

# Design Principles

I4-740: Fundamentals of Computer Networks  
Bill Nace

# Administrivia

No Paper Review for today

Next lesson: Review Norton2010

NOTE: Not Norton2003!

Do not review RFC 2901, but skim it

Paper review feedback (Clark88)

Make it short & sweet

No need to quote the paper

# Last Lecture

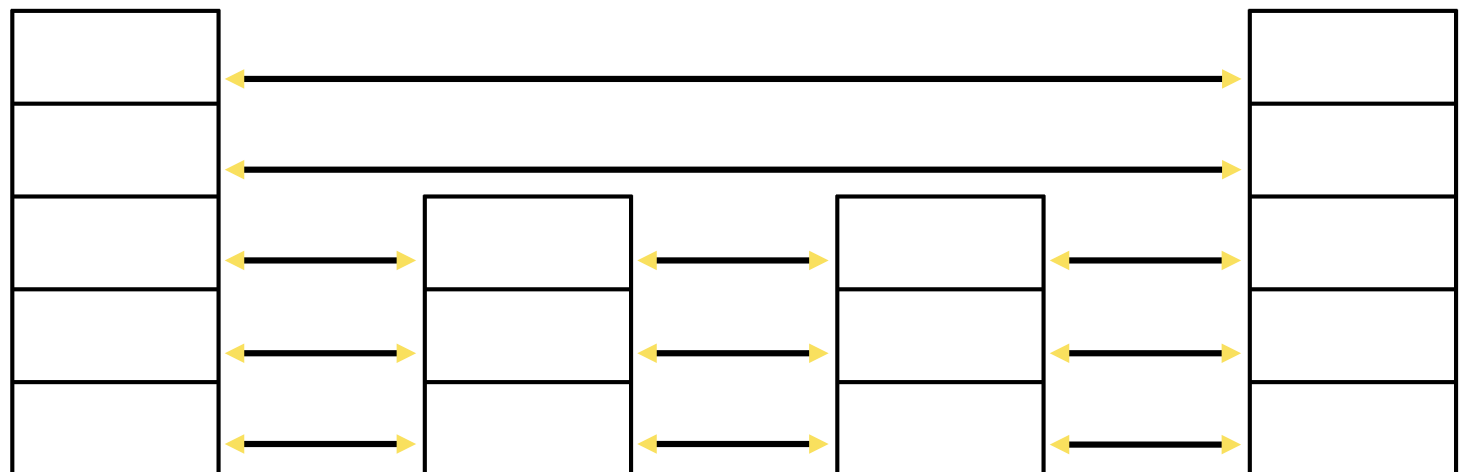
## Layered Architecture

Why layers?

What goes into each layer?

Service, interface, protocol

ISO OSI model (7-layer cake)

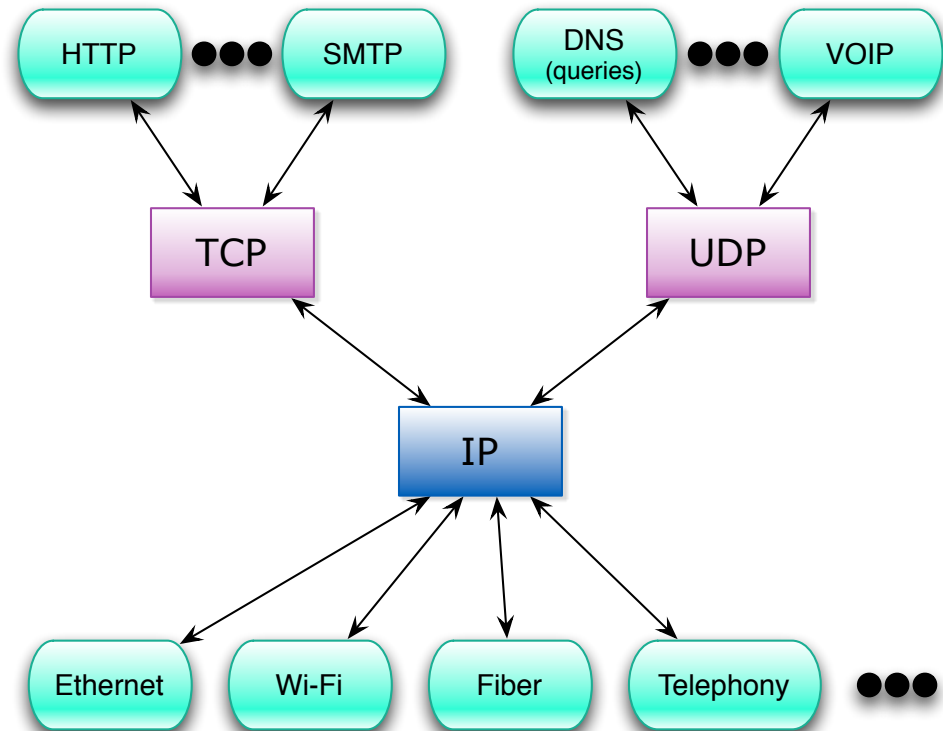


# Roadmap

- TCP/IP Architecture
  - Goals of TCP/IP Design
  - End-to-End argument

# TCP/IP Architecture

Distributed applications



Reliable stream service  
User datagram service

Best-effort connectionless  
packet transfer

Diverse network technologies

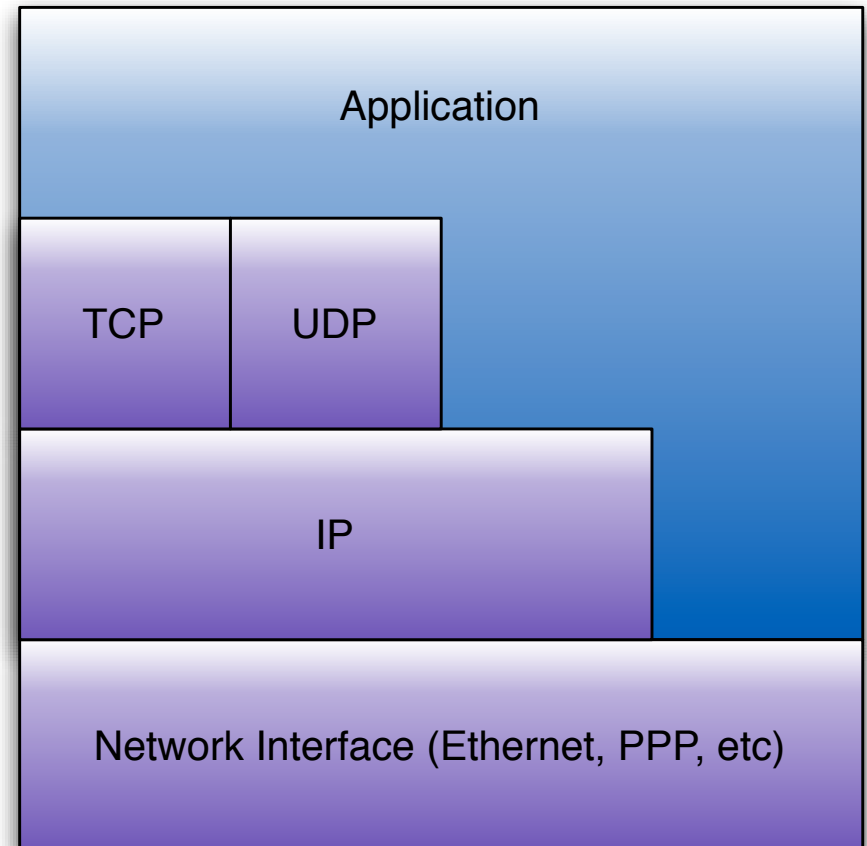
# Architecture Features: #1

Is strict layering implied?

No. Applications are free to bypass the defined transport layer and to directly use IP

or the data-link technology for that matter

or to build on top of regular applications (like HTTP, SSL)



# Architecture Features: #2

“Hourglass” figure!

Reflects the central philosophy of the Internet Architecture

IP over everything! IP is the glue

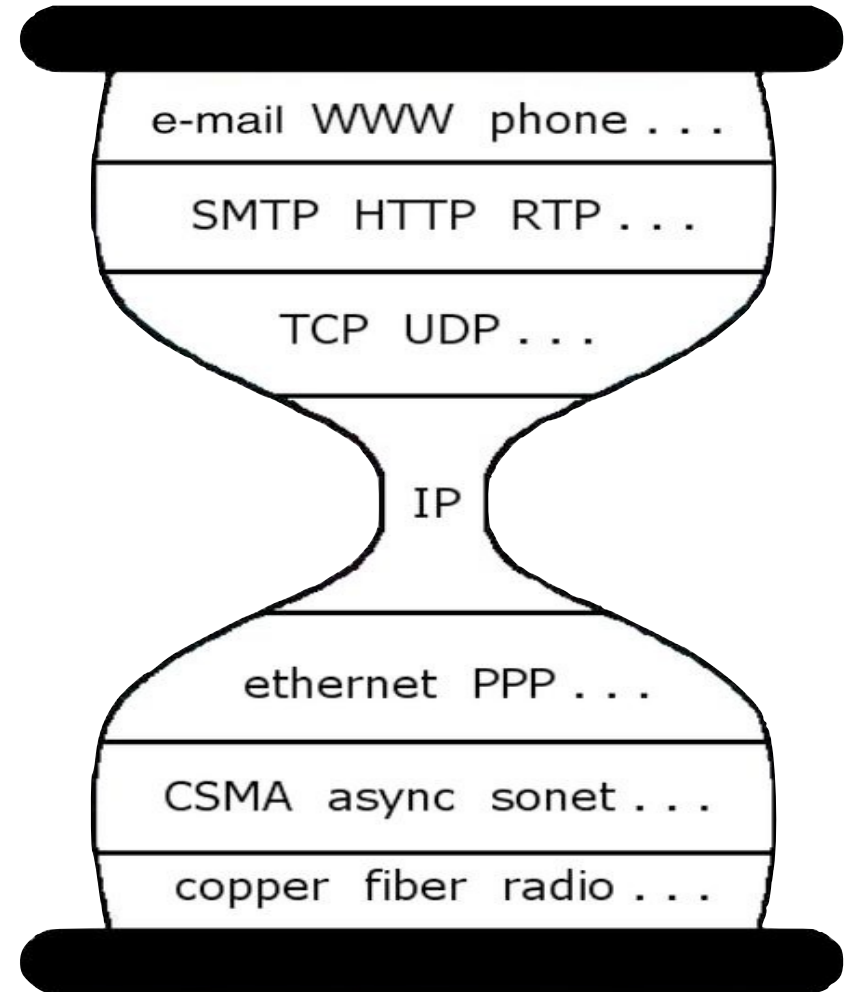
IP is a common method of exchanging packets over a wide collection of networks

Above IP: lots of stuff

Arbitrary many transport and application protocols

Below IP: lots of stuff

Arbitrarily many network technologies



# “Hourglass” importance

The narrow waist represents a minimal and carefully chosen set of global capabilities

Allows higher-level applications and lower-level communication technologies to co-exist, share capabilities and evolve rapidly

Critical to the Internet’s ability to adapt rapidly to new user demands and changing technologies

Drawback: Changing IP is difficult (see IPv6)



# Architecture Features: #3

Internet Engineering Task Force (IETF)

Standardization community for Internet protocols

For a new protocol to become standard:

A detailed protocol specification, beaten up by a “working group” of experts in the area

At least two independent implementations

Emphasis on “rough consensus and running code”

# Sidebar: Request for Comments

Every IETF standard is published as an RFC

Proposed / Draft standards

Must have at least 2 (why?!) independent implementations of each part of the standard

Internet standards

Experimental protocols

check out RFC 1149

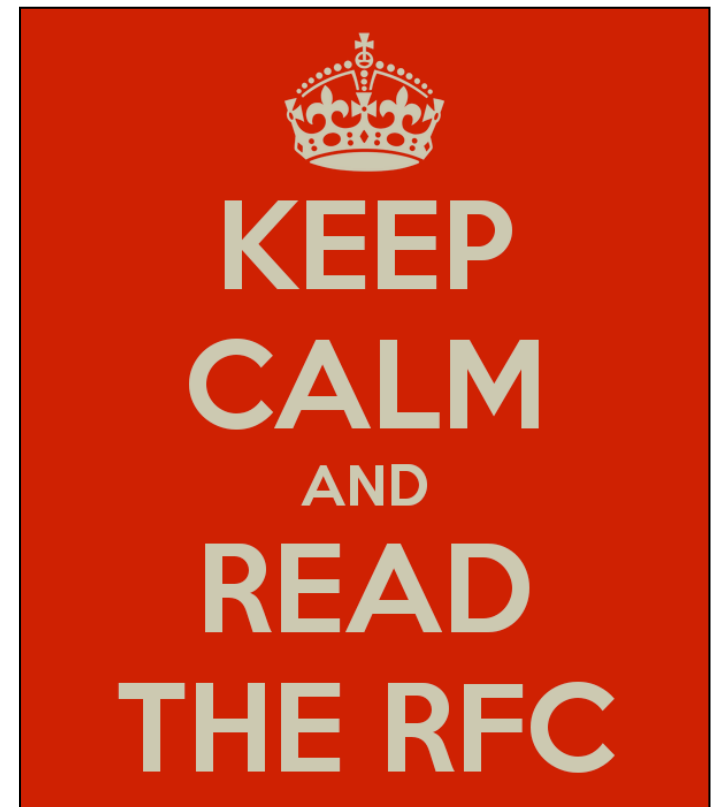
Available at [ietf.org/rfc/rfc1149.txt](http://ietf.org/rfc/rfc1149.txt)

Information documents

Historical standards

Not all RFCs are standards

What is special about RFC 2773?



# Roadmap

## TCP/IP Architecture

- Goals of TCP/IP Design

End-to-End argument

# Clark88 Paper

*Design Philosophy of the DARPA Internet Protocols*, David Clark, ACM Sigcomm, 1988

Attempts to give motivation and reasoning behind the TCP/IP architecture design

A list of goals, in order of priority

The goals were defined from the viewpoint of the U.S. government

# Assumptions

Each player in the Internet shares a common vision and sense of purpose

Build network infrastructure to hook computers together

Build applications to run on top

Not much thought on trustworthiness of entities

Not much thought on commercial players

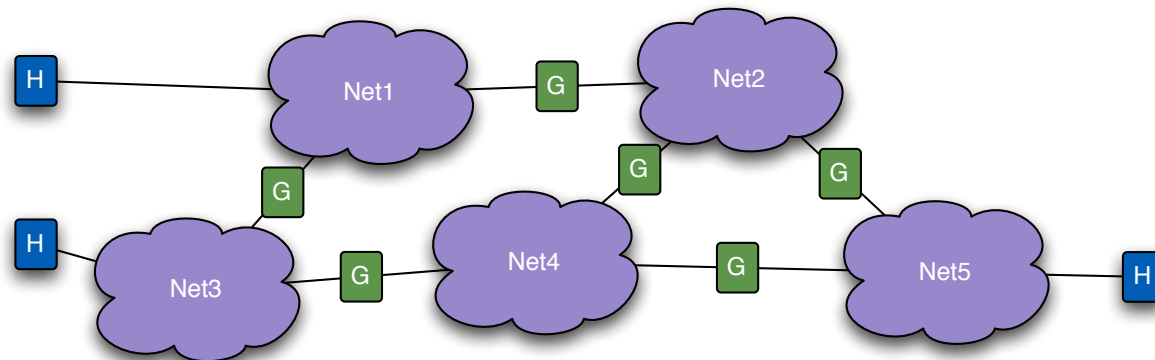
# Fundamental Goal

## *Interconnect existing networks*

The military must be able to communicate using radio and satellite, as well as wired networks

Needed to maintain separate administrative control for each network (think Army, Navy, etc)

Use well-understood technology: Store-and-forward packet-switching was known from ARPANET, add gateways to glue it together



## 2nd level goals: In order of importance

Continue despite loss of networks or gateways

Robustness against failure

Support multiple types of communication services

Connection-oriented vs connectionless

Accommodate a variety of networks – Ethernet, ATM

Permit distributed management of its resources

Network of networks

Be cost-effective (?)

Permit host attachment with a low level of effort

Rapid deployment of applications, services, networks

Account for resources used (?)

# Robustness Against Failures

Architecture needs to store state of a conversation

Answers “Where were we?” after an disruption

Which data was received, which was lost

Use this state to recover from loss of synchronization

Where should this state be stored?

Approach #1: In the intermediate nodes

Approach #2: At the end-hosts

What are the pros and cons of the two approaches?



# Where should I leave my breadcrumbs?

Approach #1: In the intermediate nodes

Requires replication of state everywhere

Difficult to build algorithms

Synchronization issues

Approach #2: At the end-hosts

“Fate sharing”

If the end-host is lost, the state associated with its conversation is also lost

Much easier to engineer than replication



# Stateless Network Core

IP uses approach #2

Intermediate nodes (routers, gateways, switches, etc) do not have any state information about on-going connections

Stateless nodes

Also called a datagram network

Datagram is another word for packet

End-hosts ensure reliable delivery of data (if they care)

# Intelligence of Internet Devices

Dumb routers (or networks)

Really only supposed to do one thing and one thing fast – forwarding packets

Do not need to know much else!

Smart hosts (or machines)

Responsible for a lot more

Reliability, flow control, congestion control, buffering ...

Why is this changing today?

Next lecture on the provider and customer relationships between ISPs and companies

# Types of Service

Originally, TCP was thought to be enough

Audio conferencing (i.e. digitized speech) does not need strict reliability guarantee

Found out reliability is the main source of delay! Must wait for a lost packet to be retransmitted before sending rest of the received packets in sequence

# Types of Service (2)

Originally, TCP and IP were at the same layer

Decided to separate them

IP as the basic datagram building block

UDP was viewed as a building block for construction of multiple services

A very wise decision indeed

# Distributed Management

Mostly achieved

Networks can be administered separately

Network can run different internal routing protocols

Original work included manual setting of routing tables!

# Today's Management

Improved with the introduction of BGP

But managing routing policies is still a big issue in today's Internet

Route hijacking: a network announces someone else's routes to attract traffic, a form of denial-of-service attack

Misconfigurations are difficult to detect and recover from

# Accounting of Resources

Last on the list of goals

Little (or no attention) in the original design

Virtually no tools at that time – the U.S. military did not need accounting!



# Today's Accounting

Coarse-grained tools are widely in use

Bits-per-second, packet-per-second  
granularity

Recent tools measure and monitor traffic  
based on flows, e.g. Netflow

Source/Destination pairs (or prefixes)

Research in inferring traffic matrices

# Roadmap

TCP/IP Architecture

Goals of TCP/IP Design

- End-to-End argument

# Saltzer84 Paper

J.H. Saltzer, D.P. Reed, and D.D. Clark, *End-to-End Arguments in System Design*, ACM Transactions on Computer Systems (4):277-88, Nov 1984

States the “End-to-End” Argument

A function that can be implemented at the application layer (i.e. end-points) should not be in the lower layers (i.e. network core)

# End-to-End Arguments

Eliminate duplication of function across layers

Allows for more innovation

Core should not be tailored to any specific application

Shouldn't inhibit deployment of new things

Reliability

“The simpler the core, the more reliable it is”

# Exception: Performance Optimization

A lower layer could implement a higher layer function if it is for improving performance

Example: Wireless networks have very high loss rate

Inefficient to wait for application to realize a packet is lost and signal retransmit

The request for retransmit message may also be lost

Add reliability to the link layer

Most link layer technologies implement some sort of error correction mechanism

**WARNING:** This is a complex design tradeoff

# How about inventing a New Internet Architecture?

How would you change the list of goals or priorities? Why?

Do the design principles we covered still apply?

See what David Clark says 15 years later

# 2002 Paper

*Tussle in Cyberspace: Defining Tomorrow's Internet*, David Clark et al, 2002

Revisiting old principles

Suggesting new design principles

Tussles: “Different stakeholders have adverse interests, each vie to favor their particular positions”

# Moving Away?

Operation in an untrustworthy world

Firewalls, content filters, DMZ, etc

More demanding applications (YouTube)

Implement using caches placed close to end users

Dramatic growth of end-hosts

Refrigerators need an IP address? IPv4 has few

Network Address Translation (NAT) boxes

Map  $N$  public IP addresses to  $M$  end-hosts (connections) behind NAT, where  $M \gg N$



# Moving Away from End-to-End

ISP service differentiation

Keep enhanced services within the ISP's network, as competitive differentiator

Less sophisticated users

Prefer simpler end hosts without constant need of software configuration, upgrade and maintenance

Proliferation of non-“computer” end hosts

# Routers are getting Smarter

Quality of Service (QoS)

Routers forwards certain packets before others

Software Defined Networking (SDN)

Router is “programmed” not just protocols

Virtual Private Networks (VPNs)

Accounting

Not designed into the architecture

Management (ditto)

# Lesson Objectives

Now, you should be able to:

describe the internet's layered architecture according to the TCP/IP model and argue the importance of architectural features of that model

explain the role of the IETF in the internet's operation

describe the goals of the TCP/IP design according to Clark88 and identify the fundamental goal

# Now, you should be able to:

explain the ramifications of the robustness, multiple services and multiple networks goals on the design of the TCP/IP protocols

explain Saltzer84's "End-to-End" argument. Be able to describe exceptional situations where it may not apply as well as ways in which the modern internet may be moving away from this design philosophy

# Summary

TCP/IP Architecture

Not strict layering

“Hourglass”

IETF

Goals of TCP/IP Design

End-to-End argument

Is it still applicable?