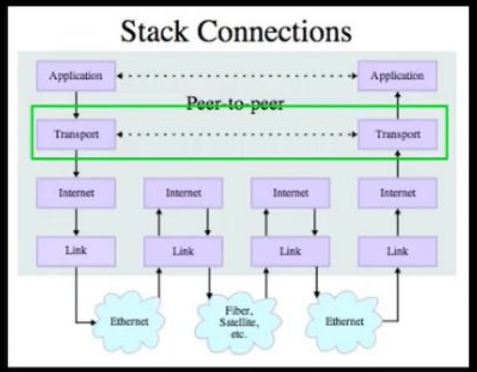
w8.1- Application Layer

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

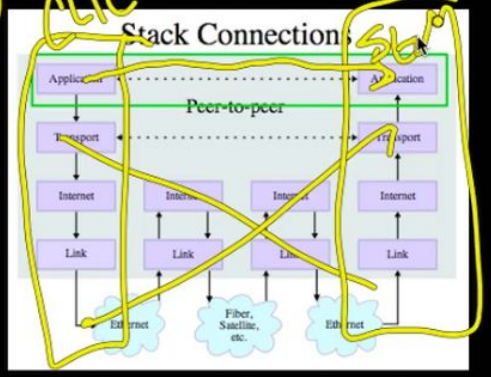


Source: http://en.wikipedia.org/wiki/Internet_Protocol_Suite

basically to give us a reliable, sequenced end to end stream, that can

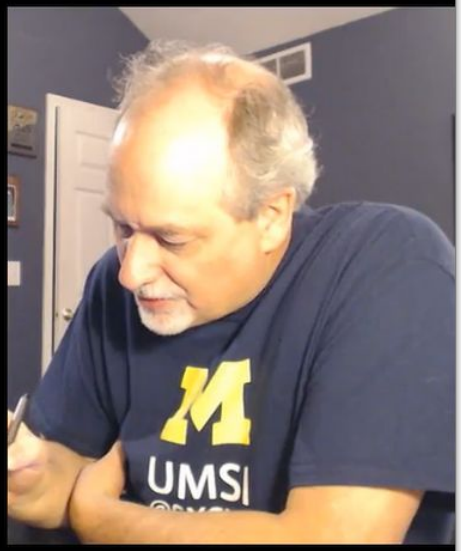
Application Protocol

- Since TCP gives us a reliable pipe, what to 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.



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

So, the, there are two basic questions that the application layer has to solve.

Ports

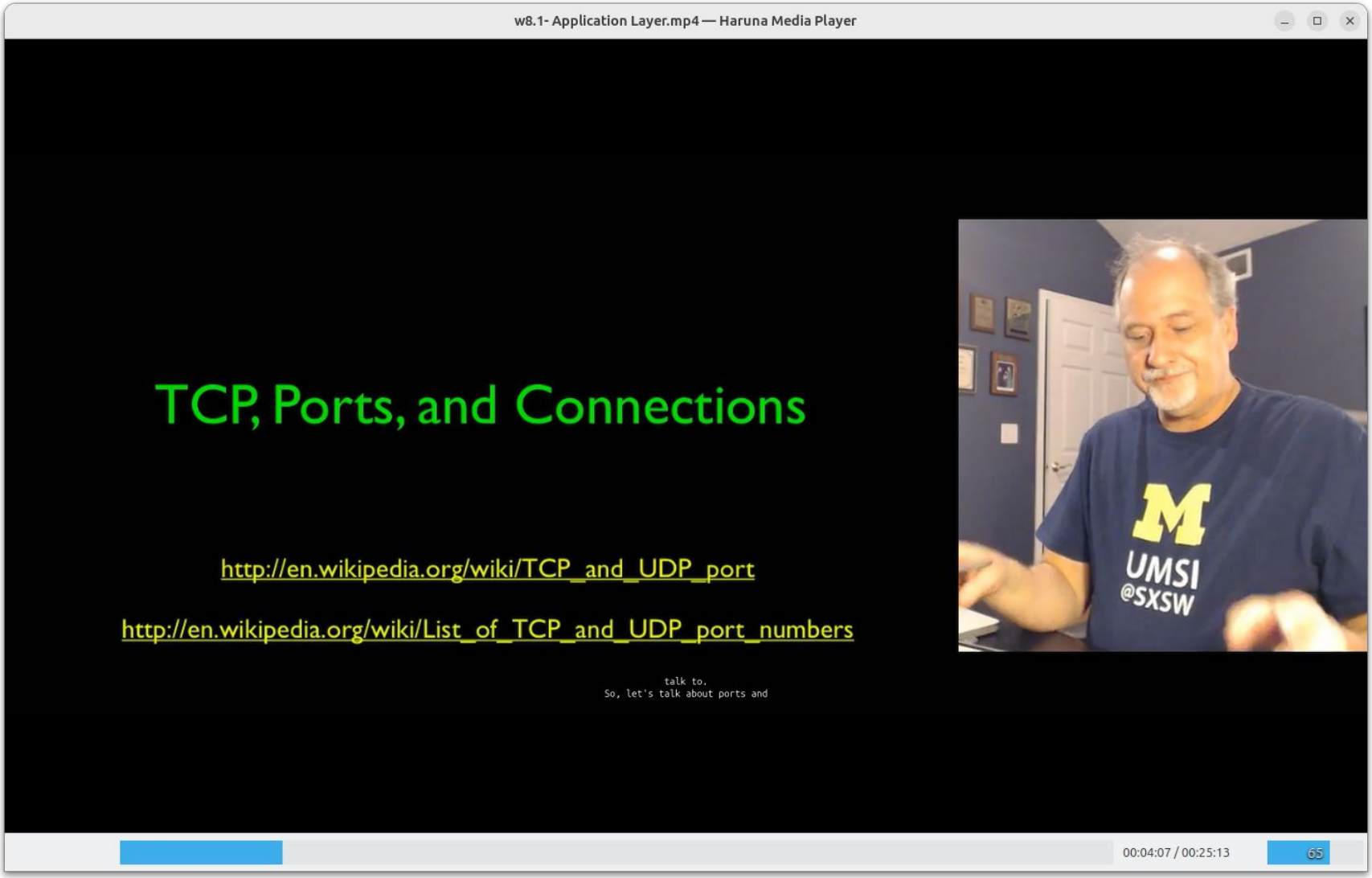
- Like extensions in a phone number
- The IP address network number (the area code) gets to the LAN
- The IP address host number (the telephone number) gets you to the destination machine
- The port (the extension) gets you to a specific application

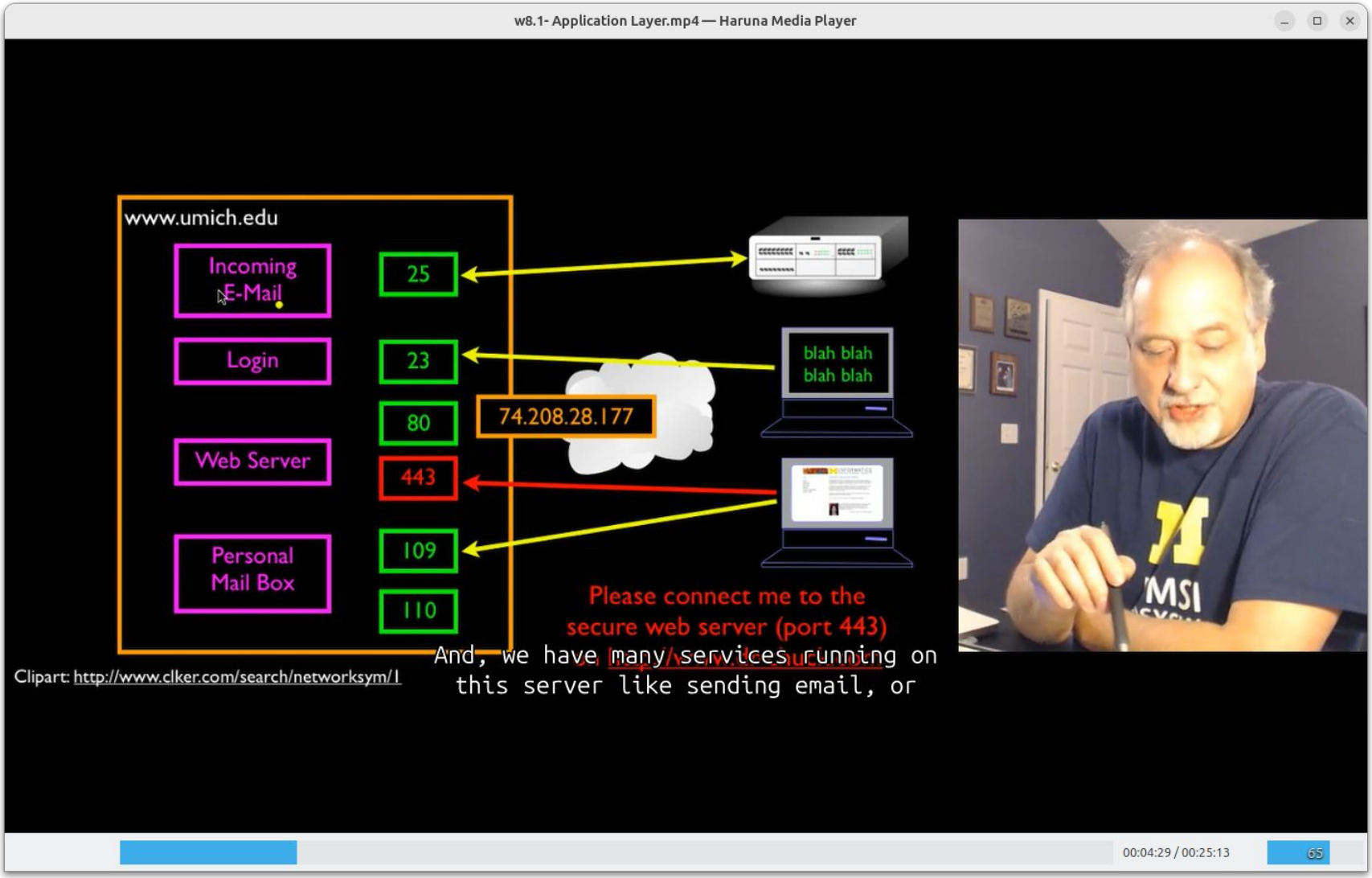
(734) 764 1855, ext. 27

141.211.144.188

Port 25

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





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/List_of_TCP_and_UDP_port_numbers

So some common TCP ports that we have are the ones I just mentioned.

Application Protocols

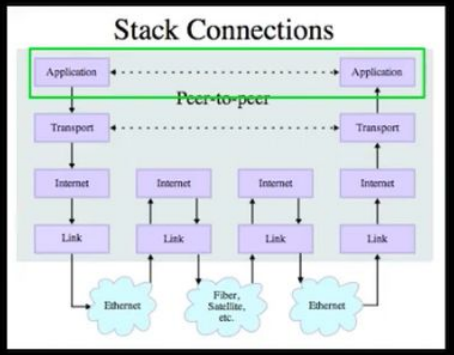
http://en.wikipedia.org/wiki/Http

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

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

Application Protocol

- Since TCP gives us a reliable pipe, what to 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

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

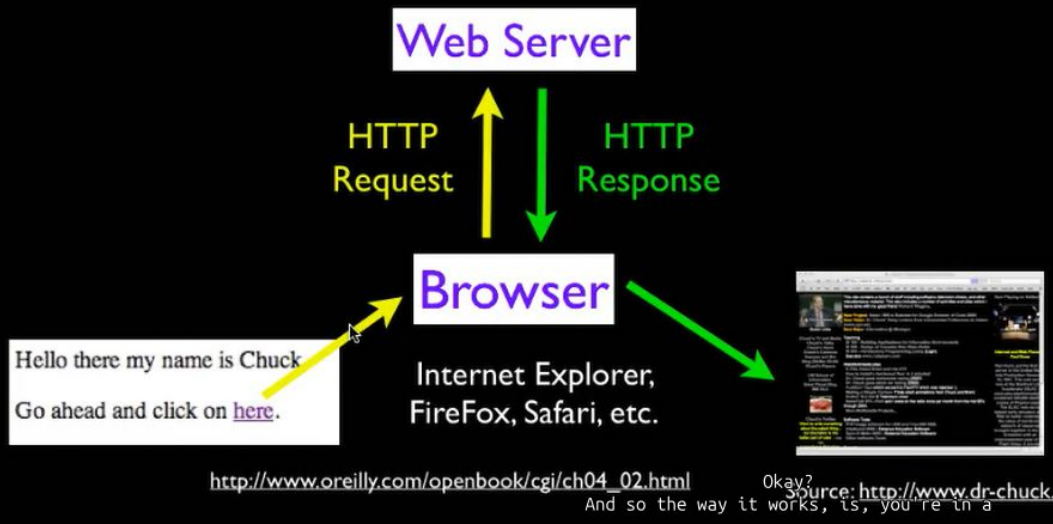
http://en.wikipedia.org/wiki/Http

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

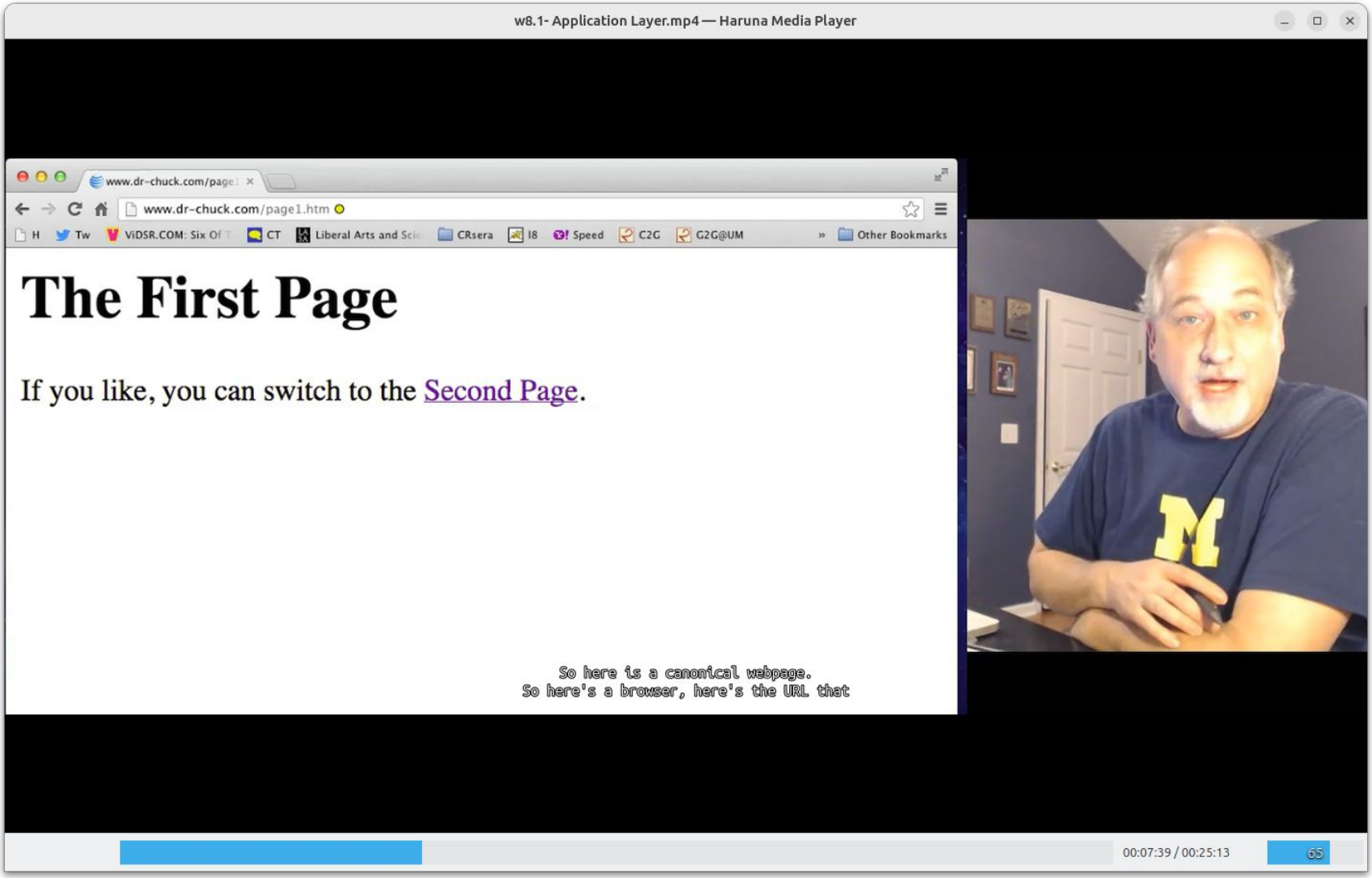
HTTP Request / Response Cycle

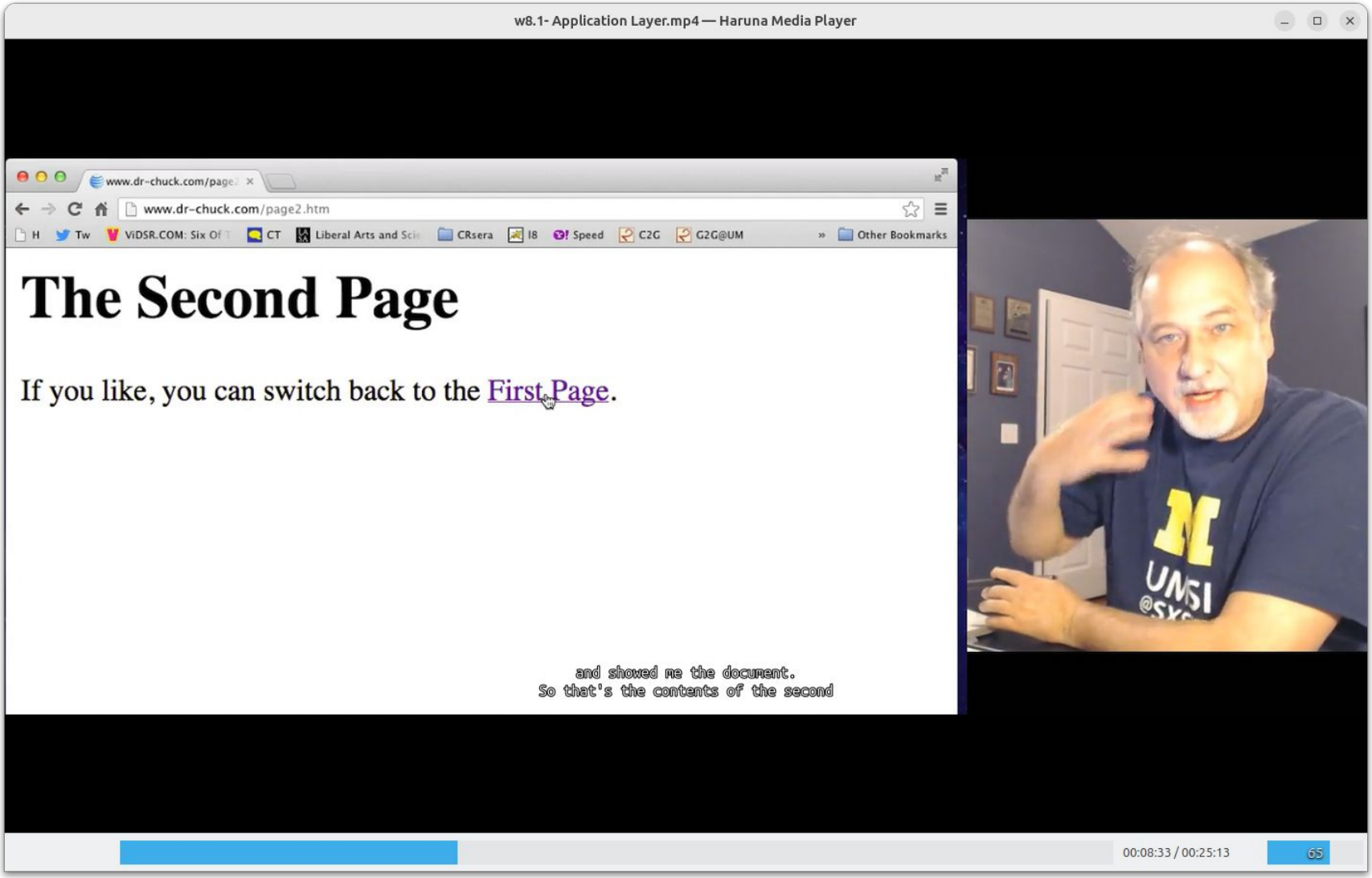


Source: http://www.dr-chuck.com/

Okay?
And so the way it works, is, you're in a

00:07:00 / 00:25:13





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

HTTP Request / Response Cycle

Web Server

GET /page2.html

HTTP Request

HTTP Response

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

Browser

Hello there my name is Chuck

Go ahead and click on here.

Internet Explorer, FireFox, Safari, etc.

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

Remember that this part here's the internet right there.

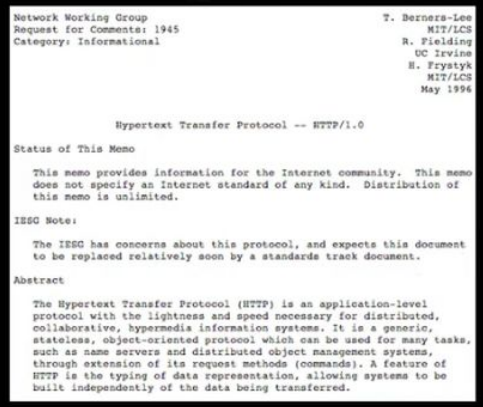
Source: http://www.dr-chuck.com/

00:09:16 / 00:25:13

65

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
MIT/LCS
R. Fielding
UC Irvine
H. Frystyk
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, object-oriented protocol which can be used for many tasks, 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.

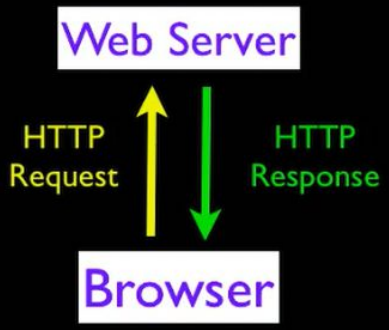
Source: http://www.ietf.org/rfc/rfc1945.txt

Well, we would go back to the IETF, the Internet Engineering Task Force, and we

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

"Hacking" HTTP

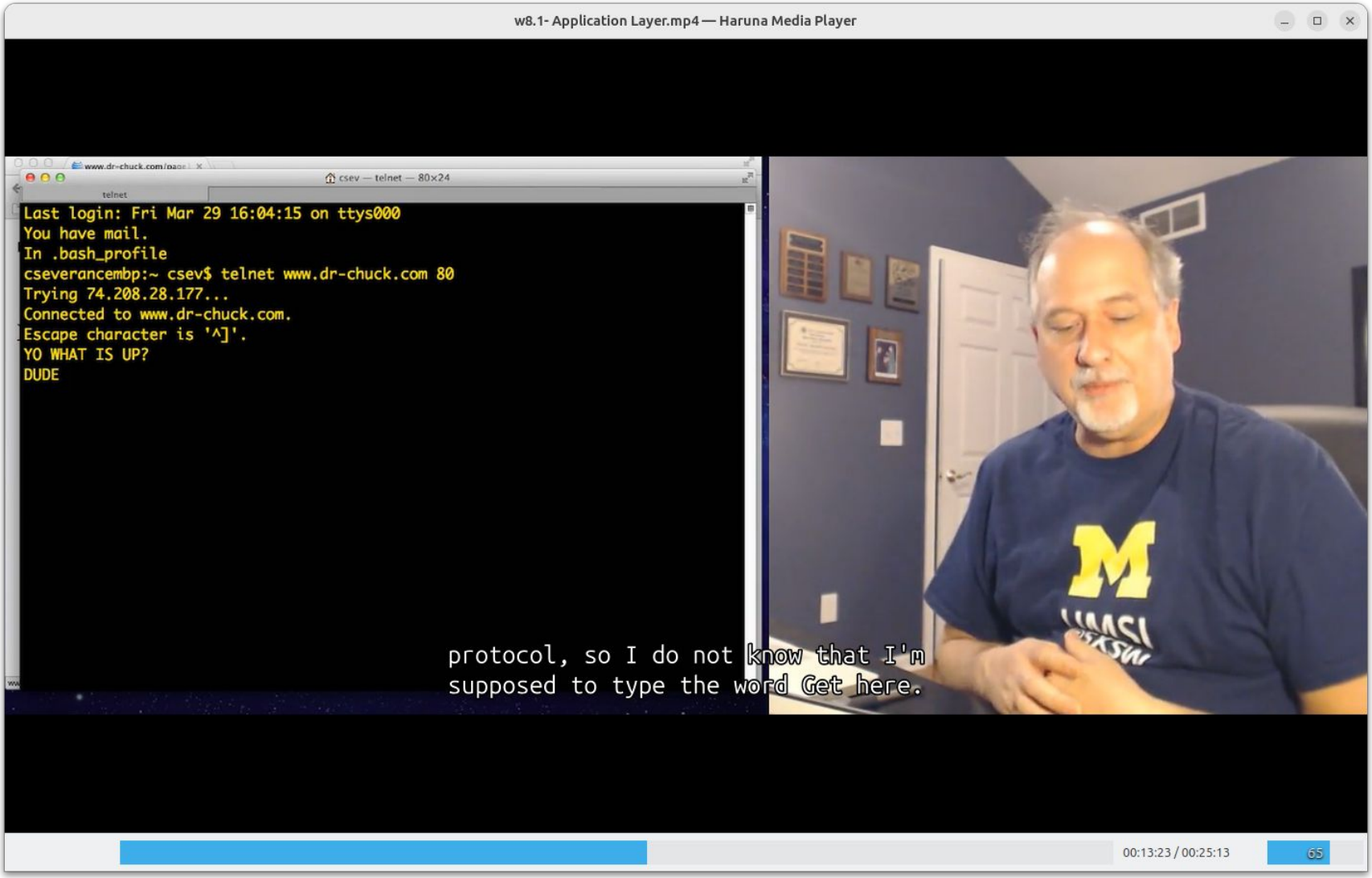
```
Last login: Wed Oct 10 04:20:19 on ttyp2
si-csev-mbp:~ csev$ telnet www.dr-chuck.com 80
Trying 74.208.28.177...
Connected to www.dr-chuck.com.
Escape character is '^]'.
GET http://www.dr-chuck.com/page1.htm
<h1>The First Page</h1>
<p>
If you like, you can switch to the
<a href="http://www.dr-chuck.com/page2.htm">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

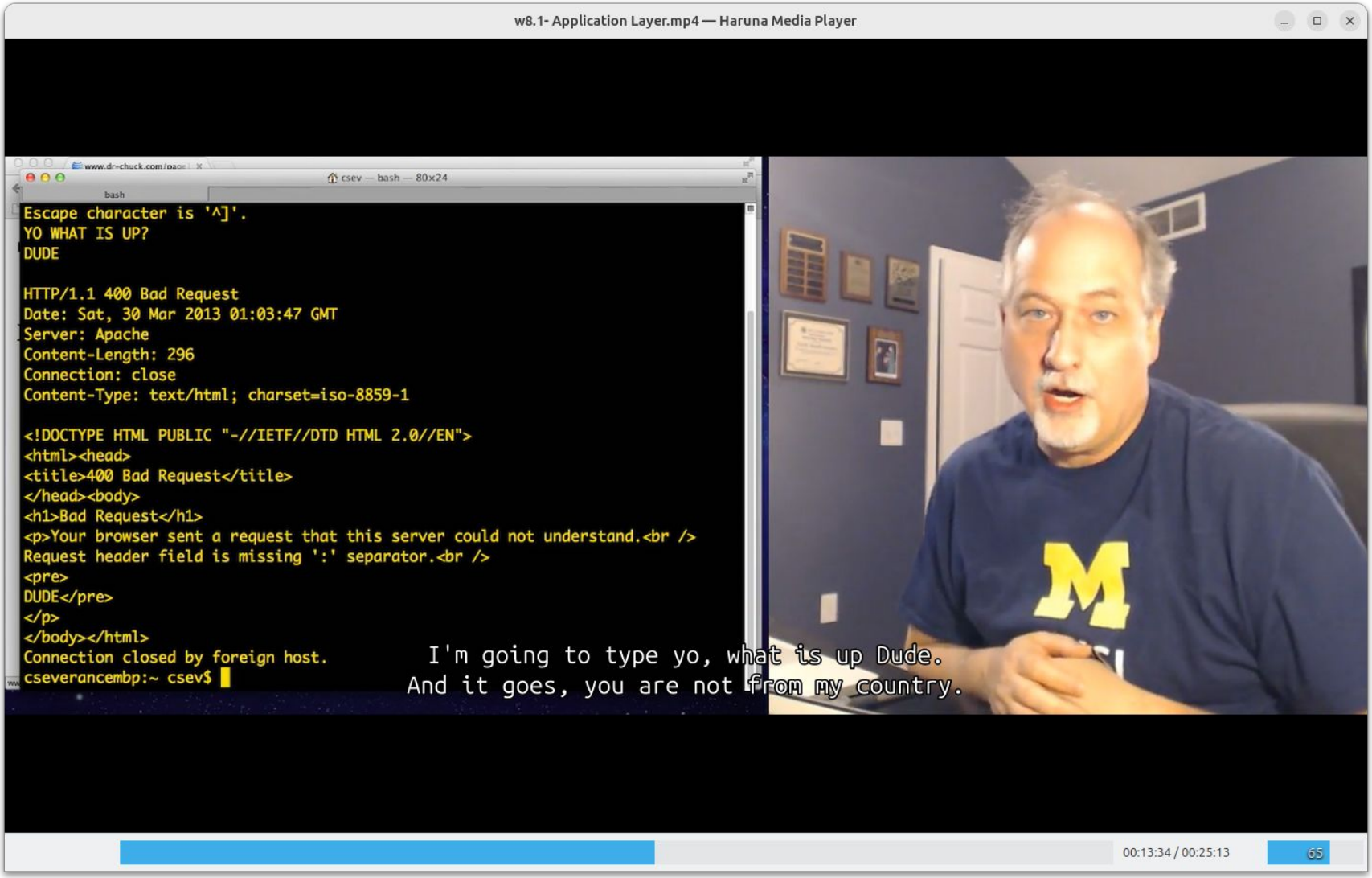


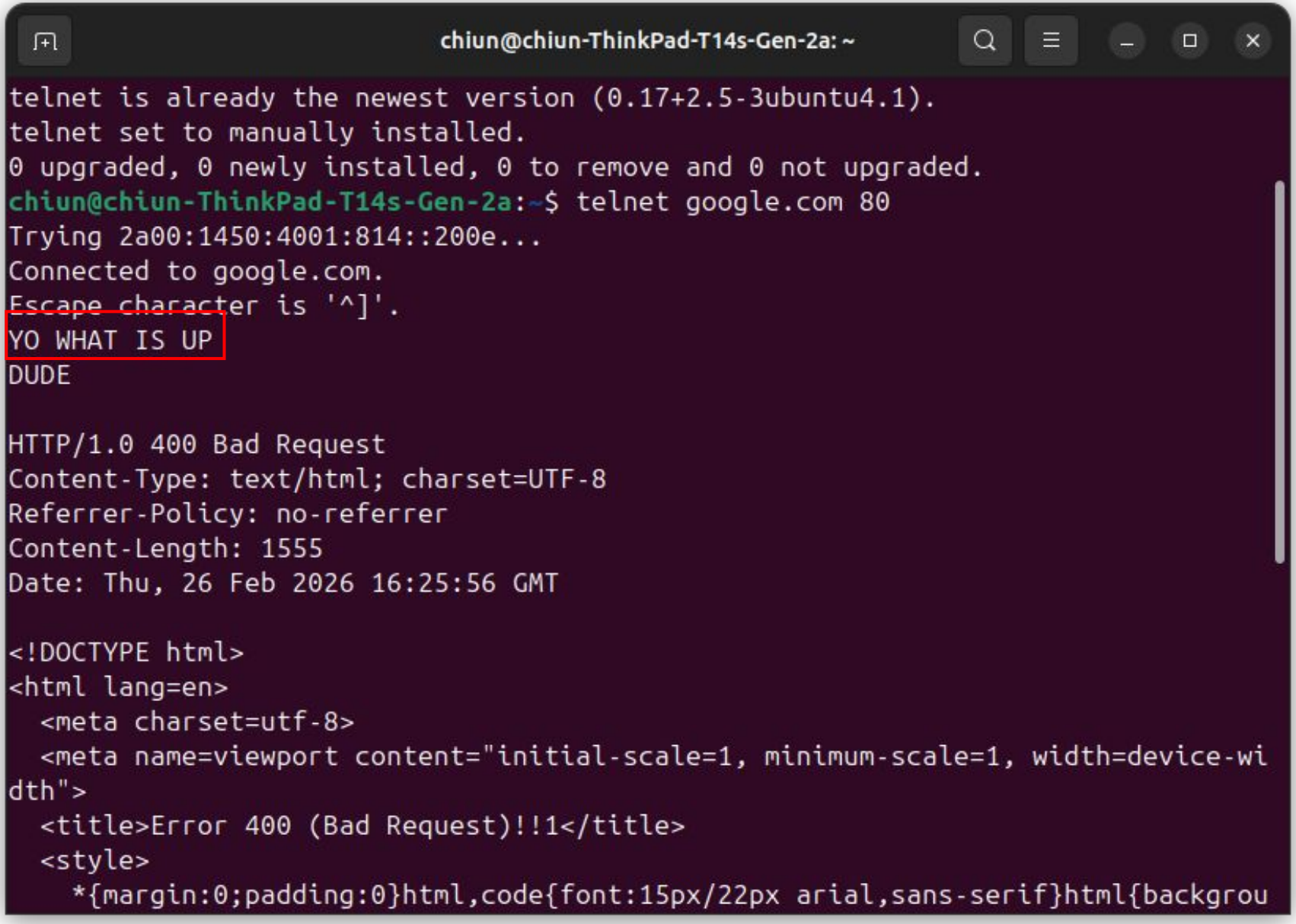
But what we're going to do, is we're going to fake being a web browser, we're

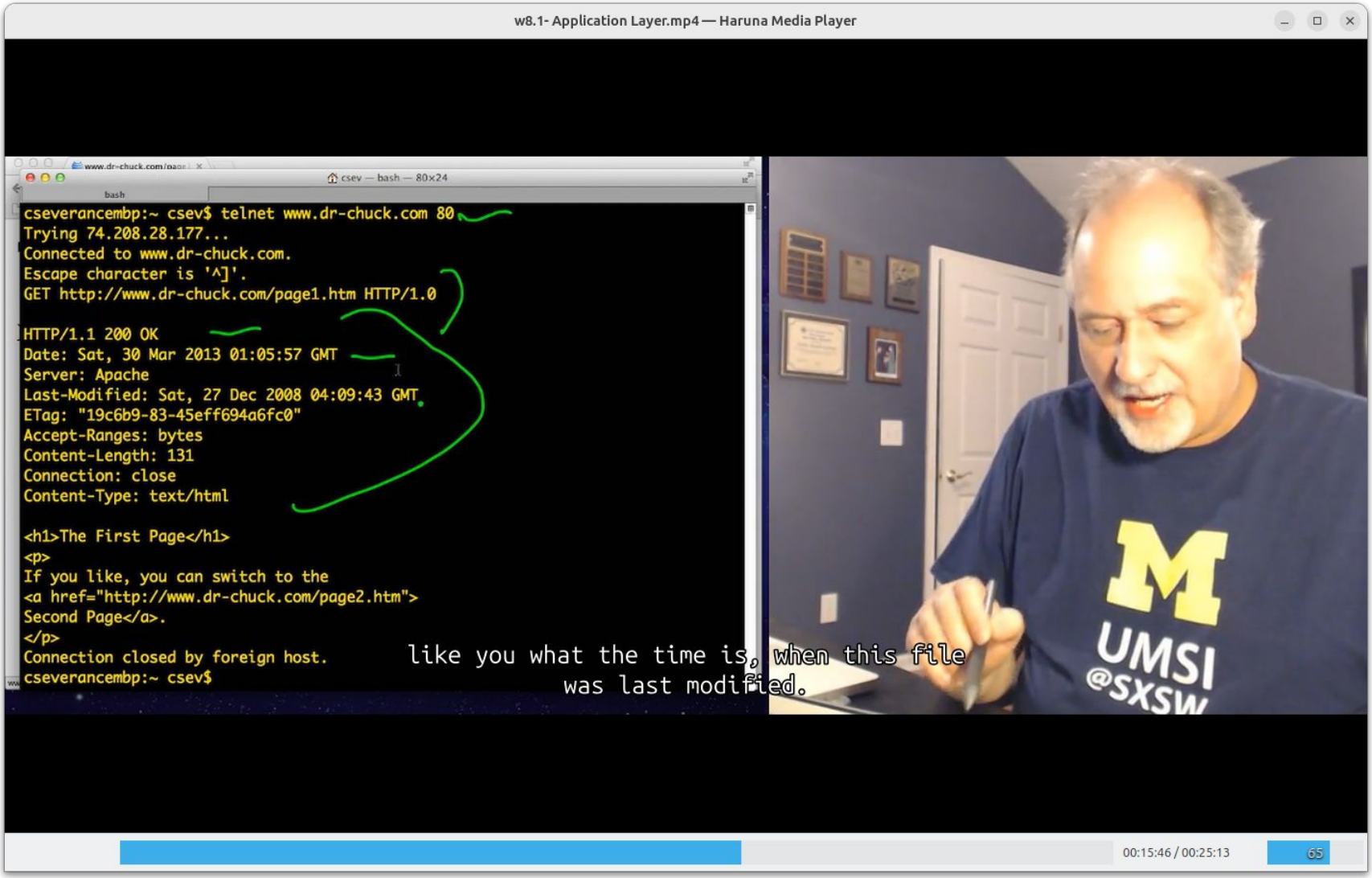
Port 80 is the non-encrypted HTTP port.

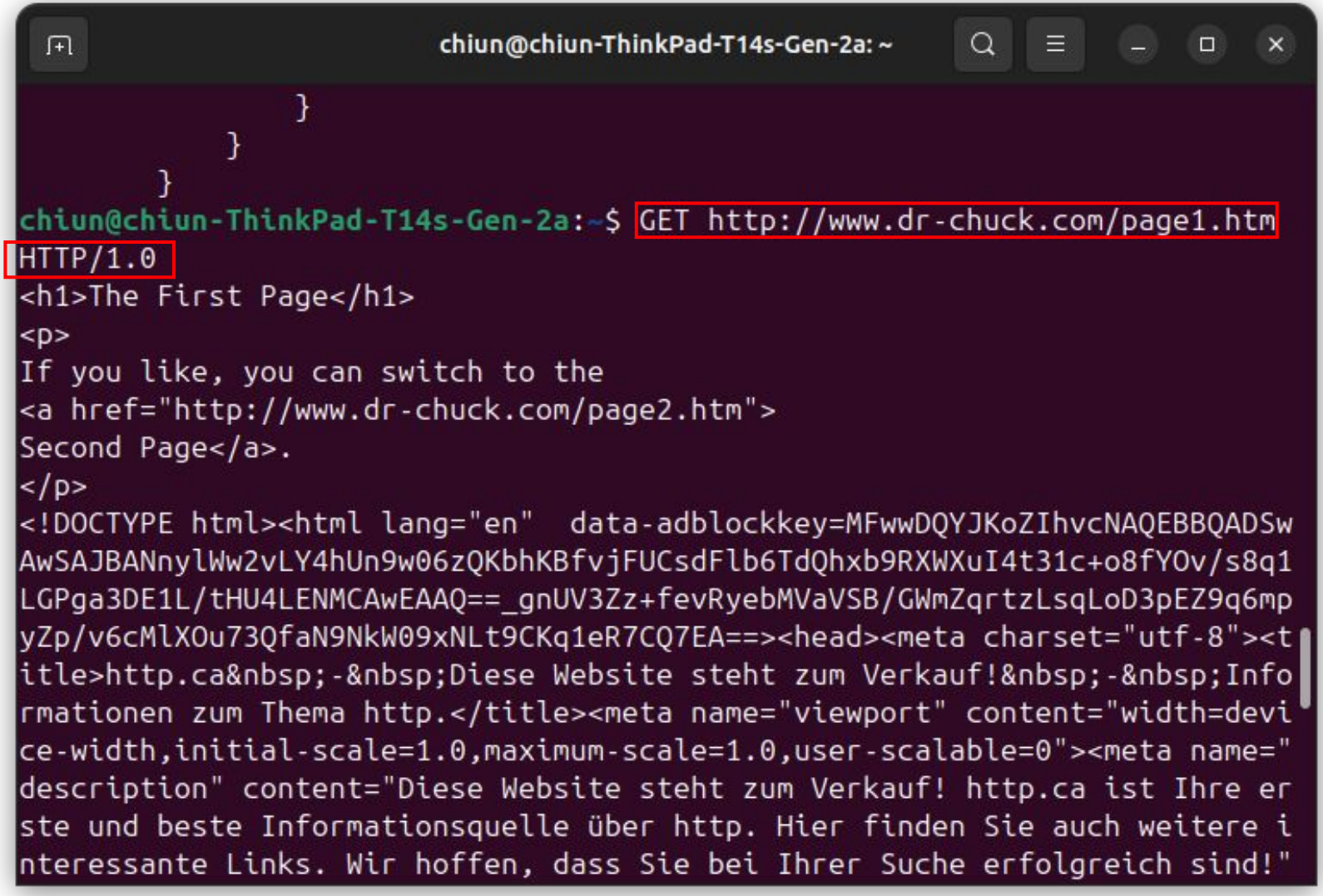
00:10:52 / 00:25:13

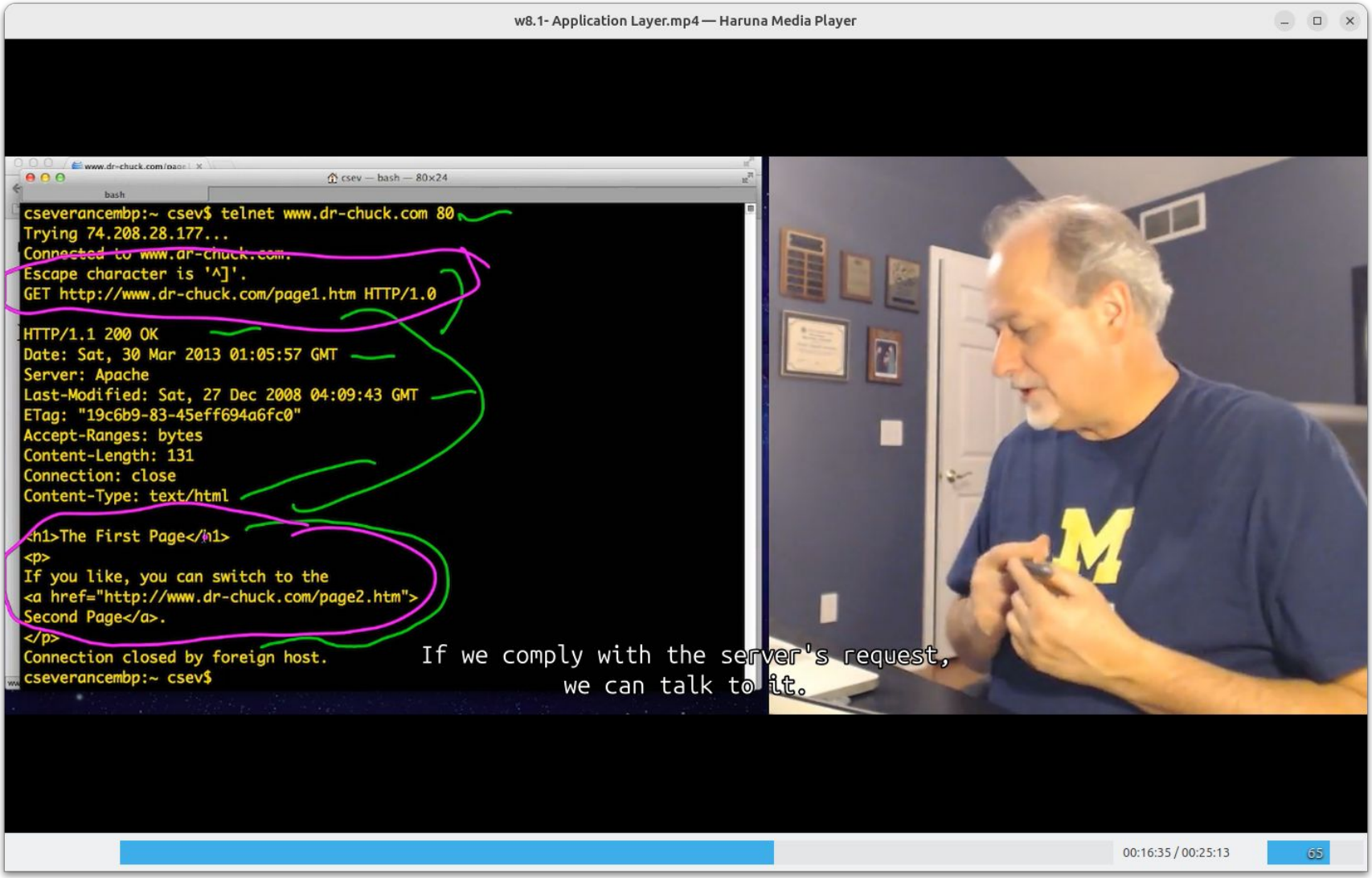








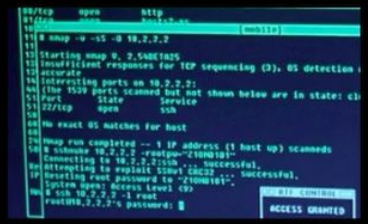




w8.1- Application Layer.mp4 — Haruna Media Player

Accurate Hacking in the Movies

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

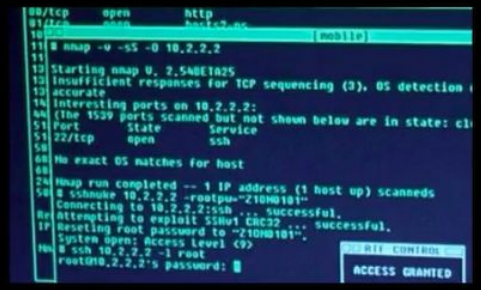


http://nmap.org/movies.html (scroll down for video)
Or search YouTube for "Trinity hacking scene"

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

00:17:58 / 00:25:13

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

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

Application Layer Summary

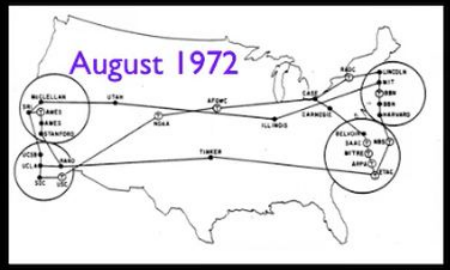
- We start with a "pipe" abstraction - we can send and receive data on the same "socket"
- We can optionally add a security layer to TCP using SSL - Secure Socket Layer (aka TLS - Transport Layer Security)
- We use well known "port numbers" so that applications can find a particular application *within*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

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

00:19:19 / 00:25:13

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

wireless, all of the wonderful things
that happen there.

00:19:54 / 00:25:13

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

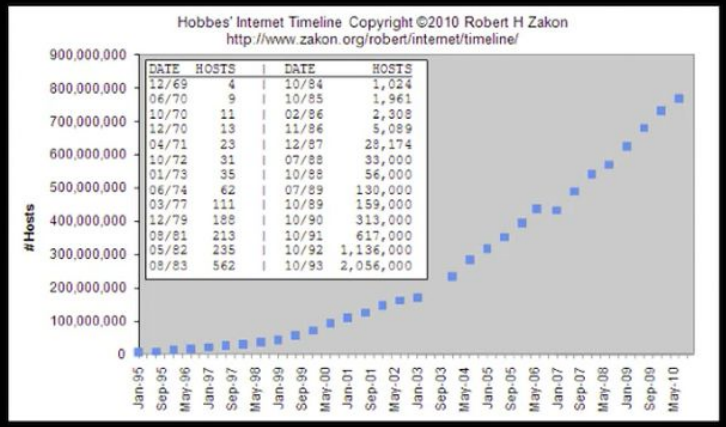
The Architecture of the Internet

- 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

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



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

Application Layer
Web, E-Mail, File Transfer

Transport Layer (TCP)
Reliable Connections

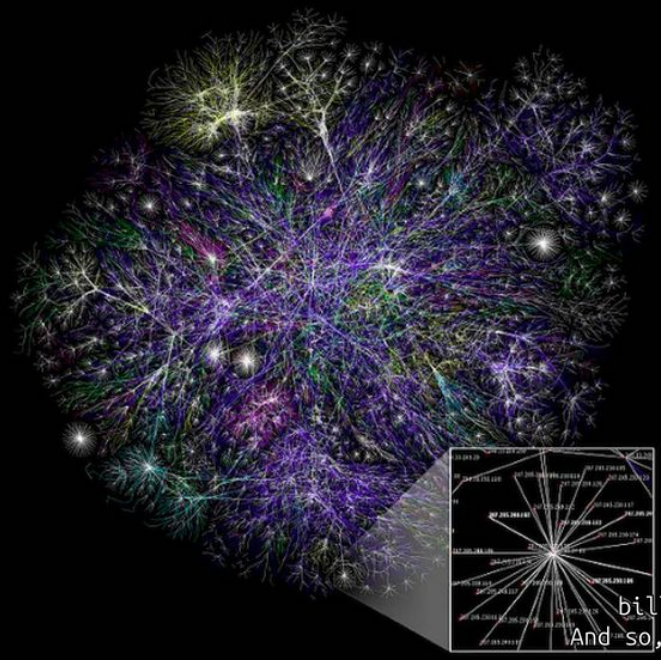
Internetwork Layer (IP)
Simple, Unreliable

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

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



Application Layer
Web, E-Mail, File Transfer

Transport Layer (TCP)
Reliable Connections

Internetwork Layer (IP)
Simple, Unreliable

Link Layer (Ethernet, WiFi)
Physical Connections

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

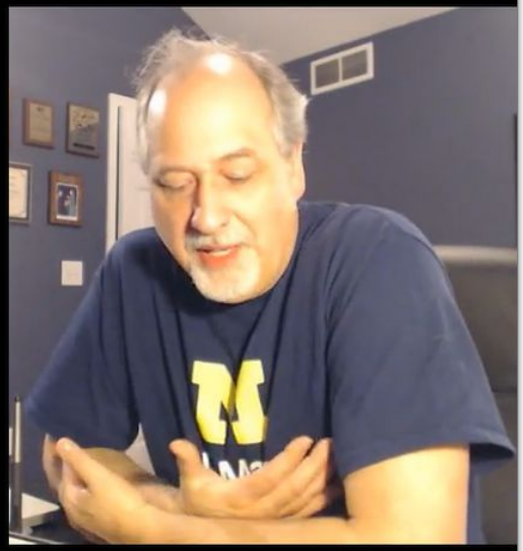
http://en.wikipedia.org/wiki/File:Internet_map_1024.jpg

00:21:11 / 00:25:13

65

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



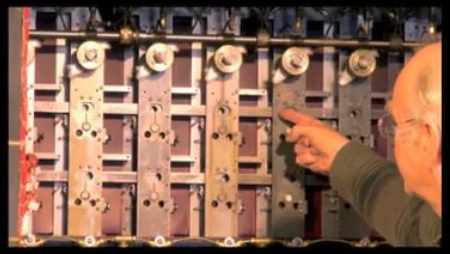
living thing.
It's billions of computers, and hundreds

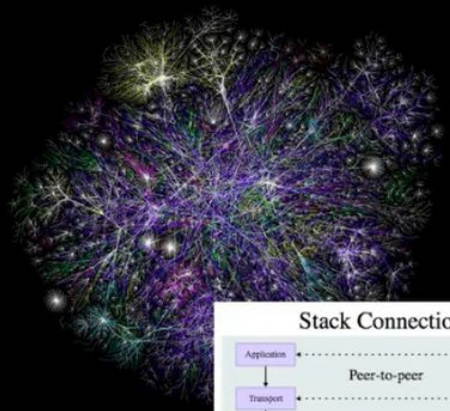
The Internet

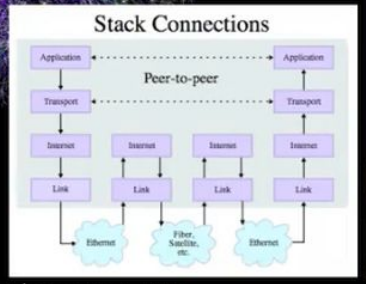
- It is said that "The Internet is the largest single engineering effort ever created by mankind"
- It was created to work in an organic way - to repair itself and automatically adjust when parts fail
- No one part of the Internet knows all of the Internet (like life)
- It is never 100% up - but it seems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

00:23:09 / 00:25:13

65

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

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

00:23:37 / 00:25:13

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

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.