University of Illinois at Urbana-Champaign
Dept. of Electrical and Computer Engineering

ECE 101: Exploring Digital Information Technologies for Non-Engineers

Distribution and Streaming

Internet Just for Point-to-Point Communication?

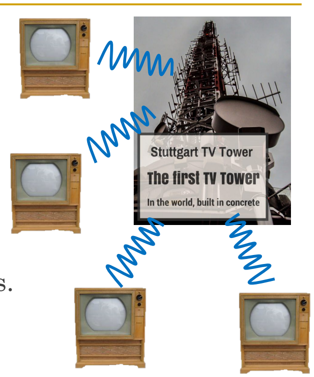
So far, we have focused on the Internet
- as a way for one computer
- to communicate with another.

That was the original design,
and how the Internet is most often used.

What about Broadcast Media?

For **most of the 20th century**,
- **part of** the electromagnetic **spectrum**
- was **dedicated to television signals**
- **broadcasting** to most of the world.

In the **later** part of the century,
- **copper cables** were **used**
- to deliver signals to areas
- that were hard to reach with EM signals.
- Again, **purely for broadcast**.

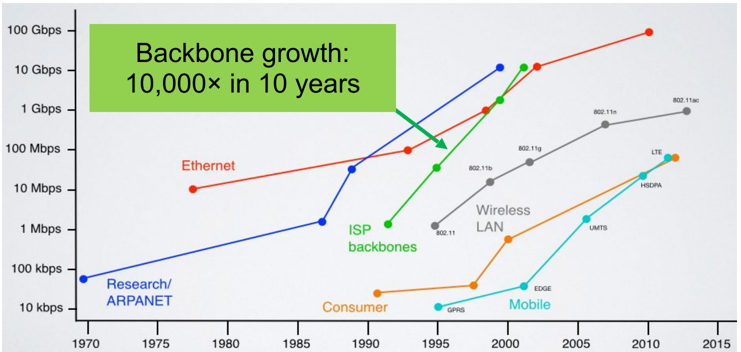


Can the Internet Replace Other Networks?

By the early **90s**,
- **link bandwidths** on university campuses
- were **large enough**
- **to transmit video** in real-time.

As bandwidth from the Internet
- continued to grow **in** the 90s and **2000s**,
- **other media**, such as telephony and broadcast media (television and cable)
- **started to migrate** their infrastructures **onto** the **Internet**.

Backbone Bandwidth Grew Rapidly Starting in 1990



Today, Internet Supports Video on Demand for All

Today's Internet is used **routinely** to **deliver** high-definition **video on demand to hundreds of millions** of people.

Next, we'll talk about how such feats are possible.

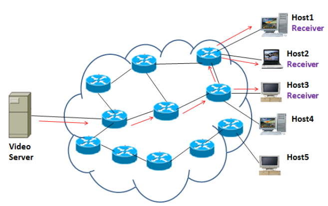
Part of the solution was
- **technological advances in communication** (last slide),
- which enabled orders of magnitude more information to traverse the networks.

Advances in Distributing Content were Also Important

But another part of these accomplishments
- is **understanding how to deliver**
- the **data** people want
- **in a** more **scalable way**.

That's where we'll focus our attention.

Let's start with **multicast**:
- **one sender**,
- **several** or many **receivers**.



Multicast is Not an Easy Problem

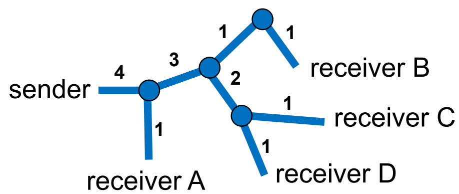
Trying to find a good way to support **multicast** in the Internet **occupied researchers for decades**.

Everyone knew it was important, but
- with millions of connections through each router,
- solutions could not expect routers
- to handle each connection separately.

Sending the Same File to Many Receivers is Slow

Imagine a sender transmitting a file to four receivers.

How many copies cross each link?

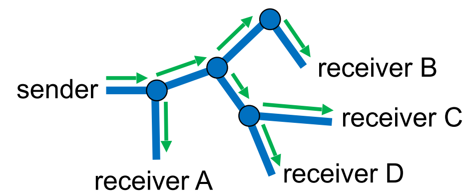


Sender uses at most ¼ of the first link's data rate for each copy.

What if there are more receivers?

Multicast as a Possible Solution

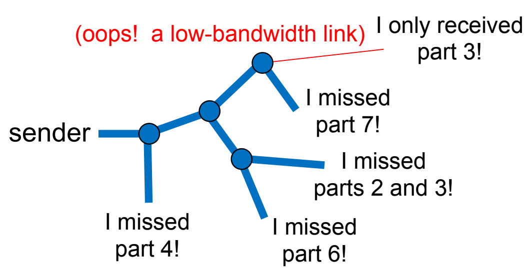
Instead, sender could **multicast** one copy to all.
- **Routers must forward onto multiple links.**
- Same data on all links.



More receivers, more links (but no more data on any link).

Multicast vs. Retransmission and Variable Rates

Multicast has issues with unreliable delivery.

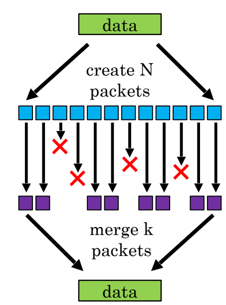


Should a sender slow down to the bandwidth of the slowest listener?

Coding Theory Can Help

One solution to such problems
- uses **coding theory**,
- adding carefully calculated extra bits
- to make obtaining the original bits easier.

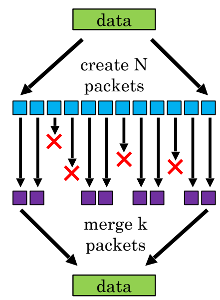
Specifically,
- given some data bits,
- we can **create N packets** such that
- **any k (< N)** of the **packets suffice**
- **to recover** the original **data** bits.



Codes Can be Tuned to an Application's Needs

These **k-out-of-N codes**
- are fairly general and powerful.
- We **can pick any value of k**,
- but smaller **k** means larger packets.

For example,
- **k=1** means each packet
- holds all of the original data,
- which is not so useful.

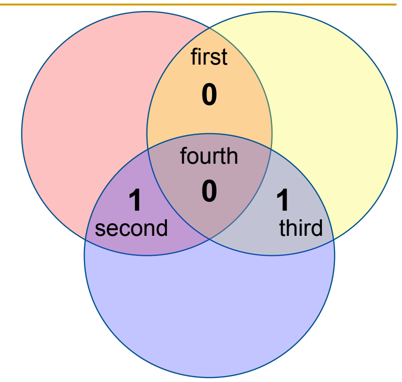


Write Any Four Bits into the Intersections

Let's try a 5-out-of-7 code.

Pick any four bits.
- **Write** your bits
- **into** the **intersection areas**
- in the order shown.

For example,
- say you want to send **0110**.
- Let's fill in the four bits…



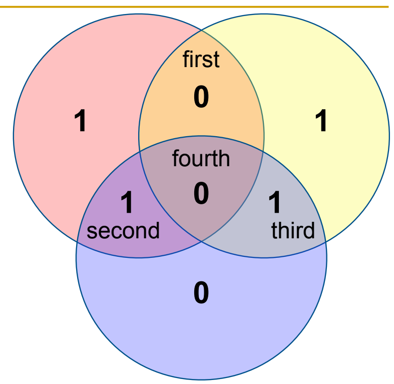
Make Each Circle Contain an Even Number of 1s

Now, **for each circle**
- **write one** more **bit**
- **so** the **circle contains**
- an **even number of 1 bits**.

In the **red** circle …

And the yellow circle…

And the **blue** circle…



Now Imagine that We Lose Two of the Bits

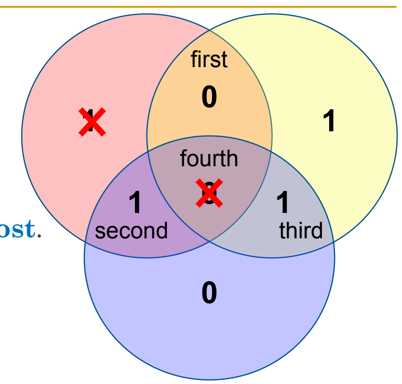
Now let's pretend that
- we **send** these bits
- **through** the **Internet**,
- each in a separate packet.

The **Internet is unreliable**.

Let's say **two** of the packets **are lost**.

Any two.

But for now, I'll pick two.



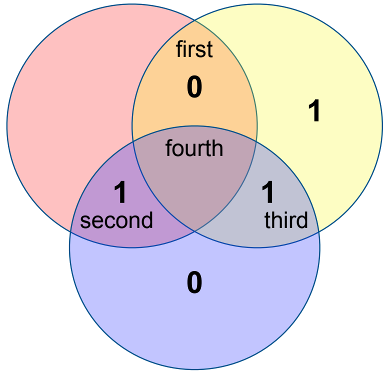
Can We Identify the Missing Bits?

What can we do?

What if we start
with the **red** circle?

It **should have** an
even number of 1s.

So … can't start there.



We Can Always Find Both Missing Bits

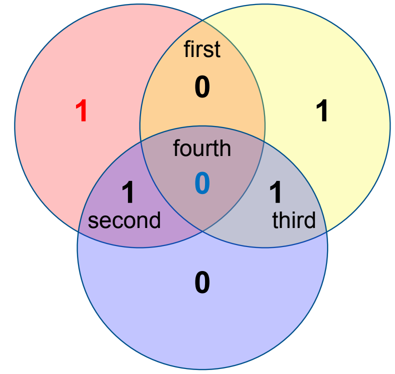
Let's try the **blue** circle.

Again, it **should have** an
even number of 1s.

What is the missing bit?

Now we can finish the **red** circle.

What is the missing bit?



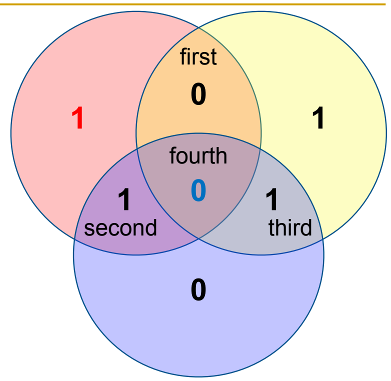
Can We Identify the Missing Bits?

Once we know all of the bits,

we can read off the original four:

0 … 1 … 1 … 0

In other words,
- as soon as we **receive**
- any **5 of the 7 bits,**
- we can **recover** the **original 4!**

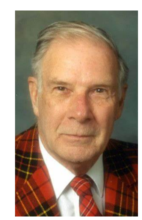


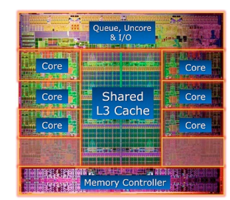
Hamming Codes Used in Nearly Every Digital Memory

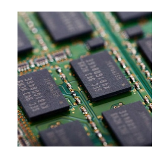
The code we just used
- is called a **Hamming code**
- after Richard Hamming,
- a UIUC Math alumnus.

Other than to illustrate the idea,
it's not so useful as a 5-out-of-7
code for multicast.

But it is **used in nearly every memory**!

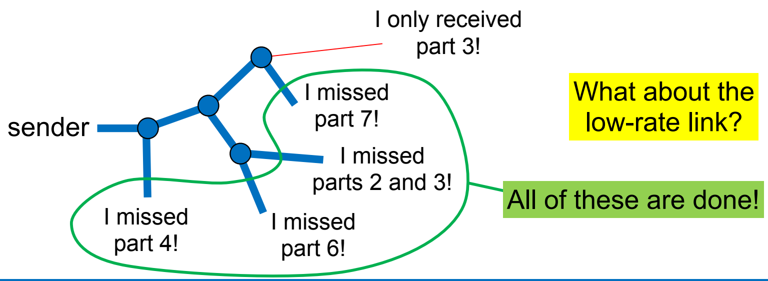






Multicast vs. Retransmission and Variable Rates

Let's **imagine** the **same multicast** as before **with** a **5-out-of-7 code.**



Repeated Multicast Eventually Reaches All Receivers

If sender
- **transmits** the 7 parts
- **over and over**
- all **receivers eventually receive 5** parts.

Having a wide variety
- of **receive rates**
- **makes no difference**!

(This technology was commercialized by Digital Fountain and eventually purchased by Qualcomm.)

Multicast Does Not Help with Video on Demand

What about on-demand video?

Multicast is only useful for
- **fully synchronous**
  - all receivers watch one video simultaneously
  - (like Zoom, Google Meet, Microsoft Teams, and other conferencing tools—we'll come back to those along with gaming in a few weeks), **and**
- **fully asynchronous**
  - all listeners receive the same data,
  - but don't display the data in real-time.

Video on Demand Leverages Growth in Bandwidth

Video on demand allows each user
- to watch a video in **real-time**
- **independently of other** users.

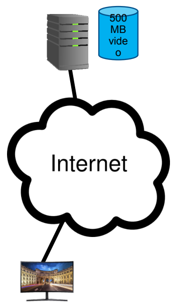
This type of service is what we expect
- from YouTube, Netflix, and Hulu, and
- or from the video advertisements injected into our viewing.

To some extent, these applications **rely on** advances in **network bandwidth**.

Locality is Also Important for Content Distribution

Locality also **plays an important role**:

- TCP's achieved bandwidth drops with increased delay, so
- moving content closer to users improves their experience, and
- not sharing heavily-loaded backbone links improves end-to-end bandwidth for everyone.



Distribution Reduces Latency and Increases Bandwidth

For example, Google has **datacenters** spread **around the world**.

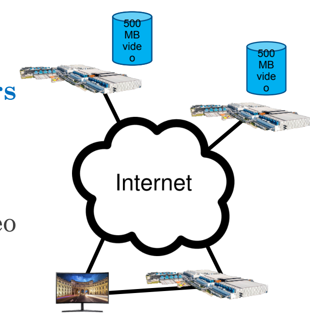
Each datacenter **has access to all** of the company's videos (or other **content**).

Anything you receive **from Google comes from** a **nearby** datacenter.

Content Replicated for Availability and Locality

Actually, for any given video,

- **copies** are kept **at several datacenters**
- **in case of catastrophe**, **and**
- any **datacenter**
  - serving a population **that**
  - **frequently requests** a particular video
- **keeps a copy cached**.



Datacenters are Connected with High-Bandwidth

How are these datacenters connected?

The answer varies:

- Google **built its own** optical network;
- other companies **rent fibers** (multiple Tbps) **or wavelengths** (10-100 Gbps), or
- P**ay another company** (such as Akamai) to handle content distribution.

Content Distribution Requires More than Locality

Distributing content amongst
a company's **datacenters**
- **can leverage multicast**
- over the company's network.

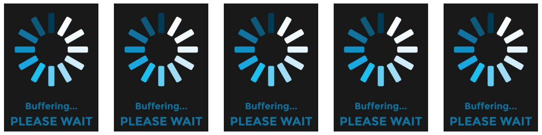
But more **tuning** is **necessary**
- to adapt **to** the wide range
- of **user needs and connectivity**.

Large Formats Take Longer to Send

Imagine that a company keeps all videos
in 4K Ultra HD format (20 Mbps bandwidth).

What if a customer has only 10 Mbps?

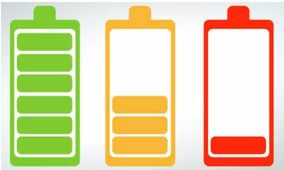
Does the company transmit the 4K format,
- forcing the user to wait for buffering, or
- to stall every so often to catch up?



Changing Image Formats Takes Time and Energy

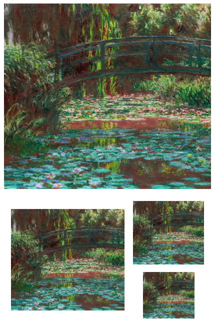
**What if a customer is using an
old phone with a small display?**

Does the company transmit the 4K format,
- forcing the phone to use battery power
- to reduce the images to fit to the screen?



Companies Keep Multiple Formats to Serve Customers

Instead, many **companies**
- **keep video in multiple formats**
- and **send the** format that **best fit**s a customer's network bandwidth and display capabilities
- (some companies allow you to override the choice, if you enjoy pain).



Transcoding Uses Significant Computational Resources

Converting video formats (**transcoding**)
- **requires a lot of computation**, thus
- costs a lot of money.

But **happy customers are worth money**!
- Unhappy customers have a bad habit
- of finding another company and
- no longer being a company's customers.

Transcoding also Used to Reduce Costs

Similar techniques are used for photos.

Compare, for example,
- a photo downloaded from Facebook
- with a photo from your camera's phone.

Facebook's app
- reduces the image size (on your phone/computer)
- before transmitting to the Facebook server,
- **reducing bandwidth/time as well as space**.

Terminology You Should Know from These Slides

- bandwidth
- buffering
- broadcast media
- multicast
- k-out-of-N code
- Hamming code
- video on demand
- content distribution
- datacenter
- locality
- transcoding

Concepts You Should Know from These Slides

- routers can drop packets
- connection bandwidth is the minimum over all links in a route
- bandwidth is shared between connections
- more delay means less achieved bandwidth
- transfer time = (# bytes) / (achieved bandwidth)
- multicast challenges: routers aware of multicast, unreliable delivery, variable bandwidth
- variability in video data rate and connection bandwidth
- use of transcoding to fit customer needs