable for legitimate user*

1. ***Vulnerability attack***. *Send a well-crafted message to a vulnerable application or OS running on a networked machine.* ***Causes the network service to stop or crash***.
2. ***Bandwidth flooding***. *Send a deluge of packets to the targeted network system.* ***Makes the target's access link clogged.***
3. ***Connection flooding***. *Open a large number of TCP connections at the target system.* ***Causes resource exhaustion at the target***.

Course Structure

Overview	*2.5 weeks*	The Internet; Network edge/core & packet switching; Network protocols
Applications layer	*2.5 weeks*	Principles; Web and HTTP; Email; Video streaming; Socket programming
Transport layer	*2 weeks*	Data transfer service; UDP; TCP; Congestion control
Network layer	*2 weeks*	Routing and forwarding; IP; Routing algorithms; OSPF and BGP
Link layer	*2 weeks*	MAC protocols; LANs and ethernet; Datacenter networking
Research topics	*2 weeks*	SDN, Cloud computing, and Prof. Henning Schulzrinne's lecture

Spot Quiz (ICON)