1. Architecture and principles

Protocols for Data Networks

(aka Advanced Computer Networks)

Lecture plan

A brief history of the Internet [BriefHistory]

[Baran1964]

On distributed communication networks

[Cerf1974]

Cerf and Kahn propose IP: a protocol for packet network intercommunication

[Saltzer1981]

Saltzer, Reed, and Clark argue for the end-to-end principle

Lecture plan

A brief history of the Internet [BriefHistory]

[Baran1964]

On distributed communication networks

[Cerf1974]

Cerf and Kahn propose IP: a protocol for packet network intercommunication

[Saltzer1981]

Saltzer, Reed, and Clark argue for the end-to-end principle

July 1961: Leonard Kleinrock (MIT) published the first paper on **packet switching theory**. Kleinrock convinced the community of the theoretical feasibility of communications using **packets** rather than circuits, and the advantages of **statistical multiplexing**.

1961 1962 1964 1965 1966 1967 1968 1969 1970 1972 1973 1974 1980 1983 1989 1990 1995

August 1962: "**Galactic network**" concept described by J.C.R. Licklider (first head of the computer research program at DARPA). Very similar to the Internet of today.

1964: Paul Baran and his group at RAND published a paper on **packet switching networks** for secure voice in the military.

1961 1962 **1964 1965** 1966 1967 1968 1969 1970 1972 1973 1974 1980 1983 1989 1990 1995

1965: The **first wide-area computer network**. Lawrence G. Roberts and Thomas Merrill connected the TX-2 computer in Massachusetts to the Q-32 in California with a low-speed dial-up telephone line. They proved it was possible to connect computers, but understood the **circuit switched** telephone line was **inadequate** for the job.

1966: Roberts went to DARPA to develop the **computer network** concept.

1961 1962 1964 1965 **1966** **1967** 1968 1969 1970 1972 1973 1974 1980 1983 1989 1990 1995

1967: Roberts presented the **idea of the "ARPANET" in a conference**. At the same conference there was also a paper on a packet network concept from the UK by Donald Davies and Roger Scantlebury of NPL.

It was the first time the three groups (MIT, NPL, and RAND) realized they were doing similar work in parallel. The word "packet" was adopted from the work at NPL.

August 1968: RFQ released by DARPA for the **development of the packet switches** (“Interface Message Processors”). The RFQ was won by Frank Heart at BBN.

1961 1962 1964 1965 1966 1967 **1968** **1969** 1970 1972 1973 1974 1980 1983 1989 1990 1995

September 1969: The **first node of the ARPANET** was chosen to be Kleinrock’s Network Measurement Center at UCLA. One IMP and one host were installed. The Stanford Research Institute provided the second node.

October 1969: The **first host-to-host message** was sent from Kleinrock's lab to SRI.

"LO"

1961 1962 1964 1965 1966 1967 1968 **1969** 1970 1972 1973 1974 1980 1983 1989 1990 1995

End of 1969: **Four host computers** connected together into the initial ARPANET.

December 1970: Initial ARPANET host-to-host protocol finished, the **Network Control Protocol**. S. Crocker led the working group.

1961 1962 1964 1965 1966 1967 1968 1969 **1970** **1972** 1973 1974 1980 1983 1989 1990 1995

1972: The **electronic mail** application was introduced. It would become the largest network application for over a decade.

1973: **Ethernet** technology developed by Bob Metcalfe at Xerox PARC.

1961 1962 1964 1965 1966 1967 1968 1969 1970 1972 **1973** **1974** 1980 1983 1989 1990 1995

May 1974: Vint Cerf and R. Kahn published the **TCP/IP paper**: "A Protocol for Packet Network Interconnection"

1980s: Widespread development of LANs, PCs and workstations allow the nascent **Internet to flourish**.

1961 1962 1964 1965 1966 1967 1968 1969 1970 1972 1973 1974 **1980** **1983** 1989 1990 1995

1983: The Domain Name System (**DNS**) was invented by Paul Mockapetris of USC/ISI.

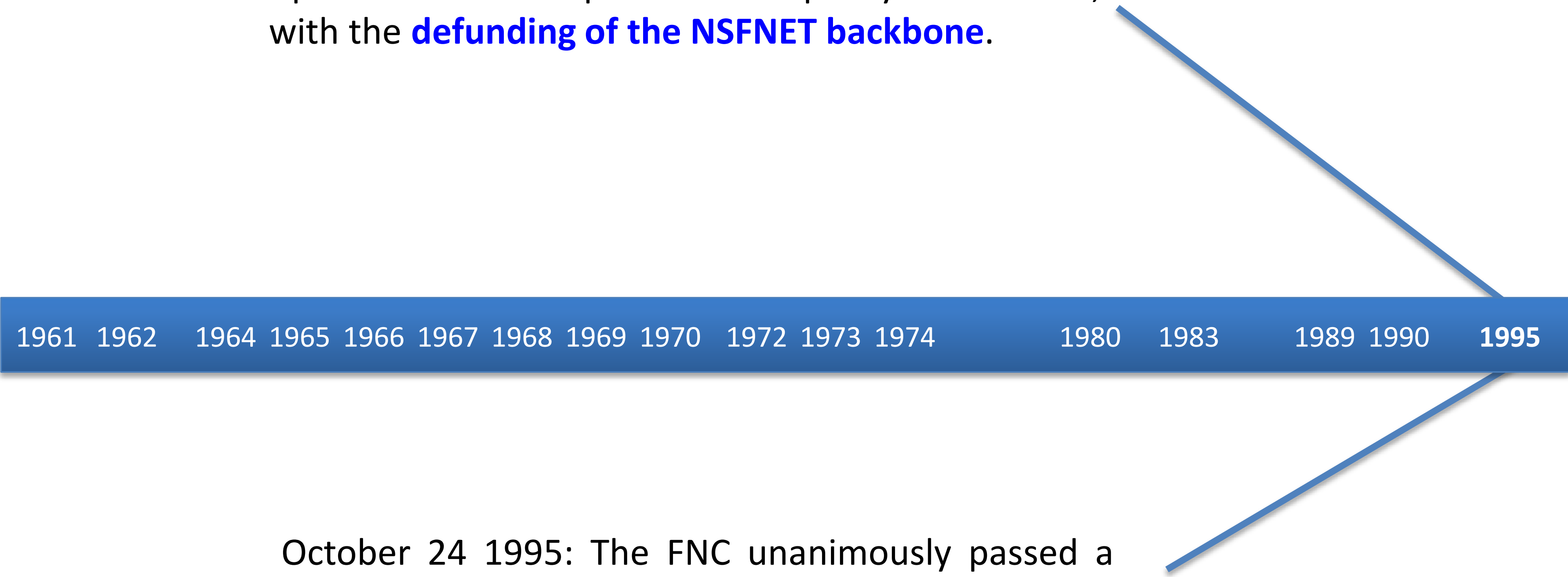
1 January 1983: Transition of the ARPANET host protocol from NCP to **TCP/IP**. A "**flag day**" style transition (all hosts converted simultaneously!)

1989: ARPANET **decommissioned**.

1961 1962 1964 1965 1966 1967 1968 1969 1970 1972 1973 1974 1980 1983 **1989** **1990** 1995

1990: Tim Berners-Lee invents the **World Wide Web**.

April 1995: NSF’s privatization policy culminates, with the **defunding of the NSFNET backbone**.



October 24 1995: The FNC unanimously passed a resolution **defining the term Internet**.

Lecture plan

A brief history of the Internet [BriefHistory]

[Baran1964]

On distributed communication networks

[Cerf1974]

Cerf and Kahn propose IP: a protocol for packet network intercommunication

[Saltzer1981]

Saltzer, Reed, and Clark argue for the end-to-end principle

Goal

- A communication network that survives enemy attacks

- Criterion of survivability
  - % of stations surviving the attack and with connection to the largest group of surviving stations
  - Small groups considered ineffective

Options

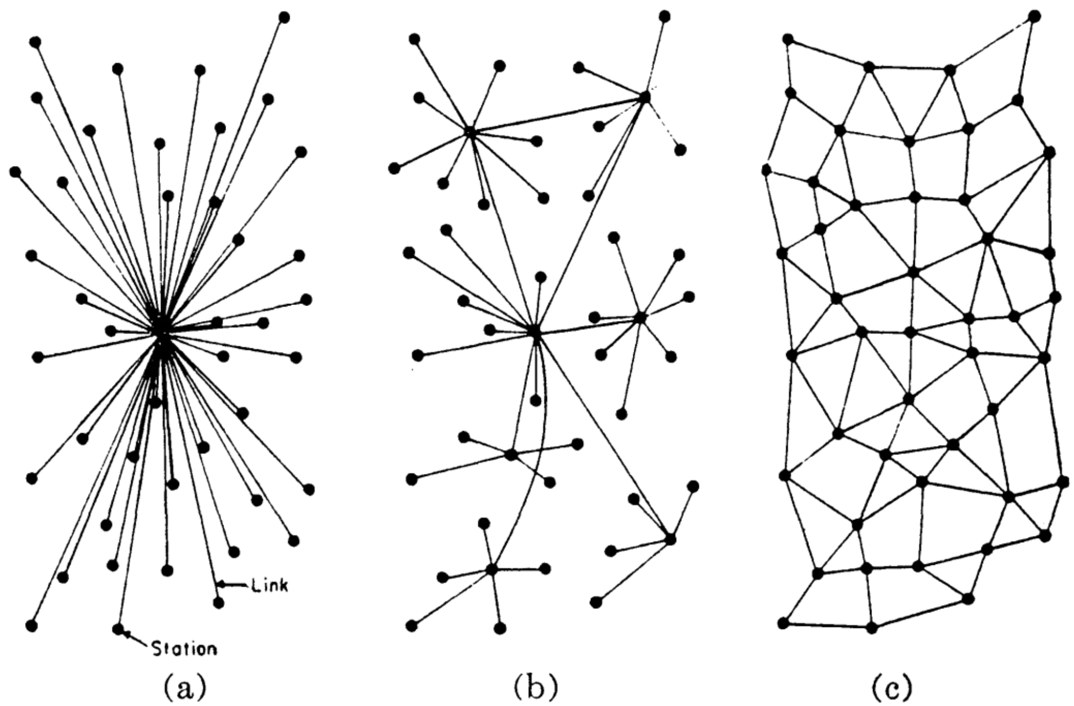


Fig. 1—(a) Centralized. (b) Decentralized. (c) Distributed networks.

- **Centralized: highly vulnerable given central point of attack**
- **Decentralized: vulnerable as an attack to a small number of nodes can destroy communications**
- **Distributed: more resilient**

Redundancy = connectivity

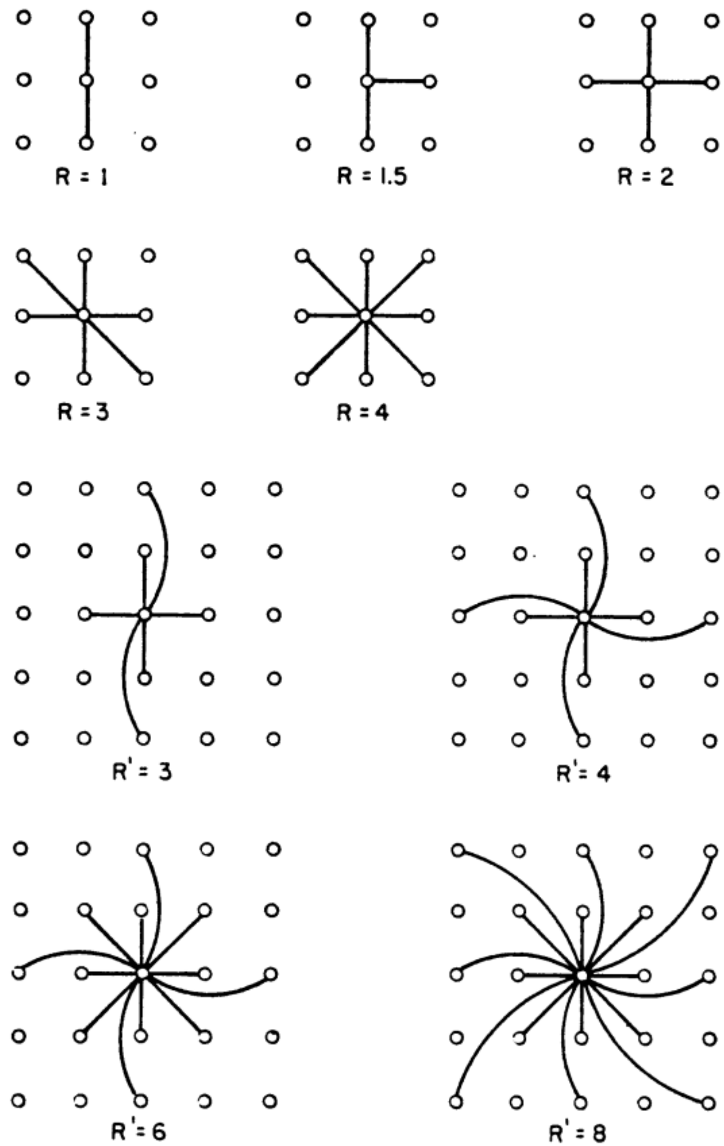


Fig. 2—Definition of redundancy level.

Node/link destruction

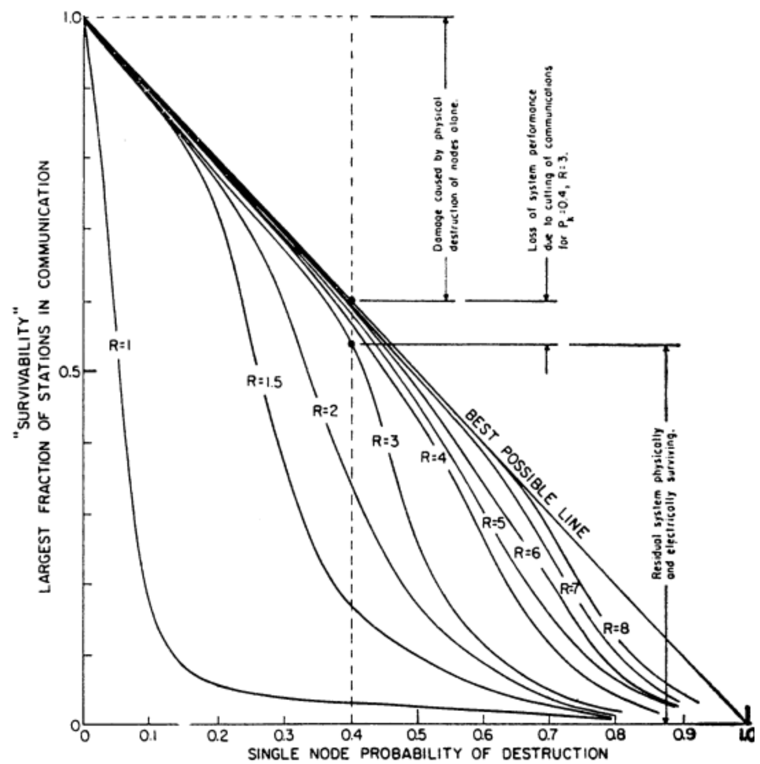


Fig. 4—Perfect switching in a distributed network: sensitivity to node destruction, 100 per cent of links operative.

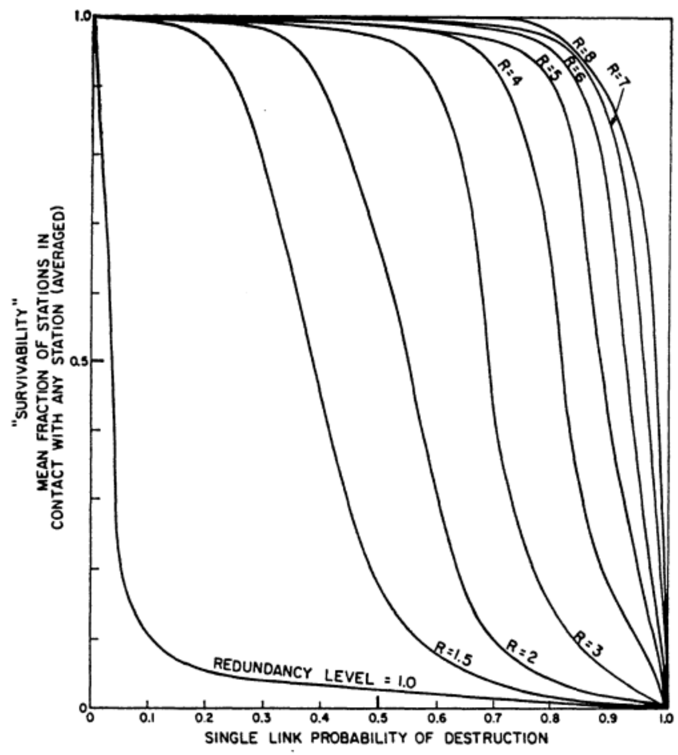


Fig. 5—Perfect switching in a distributed network: sensitivity to link destruction, 100 per cent of nodes operative.

- Extremely survivable networks are possible with moderately low node redundancy
  - Additional redundancy gains little
- Little system degradation even using extremely unreliable links

Node and link destruction

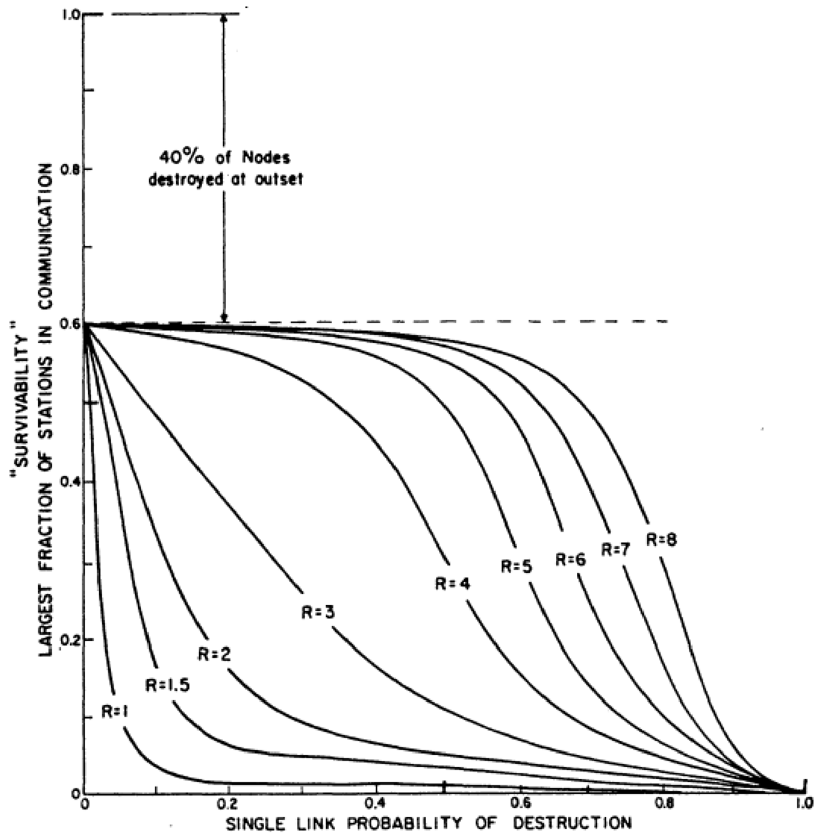


Fig. 6—Perfect switching in a distributed network: sensitivity to link destruction after 40 per cent nodes are destroyed.

- **Unreliable links** can be used in a distributed network almost as **effectively** as perfectly reliable links

On "future" systems

- Requirements
  - Take advantage of the average demand over all users instead of having to allocate a full peak demand channel to each
    - Statistical mux gain?
  - Most economic for many users to share a common resource
    - Packet switching vs circuit switching
  - A wide mix of different transmission links with different data rates combined to form a common resource
    - The Internet?
- All digital distributed system
  - Universally standardized message blocks
    - IP?
  - The routing "doctrine" should find the shortest possible path
    - Proposed "hot potato" heuristic (similar to BGP?)
  - Postman analogy
    - As in packet switching nets?

Final words

there are reasons to suspect that we may not wish to build future digital communication networks exactly the same way the nation has built its analog telephone plant.

There is an increasingly repeated statement made that one day we will require more capacity for data transmission than needed for analog voice transmission. If this statement is correct, then it would appear prudent to broaden our planning consideration to include new concepts for future data network directions. Otherwise, we may stumble into being boxed in with the uncomfortable restraints of communications links and switches originally designed for high-quality analog transmission. New digital computer techniques using redundancy make cheap unreliable links potentially usable. Some sort of switched network compatible with these links appears appropriate to meet this new upcoming demand for digital service.

might best be designed for such data transmission and survivability at the outset. Such a system should economically permit switching of very short blocks of data from a large number of users simultaneously with intermittent large volumes among a smaller set of points.

What you said

"Paul Baran's work does not only considers the communication network of the present time it was written but also takes an insight for future improvements"

Sérgio, Francisco, Daniel

"Este novo método de switching apresentado que é um predecessor do packet switching, propõe a idia de que numa rede distribuída existiriam em cada nó a informação sobre qual o melhor nó para o qual enviar a mensagem, ou seja, uma routing table"

Ricardo, Inês

Lecture plan

A brief history of the Internet [BriefHistory]

[Baran1964]

On distributed communication networks

[Cerf1974]

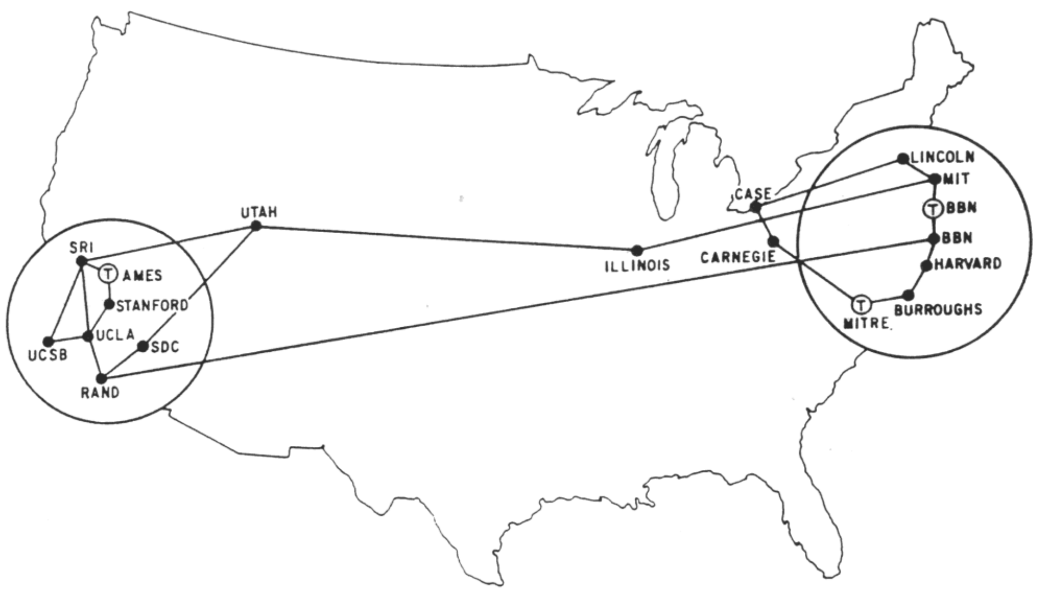
Cerf and Kahn propose IP: a protocol for packet network intercommunication

[Saltzer1981]

Saltzer, Reed, and Clark argue for the end-to-end principle

Context: nets in the 1970s

- **Multiple heterogeneous packet switching networks unconnected**
  - **ARPAnet, data-over-cable, packet satellite (Aloha), packet radio, …**
  - **They shared a common protocol, but could only communicate on the same network**



MAP 4 September 1971

Problem

- Existing protocols addressed only the problem of comunication on the same network

- Heterogeneous network designs
  - Different forms of addressing the receiver
  - Different maximum packet sizes
  - Different ways to deal with lost or corrupted data
  - Different fault detection mechanisms, routing, etc.

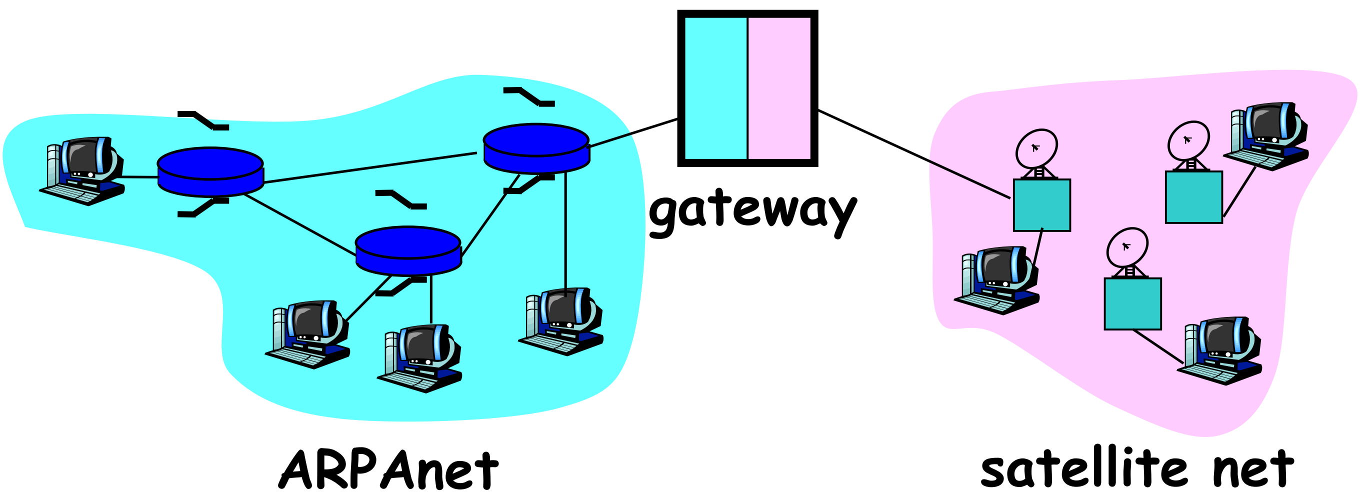
- How to economically resolve these differences between networks?

Contribution

- A protocol that supports interconnection between different packet switching networks
  - A "network of networks"
- Description of a gateway as an interface between the different networks
  - What is the modern name of this so-called gateway?

The gateway

- The gateway has the responsibility to properly **routing** data
- And to **hide** differences between networks
  - It receives packets from network A and reformats them to meet the requirements of network B



Alternatives

- **Are there other places where one could handle this heterogeneity?**

- **Maybe at the host?**
  - **Then every host would have to implement every protocol...**

Internetwork layer: key ideas

internetwork header

local header	source address	dest. address	seq. #	byte count	flag field	text	checksum

- Internetwork header in standard format
  - Who needs to interpret this header?
    - Gateways and end-hosts
- Source and destination addresses: what for?
  - To uniformly and uniquely identify every host
- Sequence number and byte count: what for?
  - Ensure proper sequencing of the data
  - Detect fault conditions
- Checksum: what for?
  - Enable end-to-end detection of corrupted text
- Is fragmentation allowed? What is the position of the authors on this technique?
  - The gateways should not perform reassembly. It should be the host

Process level communications

- TCP enables pairs of processes to communicate
  - Full duplex

- Key ideas
  - Port numbers: what for?
    - to (de)multiplex packets among processes
  - Breaking messages into segments
  - Sequence numbers and reassembly
  - Retransmission and duplicate detection
  - Window-based flow control: what for?
    - Adjust the packet rate to how much the receiver can process

Strengths

- Discussion: what did they get right?
  - Which ideas were key to the Internet's success?
  - Which decisions still seem right today?

- Enable interconnection between independent networks of arbitrary designs
  - A "network of networks"

- Interconnection preserves intact the internal operation of each individual network
  - No changes to current networks

- Providers can chose whatever network technology they prefer
  - They will interwork with others through the meta-level "Internetworking Architecture"

Weaknesses

- Discussion: what did they miss?
  - Which ideas had to be added later?
  - Which decisions seem wrong in hindsight?
- "The choice for network identification (8 bits) allows up to 256 distinct networks. This size seems sufficient for the foresseable future."
  - ☺
- Relatively small importance given to security
- No congestion control (only flow control)
  - What is the difference?
- Host mobility not considered

Discussion

- What would you do in a clean-slate design?
  - Would you do anything differently?

- Larger address space?
- Security by design?
- Add a congestion header to signal congestion?
- Separate host ID from topology?
- No fragmentation?

What you said

"We noticed that this paper definitely shows its age, the addressing space of 8- bits in the TCP that they believed would be enough… Another part that shows the age of the paper are the considerations about security, the authors believed that by having different message streams to different ports would provide relative security"

Guilherme, Francisco

"The authors did not expect the internet to blow up as it did and, as such, defined short bit lengths in certain fields that later proved not enough due to its sheer size… We can also augment the security of TCP"

António, João, João

Lecture plan

A brief history of the Internet [BriefHistory]

[Baran1964]

On distributed communication networks

[Cerf1974]

Cerf and Kahn propose IP: a protocol for packet network intercommunication

[Saltzer1981]

Saltzer, Reed, and Clark argue for the end-to-end principle

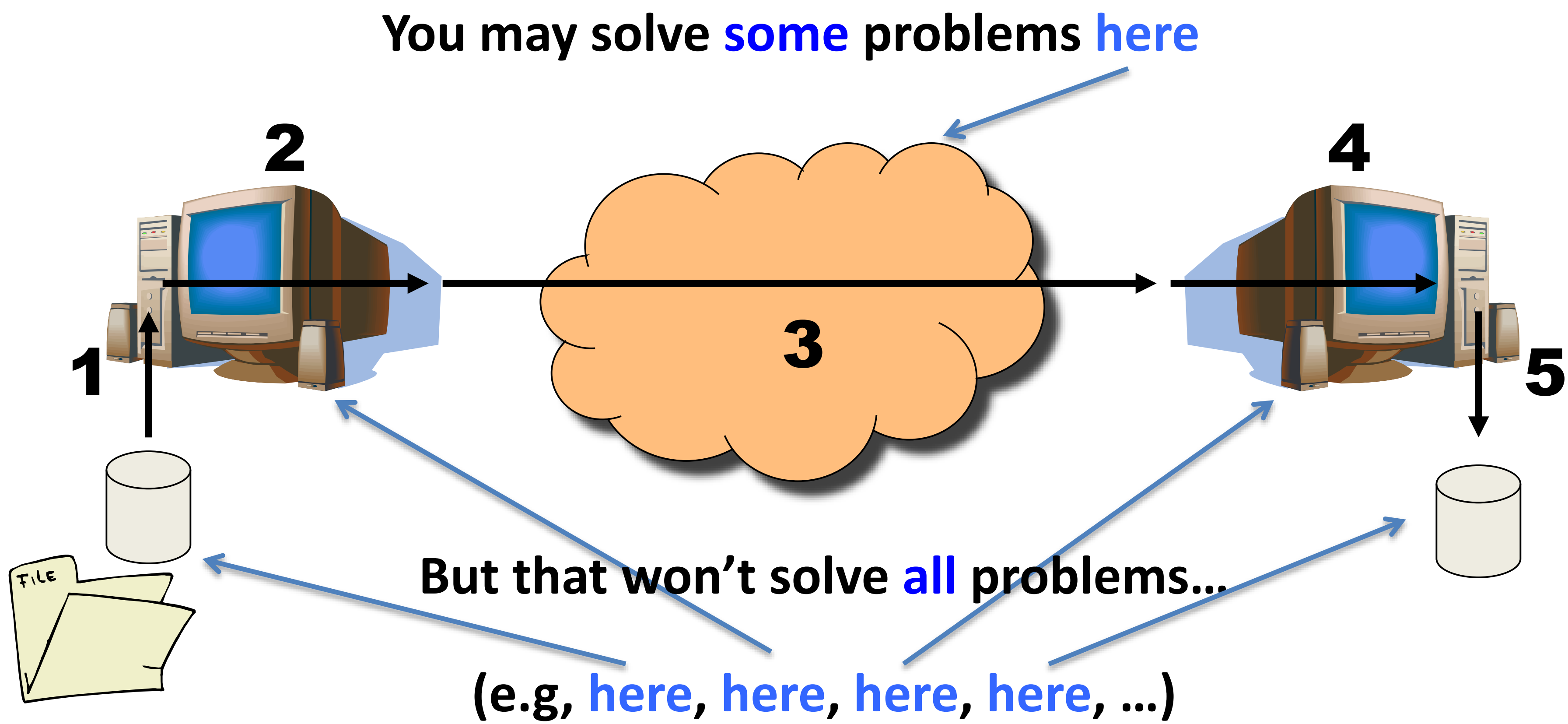
End-to-end, formally

"The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the end points of the communication system.

Therefore, providing that questioned function as a feature of the communication system itself is not possible.

(Sometimes an incomplete version of the function provided by the communication system may be useful as a performance enhancement.)"

End-to-end, pictorially



- The economic cost of trying to solve **all** problems **always** (all with low probability to occur) would be **huge**.
- Operations should thus occur **only** at the end points
  - … unless needed for **performance** optimization

e2e examples

- Error handling in reliable file transfer
  - The network sub-system may guarantee reliable data transmission
    - With checksums, sequence numbers, retry mechanisms, …
    - But you still need e2e error checksum, and retry transmission!
    - So the question: why should the network bother?!
      - Well, if the network can correct some problems, then the number of retries decreases, thus increasing performance
- End-to-end versus in-network encryption
  - Problems with in-network encryption:
    - You have to trust the network to manage the encryption keys
    - Data will be in the clear as it passes into the target node and is fanned out to the app
    - The authenticity of the message must still be checked by the application!
- Reliable communication (in general)
  - Has cost in terms of delay
    - Requiring retransmissions and waiting for in-sequence packets
  - But your VoIP apps (and others) do not need such reliable communication: they just require low delay!

Strengths

- The complexity in the core is reduced, reducing costs and facilitating upgrades
- Generality in the network increases the chances that an application can be added without having to change the core of the network
- Many argue that the e2e principle allowed Internet to grow rapidly because innovation took place at the edge, in applications and services
  - Arguably, innovation at the edge is easier

Weaknesses

- Any problems in placing functionality only at the ends?
  - Slower error detection
  - End-to-end retransmission wastes bandwidth
  - …

Alternatives and trade-offs

- **Alternatives**
  - **Put functionality at each hop (i.e., in the network)**

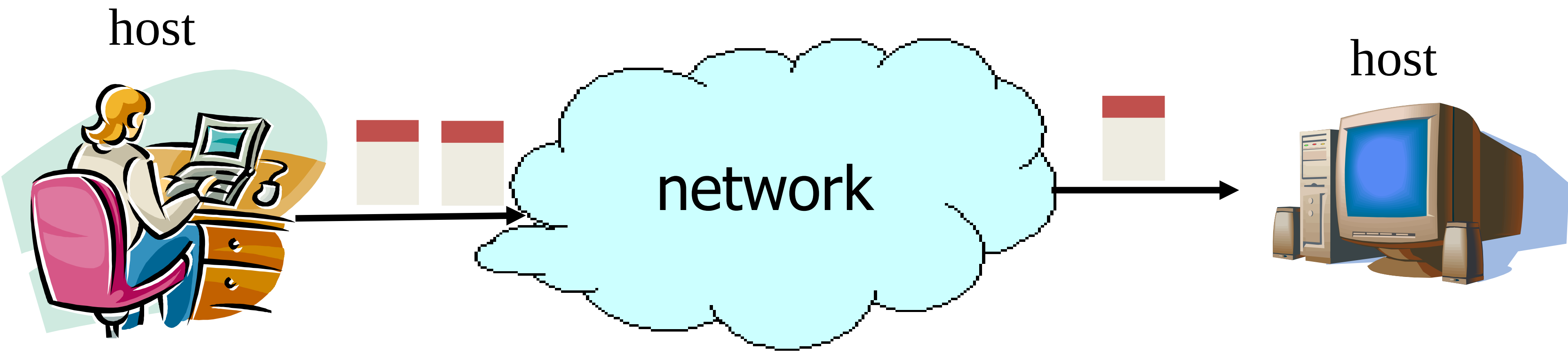
- **Problems of this alternative**
  - **All applications pay the price**
    - **Maybe you don't want reliable transmission (VoIP says: "I want speed, no prob some packets are dropped!")**
    - **Maybe you don't want your data encrypted ("I want all the world to eavesdrop me!")**
    - **Maybe …**
  - **Plus, end systems still need to check for errors**
    - **Why all the effort in the first place?!**

Consequences of e2e

- In layered design, the e2e principle provides guidance on where functions belong. Consequences:

  1. “Dumb, minimal” network
  2. “Intelligent” end-points.

- This led to an interesting host-network division of labor
  - Network
    - Best-effort packet delivery
  - Hosts
    - Everything else



Does the e2e principle still hold?

- e2e principle appears to have been diluted: NATs, firewalls, VPN tunnels
  - Perhaps not surprising: e2e principle grew in an era of trust among users. Now network must protect itself.

- The network is no longer "dumb, minimal"
  - Now over 7,000 RFCs!
  - Router OS's based on over 100M lines of source code!

Discussion

- When should the network support a function anyway?
  - Link-layer retransmission in wireless networks?
  - Access control?
- Who's interests are served by the e2e argument?
  - ISPs? End-users? Governments?
- How does a network operator influence the network without violating the e2e argument?
- Should the e2e argument apply to routing?
- Is middlebox functionality (NATs, firewalls) necessary, good, or evil?
- Is the e2e principle constraining innovation of the infrastructure?

What you said

"Such measures might be good for certain applications, as the probability of errors is decreased, but can be seen as unnecessary by other applications where the benefits brought by these efforts don't seem to outweigh the costs"

Duarte, João, Guilherme

"As Software-defined networks, que vamos abordar em PRD, representam uma grande disrupção desse princípio, colocando inteligência programável (…) Isto não implica contudo que a ideia do end-to-end principle esteja totalmente morta, como podemos ver pela proposta do QUIC"

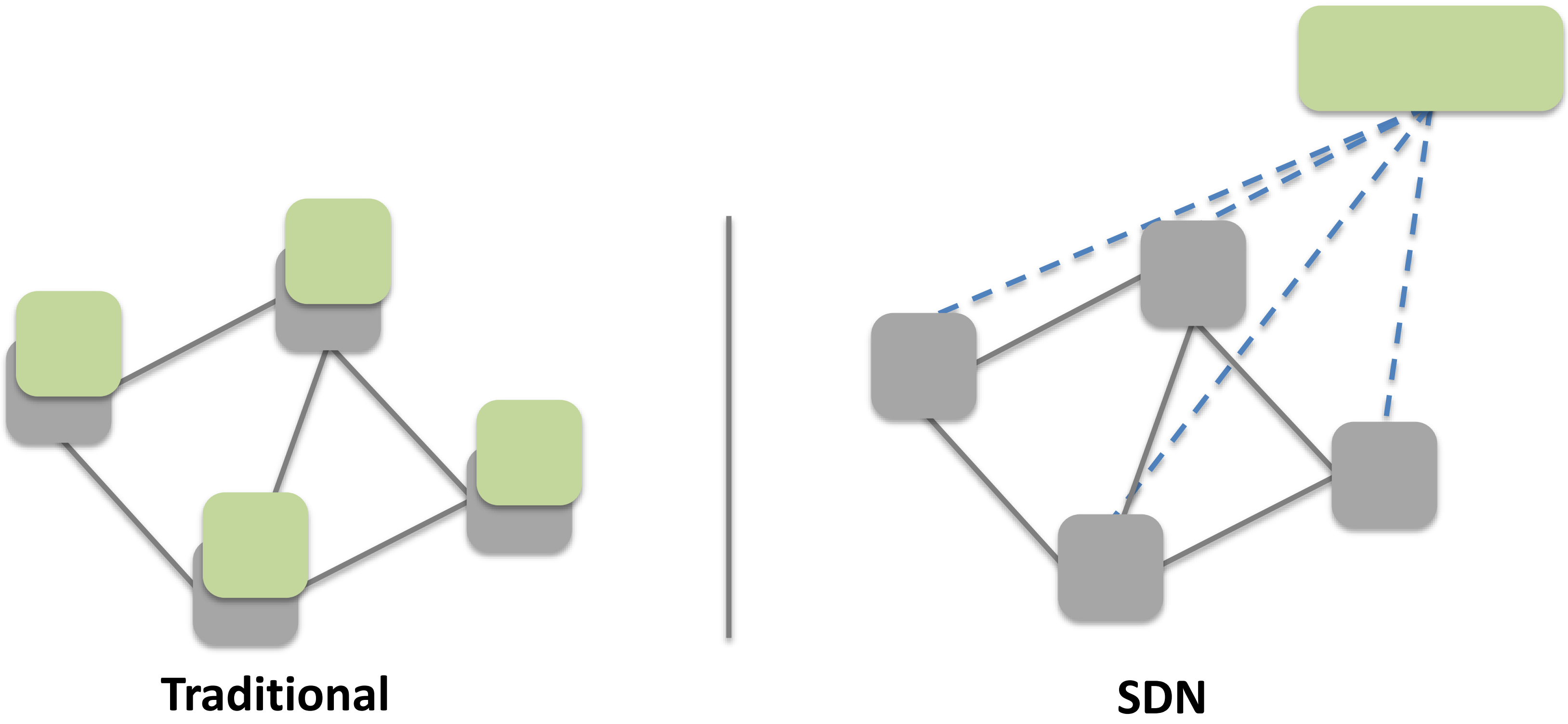
Diogo

Lecture plan

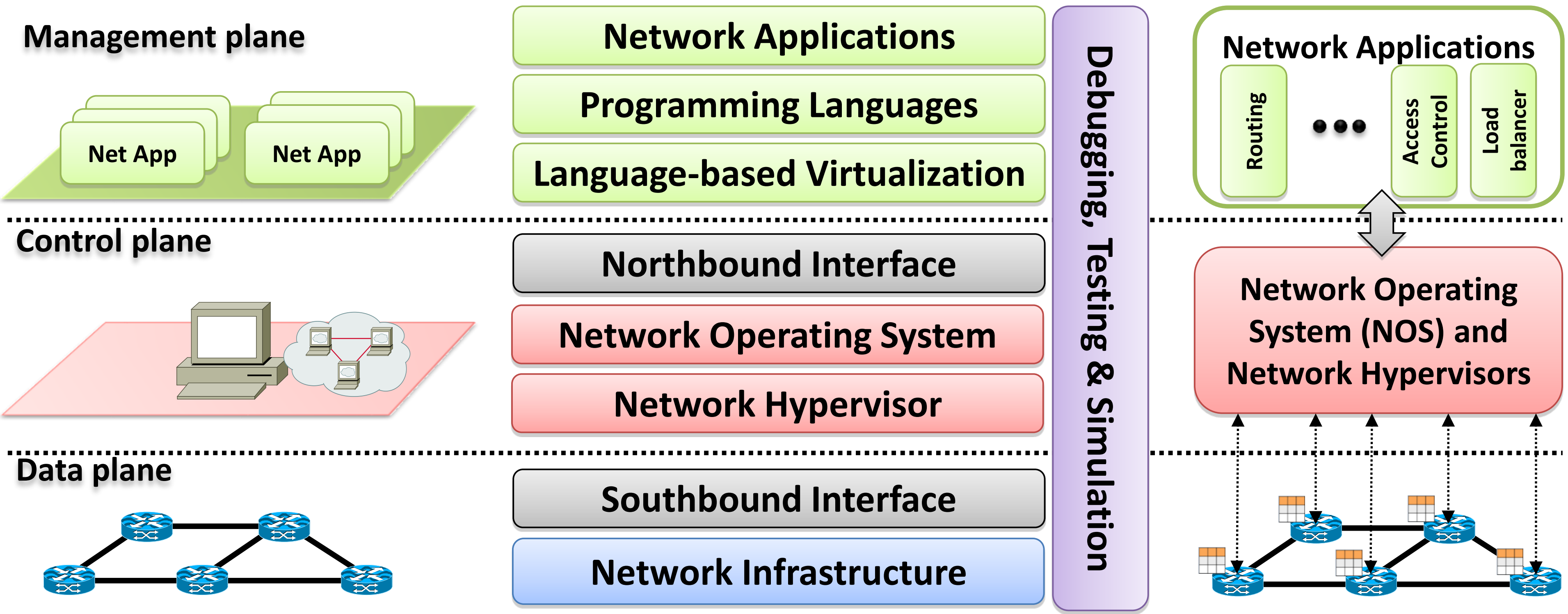
Next lecture: Software-Defined Networking

Software-Defined Networking

- Logical centralisation of control
- Network-wide visibility & direct control



Softwarization of networking



Next lecture: Software-Defined Networking

- **Mandatory (1 of these 3)**
  - T. Koponen et al., Onix: A Distributed Control Platform for Large-scale Production Networks, OSDI, 2010

  The first distributed, production-level SDN controller
  - S. Jain et al., "B4: experience with a globally-deployed software defined wan", SIGCOMM, 2013

  B4, the SDN architecture used by Google to connect its datacenters
  - C.-Y. Hong et al., "Achieving high utilization with software-driven WAN", SIGCOMM, 2013

  SWAN, the SDN solution used by Microsoft (similar to Google's B4)
- [Optional]
  - Scott Shenker, "The future of networking and the past of protocols", ONS, 2011
  - Nick McKeown, "How SDN will Shape Networking", ONS, 2011

  In these two videos Scott Shenker and Nick McKeown, two of the SDN luminaries, explain this new networking paradigm. Highly recommended for SDN starters!
  - F. Ramos et al., "Software-Defined Networks: On the Road to the Softwarization of Networking", Cutter IT journal, 2015
  - D. Kreutz et al., "Software-defined networking: A comprehensive survey", Proc of the IEEE, 2015

  A short and a long surveys on SDN

- [ppts]
  - S. Vissicchio et al., "Central Control Over Distributed Routing", SIGCOMM, 2015

  Centralised control using traditional routing protocols (i.e., non-SDN)
  - N. Kata, Ravana: controller fault-tolerance in software-defined networking, SOSR, 2015

  A consistent, fault-tolerant SDN controller

References

- [BriefHistory]
  - Leiner et al., "A Brief History of the Internet", Internet Society
- [Baran1964]
  - P. Baran, "On distributed communications networks", IEEE ToCS, 1964
- [Cerf1974]
  - V. Cerf, R. E. Kahn, "A Protocol for Packet Network Intercommunication", IEEE ToC, 1974
- [Saltzer1981]
  - J. H. Saltzer, D. P. Reed, D. D. Clark, "End-to-End Arguments in System Design", Distributed Computing Systems, 1981