

w8.1- Application Layer

w8.1- Application Layer.mp4 — Haruna Media Player

Application Layer

The diagram illustrates the TCP/IP stack structure. It consists of four horizontal layers stacked vertically. The top layer is yellow and labeled "Application Layer" with the subtext "Web, E-Mail, File Transfer". Below it is a black layer labeled "Transport Layer (TCP)" with the subtext "Reliable Connections". The third layer is orange and labeled "Internetwork Layer (IP)" with the subtext "Simple, Unreliable". The bottom layer is red and labeled "Link Layer (Ethernet, WiFi)" with the subtext "Physical Connections". Three yellow arrows point from the left towards the Transport, Internetwork, and Link layers. Below the diagram, a caption reads: "Layer and the TCP Layer. And, and, and each of these layers works".

Application Layer
Web, E-Mail, File Transfer

Transport Layer (TCP)
Reliable Connections

Internetwork Layer (IP)
Simple, Unreliable

Link Layer (Ethernet, WiFi)
Physical Connections

Layer and the TCP Layer.
And, and, and each of these layers works

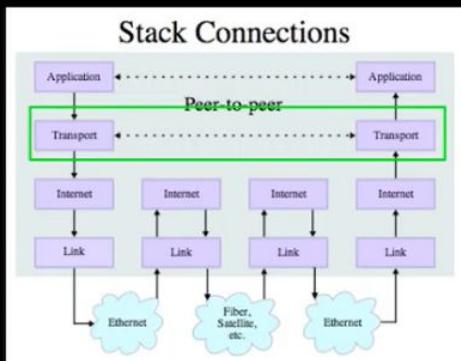
00:00:12 / 00:25:13

65

Subtitle scale: 0.3

Quick Review

- Link layer: gets the data onto the link, and manages collisions on a single hop
- Internet layer: moves the data over one hop, trying to get it “closer” to its destination
- Transport layer: Assumes that the internet layer may lose data, so request retransmission when needed— provides a nice reliable pipe from source to destination



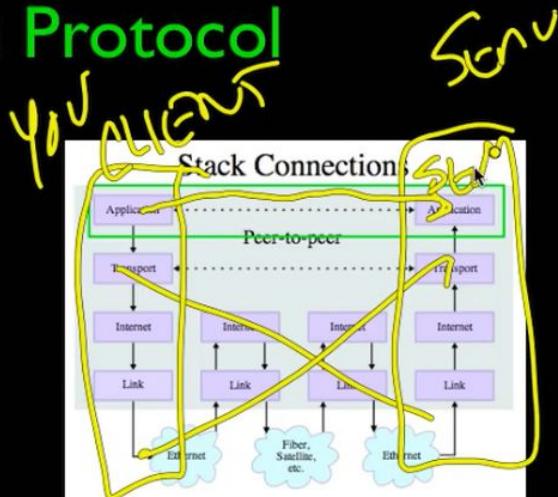
Source: http://en.wikipedia.org/wiki/Internet_Protocol_Suite
basically to give us a reliable, sequenced end to end stream, that can



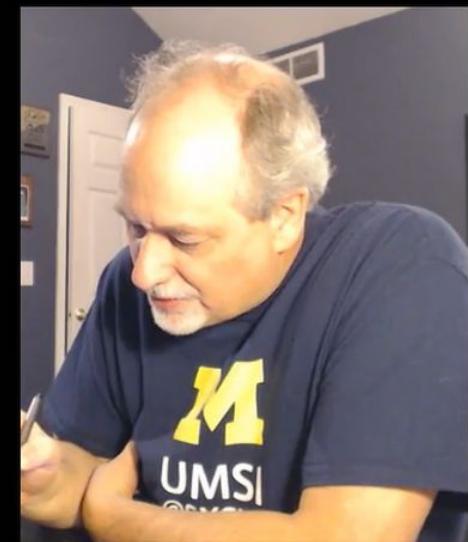
w8.1- Application Layer.mp4 — Haruna Media Player

Application Protocol

- Since TCP gives us a reliable pipe, what do we want to do with the pipe? What problem do we want to solve?
 - Mail
 - World Wide Web
 - Stream kitty videos



Source: <http://en.wikipedia.org/wiki/>
And this application is the server application.



w8.1- Application Layer.mp4 — Haruna Media Player

Two Questions for the Application Layer

- Which application gets the data?
 - Ports
- What are the rules for talking with that application?
 - Protocols

http://en.wikipedia.org/wiki/TCP_and_UDP_port So, there are two basic questions
http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers that the application layer has to solve.



w8.1- Application Layer.mp4 — Haruna Media Player

Ports

- Like extensions in a phone number
- The IP address network number (the area code) gets to the LAN
- The IP address host number (the telephone number) gets you to the destination machine
- The port (the extension) gets you to a specific application

(734) 764 1855, ext. 27

141.211.144.188
Port 25

the internet, and then a port within that tells us what application we're going to

00:04:04 / 00:25:13

65

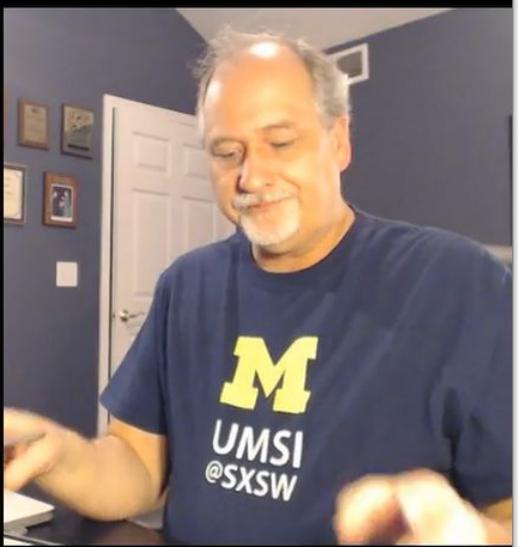
w8.1- Application Layer.mp4 — Haruna Media Player

TCP, Ports, and Connections

http://en.wikipedia.org/wiki/TCP_and_UDP_port

http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers

talk to.
So, let's talk about ports and



00:04:07 / 00:25:13

65

w8.1- Application Layer.mp4 — Haruna Media Player

The diagram illustrates the Application Layer of network communication. On the left, a server box labeled 'www.umich.edu' contains several service ports: 25 (Incoming E-Mail), 23 (Login), 80 (Web Server), and 110 (Personal Mail Box). On the right, a client computer with the IP address '74.208.28.177' has a port 443 (secure web server) active. A red arrow points from the client's port 443 to the server's port 80. A yellow arrow points from the client's port 443 to the server's port 25. Another yellow arrow points from the client's port 443 to the server's port 110. A text box on the client side says 'Please connect me to the secure web server (port 443)'. Below the diagram, a caption reads: 'And, we have many services running on this server like sending email, or'. A video player interface at the bottom shows the video is at 00:04:29 / 00:25:13, and the slide number is 65.

Clipart: <http://www.clerk.com/search/networksym/l>

00:04:29 / 00:25:13

65

w8.1- Application Layer.mp4 — Haruna Media Player

Common TCP Ports

- Telnet (23) - Login
- SSH (22) - Secure Login
- HTTP (80)
- **HTTPS (443) - Secure**
- SMTP (25) (Mail)
- IMAP (143/220/993) - Mail Retrieval
- POP (109/110) - Mail Retrieval
- DNS (53) - Domain Name
- FTP (21) - File Transfer

http://en.wikipedia.org/wiki/Some_common_TCP_ports are the ones I just mentioned.



00:05:09 / 00:25:13 65

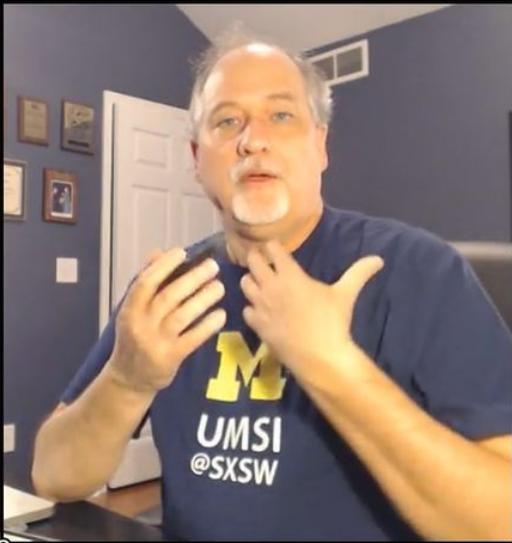
w8.1- Application Layer.mp4 — Haruna Media Player

Application Protocols

<http://en.wikipedia.org/wiki/Http>

<http://en.wikipedia.org/wiki/Pop3>

Now, once we have a connection to the web server, or to the mail server, or to the



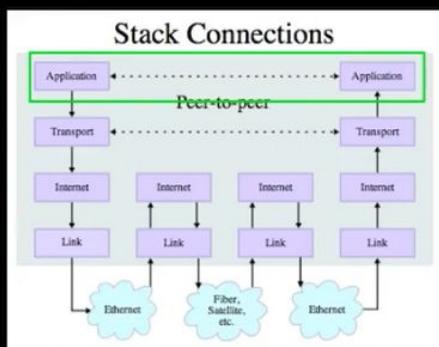
00:05:18 / 00:25:13

65

w8.1- Application Layer.mp4 — Haruna Media Player

Application Protocol

- Since TCP gives us a reliable pipe, what do we want to do with the pipe? What problem do we want to solve?
 - Mail
 - World Wide Web
 - Stream kitty videos



Source: <http://en.wikipedia.org/wiki/>
server that we desire to connect to by
using ports, and the question is, what



w8.1- Application Layer.mp4 — Haruna Media Player

HTTP - Hypertext Transport Protocol

- The dominant Application Layer Protocol on the Internet
- Invented for the Web - to Retrieve HTML, Images, Documents etc
- Extended to be data in addition to documents - RSS, Web Services, etc..
- Basic Concept - Make a Connection - Request a document - Retrieve the Document - Close the Connection

<http://en.wikipedia.org/wiki/HTTP> as to what's going on.
So the world wide web clients, and the



w8.1- Application Layer.mp4 — Haruna Media Player

Subtitle scale: 0.3

HTTP Request / Response Cycle

The diagram illustrates the HTTP Request/Response cycle. At the top, a purple box labeled "Web Server" has a yellow arrow pointing upwards labeled "HTTP Request". Below it, a purple box labeled "Browser" (with "Internet Explorer, FireFox, Safari, etc." written underneath) has a green arrow pointing downwards labeled "HTTP Response". A yellow box at the bottom left contains the text "Hello there my name is Chuck" and "Go ahead and click on here.". A yellow arrow points from this box to the "Browser" box. A green arrow points from the "Browser" box to a screenshot of a web browser window on the right, which displays a page with text and images.

http://www.oreilly.com/openbook/cgi/ch04_02.html

Okay?
And so the way it works, is, you're in a
Source: http://www.dr-chuck.com/

00:07:00 / 00:25:13

65

w8.1- Application Layer.mp4 — Haruna Media Player

The image shows a video player interface with a dark background. At the top, it says "w8.1- Application Layer.mp4 — Haruna Media Player". Below this is a video frame showing a man with grey hair, wearing a blue t-shirt with a large yellow 'M' on it, sitting at a desk and talking. To the left of the video frame is a screenshot of a web browser window. The browser has a title bar "www.dr-chuck.com/page1" and a URL bar "www.dr-chuck.com/page1.htm". The main content area of the browser shows the text "The First Page". Below this, there is a smaller text block that reads "So here is a canonical webpage.
So here's a browser, here's the URL that". The bottom of the video player has a progress bar, a timestamp "00:07:39 / 00:25:13", and a page number "65".

www.dr-chuck.com/page1

www.dr-chuck.com/page1.htm

H Tw ViDSR.COM: Six Of CT Liberal Arts and Scic CRsera I8 Speed C2G G2G@UM Other Bookmarks

The First Page

If you like, you can switch to the [Second Page](#).

So here is a canonical webpage.
So here's a browser, here's the URL that

00:07:39 / 00:25:13 65

w8.1- Application Layer.mp4 — Haruna Media Player

The image shows a video player interface with a progress bar at the bottom. The video title is "w8.1- Application Layer.mp4 — Haruna Media Player". The video frame on the right shows a man with a beard and short hair, wearing a dark blue t-shirt with a large yellow 'M' and the text "UNSI @SYC". He is gesturing with his right hand near his face. The background behind him is a room with blue walls and framed pictures. The browser window on the left displays a webpage from "www.dr-chuck.com/page2.htm". The page has a large black header "The Second Page" and a paragraph of text below it. A small caption at the bottom of the browser window reads: "and showed me the document. So that's the contents of the second". The browser's address bar shows "www.dr-chuck.com/page2.htm". The toolbar above the browser includes icons for Home, Back, Forward, Stop, Refresh, and various bookmarks.

[First Page](#)

and showed me the document.
So that's the contents of the second

00:08:33 / 00:25:13

65

w8.1- Application Layer.mp4 — Haruna Media Player

HTTP Request / Response Cycle

The diagram illustrates the HTTP Request/Response cycle. A purple box labeled "Web Server" is at the top, and a purple box labeled "Browser" is at the bottom. Two green arrows connect them: an upward-pointing arrow labeled "HTTP Request" and a downward-pointing arrow labeled "HTTP Response". To the left of the browser, there is a screenshot of a web page with the text: "Hello there my name is Chuck" and "Go ahead and click on [here](#)". A yellow arrow points from this screenshot up towards the browser. To the right of the browser, there is a screenshot of a web browser window displaying a page with the text: "<head> .. </head>" and "<body> Welcome to my application</h1> ... </body>". Below the browser, the URL "http://www.oreilly.com/openbook/cgi/ch04" is shown, followed by the text "Remember that this part here's the source: http://www.dr-chuck.com/internet right there."

GET /page2.html

Web Server

HTTP Request

HTTP Response

Browser

Hello there my name is Chuck
Go ahead and click on [here](#).

Internet Explorer,
FireFox, Safari, etc.

<head> .. </head>
<body>
<h1>Welcome to my
application</h1>
...
</body>

<http://www.oreilly.com/openbook/cgi/ch04> Remember that this part here's the
Source: <http://www.dr-chuck.com/internet> right there!

00:09:16 / 00:25:13

65

w8.1- Application Layer.mp4 — Haruna Media Player

Internet Standards

- The standards for all of the Internet protocols (inner workings) are developed by an organization
- Internet Engineering Task Force (IETF)
- www.ietf.org
- Standards are called “RFCs” - “Request for Comments”

Network Working Group
Request for Comments: 1945
Category: Informational

T. Berners-Lee
M. Frystyk
R. Fielding
UC Irvine
H. Frysztak
MIT/LCS
May 1996

Hypertext Transfer Protocol -- HTTP/1.0

Status of This Memo

This memo provides information for the Internet community. This memo does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

IESG Note:

The IESG has concerns about this protocol, and expects this document to be replaced relatively soon by a standards track document.

Abstract

The Hypertext Transfer Protocol (HTTP) is an application-level protocol with the lightness and speed necessary for distributed, collaborative, hypermedia information systems. It is a generic, stateless, request-response protocol used for transfer of data, such as name servers and distributed object management systems, through extension of its request methods (commands). A feature of HTTP is the typing of data representation, allowing systems to be built independently of the data being transferred.

Source: <http://www.ietf.org/rfc/rfc1945.txt>
Well, we would go back to the IETF, the Internet Engineering Task Force, and we



00:09:53 / 00:25:13

65

w8.1- Application Layer.mp4 — Haruna Media Player

The video player window has a dark background. On the left, there is a white rectangular area containing text from a presentation slide. On the right, there is a video feed of a man with a beard and short hair, wearing a dark blue t-shirt with a large yellow 'M' and the text 'UMSI @SYU'. He is looking down at something in his hands. The video player interface includes a title bar at the top, a progress bar at the bottom, and a timestamp '00:10:41 / 00:25:13' in the bottom right corner.

5.1.2 Request-URI

The Request-URI is a Uniform Resource Identifier (Section 3.2) and identifies the resource upon which to apply the request.

Request-URI = absoluteURI | abs_path

The two options for Request-URI are dependent on the nature of the request.

The absoluteURI form is only allowed when the request is being made to a proxy. The proxy is requested to forward the request and return the response. If the request is GET or HEAD and a prior response is cached, the proxy may use the cached message if it passes any restrictions in the Expires header field. Note that the proxy may forward the request on to another proxy or directly to the server specified by the absoluteURI. In order to avoid request loops, a proxy must be able to recognize all of its server names, including any aliases, local variations, and the numeric IP address. An example Request-Line would be:

GET http://www.w3.org/pub/WWW/TheProject.html HTTP/1.0

That is what you're supposed to send down this connection.

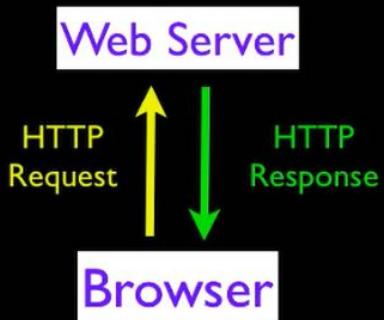
00:10:41 / 00:25:13

65

“Hacking” HTTP

```
Last login:Wed Oct 10 04:20:19 on ttyp2  
si-csev-mbp:~ csev$ telnet www.dr-chuck.com 80  
Trying 74.208.28.177...  
Connected to www.dr-chuck.com.  
Escape character is '^]'.  
GET http://www.dr-chuck.com/page1.htm  
<h1>The First Page</h1>  
<p>  
If you like, you can switch to the  
<a href="http://www.dr-chuck.com/page2.htm">  
Second Page</a>  
</p>
```

But what we're going to do, is we're
Port 80 is the non-encrypted HTTP port
going to fake being a web browser, we're



w8.1- Application Layer.mp4 — Haruna Media Player

The video player interface displays two side-by-side frames. The left frame shows a terminal window titled 'csev — telnet — 80x24' with the following text output:

```
Last login: Fri Mar 29 16:04:15 on ttys000
You have mail.
In .bash_profile
cseverancembp:~ csev$ telnet www.dr-chuck.com 80
Trying 74.208.28.177...
Connected to www.dr-chuck.com.
Escape character is '^]'.
YO WHAT IS UP?
DUDE
```

The right frame shows a man with a beard and mustache, wearing a blue t-shirt with a large yellow 'M' and the words 'UNIVERSITY OF MICHIGAN'. He is looking down and to his right with a thoughtful expression. The background shows a wall with several framed certificates or awards.

protocol, so I do not know that I'm supposed to type the word Get here.

00:13:23 / 00:25:13 65

w8.1- Application Layer.mp4 — Haruna Media Player

The video player interface shows a terminal window on the left and a video feed of a man on the right.

Terminal Window Content:

```
bash
Escape character is '^]'.
YO WHAT IS UP?
DUDE

HTTP/1.1 400 Bad Request
Date: Sat, 30 Mar 2013 01:03:47 GMT
Server: Apache
Content-Length: 296
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
Request header field is missing ':' separator.<br />
<pre>
DUDE</pre>
</p>
</body></html>
Connection closed by foreign host.
cseverancembp:~ csev$
```

Video Feed:

A man with a beard and grey hair, wearing a dark blue t-shirt with a large yellow 'M' on it, is speaking. Subtitles at the bottom of the video feed read:

I'm going to type yo, what is up Dude.
And it goes, you are not from my country.

00:13:34 / 00:25:13 65

```
chiun@chiun-ThinkPad-T14s-Gen-2a: ~          Q  ⌂  -  □  ×  
telnet is already the newest version (0.17+2.5-3ubuntu4.1).  
telnet set to manually installed.  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
chiun@chiun-ThinkPad-T14s-Gen-2a:~$ telnet google.com 80  
Trying 2a00:1450:4001:814::200e...  
Connected to google.com.  
Escape character is '^]'.  
YO WHAT IS UP  
DUDE  
  
HTTP/1.0 400 Bad Request  
Content-Type: text/html; charset=UTF-8  
Referrer-Policy: no-referrer  
Content-Length: 1555  
Date: Thu, 26 Feb 2026 16:25:56 GMT  
  
<!DOCTYPE html>  
<html lang=en>  
  <meta charset=utf-8>  
  <meta name=viewport content="initial-scale=1, minimum-scale=1, width=device-width">  
  <title>Error 400 (Bad Request)!!1</title>  
  <style>  
    *{margin:0;padding:0}html,code{font:15px/22px arial,sans-serif}html{background
```

w8.1- Application Layer.mp4 — Haruna Media Player

The video player interface shows a terminal window on the left and a video feed of a man on the right.

Terminal Window (Left):

```
cseverancembp:~ csev$ telnet www.dr-chuck.com 80
Trying 74.208.28.177...
Connected to www.dr-chuck.com.
Escape character is '^]'.
GET http://www.dr-chuck.com/page1.htm HTTP/1.0

HTTP/1.1 200 OK
Date: Sat, 30 Mar 2013 01:05:57 GMT
Server: Apache
Last-Modified: Sat, 27 Dec 2008 04:09:43 GMT
ETag: "19c6b9-83-45eff694a6fc0"
Accept-Ranges: bytes
Content-Length: 131
Connection: close
Content-Type: text/html

<h1>The First Page</h1>
<p>
If you like, you can switch to the
<a href="http://www.dr-chuck.com/page2.htm">
Second Page</a>.
</p>
Connection closed by foreign host.
cseverancembp:~ csev$
```

Video Feed (Right):

A man with a beard and grey hair, wearing a blue t-shirt with a large yellow 'M' and the text 'UMSI @SXSW', is speaking. A caption at the bottom of the video feed reads: "like you what the time is, when this file was last modified."

00:15:46 / 00:25:13 65

```
chiun@chiun-ThinkPad-T14s-Gen-2a:~
```

```
}
```

```
}
```

```
}
```

```
chiun@chiun-ThinkPad-T14s-Gen-2a:~$ GET http://www.dr-chuck.com/page1.htm
```

```
HTTP/1.0
```

```
<h1>The First Page</h1>
```

```
<p>
```

```
If you like, you can switch to the
```

```
<a href="http://www.dr-chuck.com/page2.htm">
```

```
Second Page</a>.
```

```
</p>
```

```
<!DOCTYPE html><html lang="en" data-adblockkey=MFwwDQYJKoZIhvcNAQEBBQADSw
```

```
AwSAJBANnylw2vLY4hUn9w06zQKbhKBfvjFUCsdFlb6TdQhxb9RXWXuI4t31c+o8fY0v/s8q1
```

```
LGPga3DE1L/tHU4LENMCAwEAAQ==_gnUV3Zz+fevRyebMVaVSB/GWmZqrtzLsqLoD3pEZ9q6mp
```

```
yZp/v6cMlX0u73QfaN9NkW09xNLt9CKq1eR7CQ7EA==><head><meta charset="utf-8"><t
```

```
itle>http.ca- -&nbsp;Diese Website steht zum Verkauf!&nbsp;-&nbsp;Info
```

```
rmationen zum Thema http.</title><meta name="viewport" content="width=devi
```

```
ce-width,initial-scale=1.0,maximum-scale=1.0,user-scalable=0"><meta name="
```

```
description" content="Diese Website steht zum Verkauf! http.ca ist Ihre er
```

```
ste und beste Informationsquelle über http. Hier finden Sie auch weitere i
```

```
nteressante Links. Wir hoffen, dass Sie bei Ihrer Suche erfolgreich sind!"
```

w8.1- Application Layer.mp4 — Haruna Media Player

The image shows a video player interface with a terminal window on the left and a man on the right.

Terminal Window (Left):

```
cseverancembp:~ csev$ telnet www.dr-chuck.com 80
Trying 74.208.28.177...
Connected to www.dr-chuck.com.
Escape character is '^A'.
GET http://www.dr-chuck.com/page1.htm HTTP/1.0

HTTP/1.1 200 OK
Date: Sat, 30 Mar 2013 01:05:57 GMT
Server: Apache
Last-Modified: Sat, 27 Dec 2008 04:09:43 GMT
ETag: "19c6b9-83-45eff694a6fc0"
Accept-Ranges: bytes
Content-Length: 131
Connection: close
Content-Type: text/html

<h1>The First Page</h1>
<p>
If you like, you can switch to the
<a href="http://www.dr-chuck.com/page2.htm">
Second Page</a>.
</p>
Connection closed by foreign host.
cseverancembp:~ csev$
```

Man on the Right:

If we comply with the server's request,
we can talk to it.

00:16:35 / 00:25:13 65

Accurate Hacking in the Movies

- Matrix Reloaded
- Bourne Ultimatum
- Die Hard 4
- ...

<http://nmap.org/movies.html> (scroll down for video)
Or search YouTube for "Trinity hacking scene"

hacking software, its called Nmap, which stands for network map.



w8.1- Application Layer.mp4 — Haruna Media Player

http://nmap.org/movies.html (scroll down for video)
Or search YouTube for "Trinity hacking scene"

the nmap people will know it or maybe you
just have to search for the Trinity

00:19:06 / 00:25:13 65

w8.1- Application Layer.mp4 — Haruna Media Player

Application Layer Summary

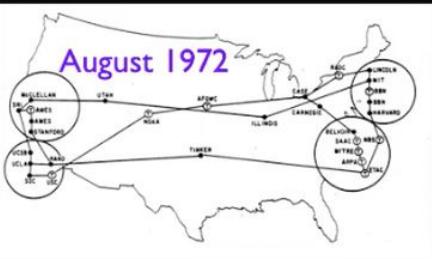
- We start with a “pipe” abstraction - we can send and receive data on the same “socket”
- We can optionally add a security layer to TCP using SSL - Secure Socket Layer (aka TLS - Transport Layer Security)
- We use well known “port numbers” so that applications can find a particular application *within* a server such as a mail server, web service, etc

legitimately hack a web server.
So, [COUGH] basically the application



w8.1- Application Layer.mp4 — Haruna Media Player

The Architecture of the Internet



A map of the United States showing the first four nodes of the ARPANET. The nodes are labeled: SRI, UCBB, UCB, and STANFORD. They are interconnected by lines representing the network links. The date "August 1972" is written in blue at the top left of the map.

Application Layer
Web, E-Mail, File Transfer

Transport Layer (TCP)
Reliable Connections

Internet Layer (IP)
Simple, Unreliable

Link Layer (Ethernet, WiFi)
Physical Connections

wireless, all of the wonderful things
that happen there.

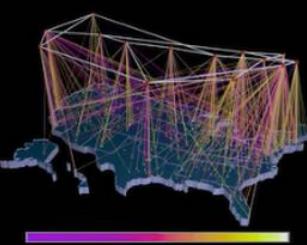


00:19:54 / 00:25:13

65

w8.1- Application Layer.mp4 — Haruna Media Player

The Architecture of the Internet



Application Layer Web, E-Mail, File Transfer
Transport Layer (TCP) Reliable Connections
Internetwork Layer (IP) Simple, Unreliable
Link Layer (Ethernet, WiFi) Physical Connections

still present with us today.
[COUGH] As the NSF net came out, these



00:20:31 / 00:25:13 65

w8.1- Application Layer.mp4 — Haruna Media Player

The video player interface shows a graph titled "Harries' Internet Timeline" with data from Robert H. Zakon. The graph plots the number of hosts (#Hosts) against time (Date). The data shows a exponential increase in the number of hosts over time, starting from 1985 and reaching approximately 800 million by 2010. A table below the graph provides specific data points.

DATE	HOSTS	DATE	HOSTS
12/69	4	10/84	1,024
06/70	9	10/85	1,961
10/70	11	02/86	2,308
12/70	13	11/86	5,089
04/71	23	12/87	28,174
10/72	31	07/88	33,000
01/73	35	10/88	56,000
06/74	62	07/89	130,000
03/77	111	10/89	159,000
12/79	188	10/90	313,000
08/81	213	10/91	617,000
05/82	235	10/92	1,136,000
08/83	562	10/93	2,056,000

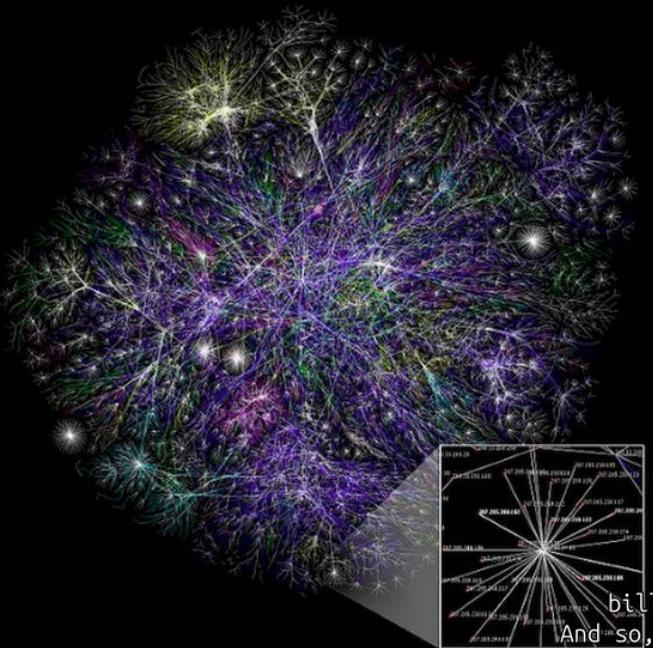
The video also displays a vertical stack of network layers:

- Application Layer
Web, E-Mail, File Transfer
- Transport Layer (TCP)
Reliable Connections
- Internet Layer (IP)
Simple, Unreliable
- Link Layer (Ethernet, WiFi)
Physical Connections

<http://www.zakon.org/robert/internet/timeline/> happened in the late 1980s.
The [COUGH] the number of web hosts or

00:20:45 / 00:25:13 65

w8.1- Application Layer.mp4 — Haruna Media Player



A large, dense network graph visualization representing the Internet. It consists of numerous small nodes (ranging from green to purple) connected by a complex web of thin lines, forming a complex, organic shape against a black background. In the lower-left corner of the graph area, there is a smaller, semi-transparent inset showing a zoomed-in view of a specific cluster of nodes and their connections.

bill, billions of computers.
And so, if you start looking at the

http://en.wikipedia.org/w/index.php?title=Internet_map&oldid=10241024

- Application Layer
Web, E-Mail, File Transfer
- Transport Layer (TCP)
Reliable Connections
- Internetwork Layer (IP)
Simple, Unreliable
- Link Layer (Ethernet, WiFi)
Physical Connections



A video frame showing a man with a beard and short hair, wearing a dark blue t-shirt with a yellow 'MSI' logo and '@SXSW' text. He is gesturing with his hands while speaking. The background shows an indoor setting with framed pictures on the wall.

00:21:11 / 00:25:13

65

w8.1- Application Layer.mp4 — Haruna Media Player

The Internet: An Amazing Design

- Hundreds of millions of computers
- Thousands of routers inside the Internet
- Hundreds of millions of simultaneous connections
- Trillions of bytes of data moved per second around the world
- And it works

living thing.
It's billions of computers, and hundreds



w8.1- Application Layer.mp4 — Haruna Media Player

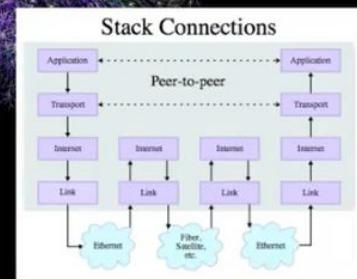
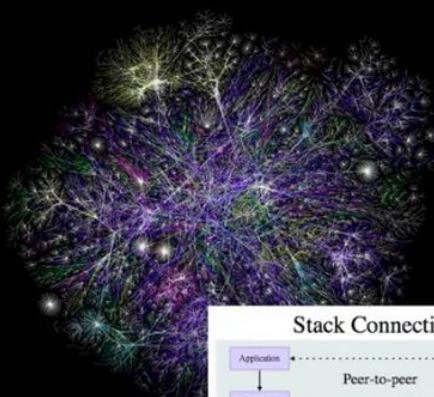
The Internet

- It is said that “The Internet **is the largest single engineering effort ever created by mankind**”
- It was created to work in an organic way - to repair itself and automatically adjust when parts fail
- No one part of the Internet knows all of the Internet (like life)
- It is never 100% up - but it seems up all the time

perfect, but it kind of works.
You can think of it as like, the largest,



w8.1- Application Layer.mp4 — Haruna Media Player



Oops, went the wrong way.

And so, we kind of come to the end of

w8.1- Application Layer.mp4 — Haruna Media Player

We are not done experimenting...

- There is still very active exploration on how network technology can be improved
- Content-Centric Networking is only one advanced idea
- Routers in the future can have *lots* of memory - lets try not to send the same piece of data more than once

The video player interface at the top shows the title "w8.1- Application Layer.mp4 — Haruna Media Player". Below the title, there is a large black area containing the slide content. To the right of the slide, there is a video frame showing a man with a mustache, wearing a blue t-shirt with a yellow 'M' logo, sitting at a desk and speaking. Below the video frame, there is a caption: "Van Jacobson PARC, a Xerox Company". At the bottom of the slide, there is a text block: "And so the last thing that I'll close with is another video from Van Jacobson." The video player has a progress bar at the bottom, showing "00:23:37 / 00:25:13" and a page number "65" in the bottom right corner.

Van Jacobson
PARC, a Xerox Company

And so the last thing that I'll close with is another video from Van Jacobson.

00:23:37 / 00:25:13 65

Summary

This transcript provides a Magnitude 1,000,000 deep dive into the **Application Layer**, the final frontier of the four-layer internet architecture. It explains how applications leverage the "magic" of the lower layers (Link, IP, and TCP) to facilitate human communication.

1. The Application Layer & TCP Services

The Application Layer sits at the top of the stack. It views the complex infrastructure below—billions of dollars in hardware and decades of engineering—as a "reliable pipe."

- **TCP's Role:** It provides a reliable, sequenced, end-to-end stream (two-way communication).
- **The "Magic":** Applications ignore the "hopping" of IP or the physics of the Link Layer, treating the connection as a direct, reliable link between a Client and a Server.

2. Identifying Services: Ports

To handle multiple applications on a **single IP address (computer)**, the internet uses **Ports**.

Think of an IP address as a building's phone number and a Port as a specific **extension**.

- **Key Ports mentioned:**

| Service | Port | Simplified Chinese (简体中文) |

| :--- | :--- | :--- |

| Email (SMTP) | 25 | 电子邮件 |

| Remote Login (SSH/Telnet) | 22 / 23 | 远程登录 |

| Web (HTTP) | 80 | 网页 |

| Secure Web (HTTPS) | 443 | 安全网页 |

| Mailbox (POP3) | 110 | 个人信箱 |

3. The HTTP Request-Response Cycle

The World Wide Web uses the **HTTP** protocol (Hypertext Transfer Protocol). It follows a simple, elegant cycle:

1. **Click:** The user interacts with the client (browser).
2. **Request:** The client sends a `GET` command for a specific document.
3. **Response:** The server sends the document (HTML) back.
4. **Display:** The browser renders the HTML and drops the connection.

4. "Hacking" the Protocol

The speaker demonstrates how to simulate a browser using a **Telnet** client. By connecting to a server on Port 80 and typing raw **GET** commands (following the RFC 1945 specification), you can manually retrieve webpage data. This highlights that protocols are simply a set of **rules for conversation**.

Note on Nmap: The transcript references the "Trinity hacking scene" from *The Matrix*, noting that real-world tools like **Nmap** (Network Map) are used to probe these ports for vulnerabilities.

5. Evolution and Organic Resilience

The internet architecture has scaled from **6 hosts in 1969** to **over 1 billion in 2011** using the same core logic.

- **Resilience:** The internet is designed to "heal" and route around failures (e.g., Hurricane Sandy) rather than being a "perfect," fragile system.
- **Future Outlook:** While the current model is robust, innovators like Van Jacobson suggest a shift toward **Content-Centric Networking**, reminding us that engineering is never static.

w8.2- Van Jacobson - Content Centered Networking

w8.2- Van Jacobson - Content Centered Networking.mp4 — Haruna Media Player

Subtitle scale: 0.4

Van Jacobson

PARC, a Xerox Company today.

Trying to join together the very

00:00:20 / 00:10:00

45

Summary

1. The Legacy "Telephone" Model

The Internet was originally designed based on a 140-year-old telephony mindset.

- **The Conversational Model:** TCP/IP treats communication as a private conversation between two specific endpoints (hosts).
- **The "Long Wire":** Like the phone system, it focuses on building a virtual connection from point A to point B.
- **The Problem:** This worked when computers were just exchanging data, but it fails when millions of people want the exact same content simultaneously.

2. The Information-Centric Shift

In the late 90s and 2000s, the Web changed everything. Users stopped caring *which* computer they were talking to and started caring only about the **Content**.

- **Naming Information:** We now **use URLs (YouTube videos, Amazon pages)** to name the data itself, not the process of getting it.
- **The "Overlay" Hack:** Today, the Web is just an **"overlay" on top of TCP/IP.** We spend all our energy "lying" to the network (using load balancers and complex routing) to make a single IP address appear to be in thousands of locations at once.

3. The "Slashdot Effect" & Abysmal Scaling

Because TCP/IP is conversational, it cannot inherently handle popularity.

- **Inefficient Distribution:** If 1,000 people in the same building watch the same YouTube video, the network currently sends 1,000 separate copies of those bits from the source to the destination.
- **Memory Waste:** Every router (gateway) has memory for buffering, but because it sees "conversations" instead of "content," it can't say, "I already have this video in my memory; I'll just give it to the next person who asks."

4. The Future: Content-Centric Networking (CCN)

Jacobson proposes recasting the low-level infrastructure to match how we actually use it.

- **Named Packets:** Instead of source/destination IP addresses, packets should have a **Name** on the front.
- **Center-Structured Information:** Remove the rigid layers (Layer 3, Layer 4, etc.). A router looks at only as much of the "Name" as it needs to route the packet.
- **The Network as a Source:** If data is named, any router's memory becomes a viable source. You start moving toward the data's origin, and the moment you encounter a router that already has a copy, you're done. The load no longer scales with popularity.

Content centric networking- Explanation

The Efficiency of "Named" Data

In the current system, if you want a file, you must connect to a specific location (a server) to get it. In the **Content-Centric Networking (CCN)** model, you simply ask for the data by **Name** (e.g., "YouTube-Video-X").

- **Router Memory as a Store:** Every router in the network has memory used for buffering. In CCN, if a router has recently passed "YouTube-Video-X" to another user, it keeps a copy of that data in its memory.
- **Proximity over Origin:** When you request that video, your request starts traveling toward the original server. However, the moment your request hits *any* router that already has a copy in its memory, that router gives you the data immediately.
- **Abysmal Scaling vs. Efficiency:** Currently, if 1,000 people want the same video, the server sends it 1,000 times. In CCN, the server sends it **once** to the first person; the next 999 people simply get it from the nearest router that saved it.

Why "Popularity" No Longer Saturated the Links

In the traditional model, traffic scales with popularity: the more people want a video, the more the server's link becomes saturated. In the CCN model, **the more popular a piece of data is, the easier it is to find**, because it will be stored in the memory of almost every router near the users.

