Backstage Prep:

clubs

3 rubber duckies go in left labcoat pocket for two coats

Action	Talk
Morning. The team is planted through the audience, dressed normally.	"I'm in need of a couple volunteers for my talk on how to speed up the internet at 11:30. Is there anyone out here that would be willing to jump on stage and help out?" Two of the planted jugglers volunteer. Hopefully someone else does, or a plant... Go up to that person, and ask: "Is it ok that you get a little wet, and splattered with paint?". That person sits down, rapidly. "Uh, anyone else?" Third juggler stands up with great vigor! "OK, you, you, and you, could we go back and talk about it for a few minutes? Thanks for your willingness to do something crazy, for Science!"
Team vanishes with me, conference starts, team reappears with lab coats on (and a change of shirt underneath to the juggling t-shirt) and I too rejoin (IF I can stand being in the room, normally I'm a complete wreck, outside, chainsmoking, reviewing my text)	
If there's a coffee break, stay in the lab coat, socialize, talk about your studies, anything but the talk or juggling,	
Intro	Nancy does the introduction
	Thanks everyone for coming. In my coming talk we are going to try and tackle some really difficult networking topics like cryptography, the importance of Round Trips (otherwise known as RTT) – congestion control - how web traffic and videoconferencing traffic have trouble co-existing and my beta noir - "bufferbloat" - "network qeueing delay" - which if we can somehow fix in the coming years, will lead to a better internet for everyone.

	Could I get a show of hands? Anyone here ever heard of “Fair Queuing”? “Active Queue Management?” How about more basic stuff? TCP? UDP? WEBRTC?
	Those of you not paying attention, raise your hands? Queue Theory? Control Theory? - Control Theory is useful for the design of robots… and queue theory – have any of you ever been stuck in traffic and wonder why it was moving so slowly? Queue theory gives us the analytical methods for how analyze and predict the problems that form queues like that. I learned nearly everything I know about from TTI Vanguard’s founder’s wonderful books, which are all now sadly out of print.
	OK, there’ll be a quiz later. Now with my brave assistants here I’m going to start with the simplest most basic thing all of you are familiar with. First we’re going to describe how voice traffic used to work, and how it works after it got packetized. And then... We’re going to simulute loading a simple, basic, 1993-era web page and then hopefully get to explaining some of the big concepts I just mentioned.
Give end of wire to person S, S-C, C-D	Up until fairly recently, when making a telephone call, your sound impulses were translated into a “waveform” and the analog signal transmitted over wires via a “circuit switched” network, that was plugged in by various operators. San Francisco calling Dallas, Dallas calling ... Gradually we automated this task, but… it was a lot of wires that had to get plugged into each other to make a cross country call. After you got the call setup… you could talk to someone clear across the country...
We do a “wave” across the room	
D: Isn’t S: That C: Amazing	

Me holds up envelope.	We learned, back in the 60s, that we didn't need to reconnect all those wires all the time for every call. That if we instead packetized the audio, we could send little individual chunks, and leave the wires connected to each other all the time. And just put them in an envelope addressed to the sender. And have the network "automatically" figure out how to get it there.
D: Isn't S: That C: Amazing!	And we learned that basically, every 20ms, we could send a tiny fragment of a voice, over what was then called "The arpanet", and reassemble them at the end.
The three start to traverse the stage again	And we also learned that packet loss – for any reason, on such large chunks of data as large as a verbal word
D: Isn't C: Amazing!	[Grab the S out of line] … Packet loss was a real problem… and we have now spent many decades trying to compensate for packet loss, and jitter, and delay, and for that matter, reordering – one packet from the stream might take a detour around the moon and tons of stuff like
S: That.	A much better way of thinking about "Bandwidth" is not mbits per second, but kilibits per microsecond, 1000 times finer granularity.
Picture frame in front of my head,	Videoconferncing is very similar. You take a picture, then another picture, then another picture. So long as the intervals between "frames" are constant
Picture frame per word	It works ok. But web traffic is very different from this. All we care about really is that the page load completes in typically under 3 seconds. Movie streaming is similar also, you typically download 10 seconds of a movie, or more, before starting to play it.
Teams Get clubs	Here we have "" the DNS server. Over here is the "SERVER" that has the content you want, and you, sir, are the client.
Clumsy CLIENT throws club to DNS server DNS server replies REPEAT	Let's LOAD a web page. First up, we need to translate a domain name, like www.ttivanguard.com into a number, called an IP address. So we throw a packet to a DNS server which translates that name into a number, and it sends

	the number back to the client. The client attempts to connect to that server's IP address through "the cloud". Through a bunch of tubes, interconnected via routers (funnels).
Clumsy CLIENT throws club to SERVER SERVER throws club to CLIENT CLIENT throws club to SERVER	
Juggle for real! I am unrolling a 3 page foldout of something.	And after that, we start downloading the various objects – text, pictures, formatting in the web page. We might need to make a few more DNS queries along the way, or connect to other servers in the cloud…
Juggling gets more complicated	And we have this nearly continuous exchange of packets, or in this case clubs, transfering all the data… [me slowing down so the audience starts getting the joke], from all the sources and sinks… until, WALLA! YOU'VE GOT YOUR WEBPAGE!
Stop juggling. Pause for applause.	Isn't THAT AMAZING!
After I introduce each person, that person runs off to get setup for the next trick.	Ladies and gentlemen, the mit juggling club! MIT's work on networking is unparalleled, and it's not co-incidence that in this field, jugglers can be found, everywhere. Jaimie here is studing computer science at Hahhvard, Vasu, is studying "computer engineering", and joshua, when not juggling professionally , moonlights as a network engineer at ma.
	The web evolved. Things started to get more complicated after 1993 with the introduction of cryptography,
Cryptography CLIENT throws club SERVER recieves club (feel free to just make up a string of nonsense) CLIENT Pauses then SAYS A4KCURLYBRACE925AJK to the club and tosses club back SERVER Catches then SAYS [nonsense] throws it back CLIENT looks at SERVER's club (with a double take) and says more [nonsense] to his club and	

tosses it back. SERVER looks at CLUB, LOOKS at DAVE DAVE leans over to SERVER and says “This is how cryptography actually works” SERVER SAYS [string of nonsense] back at club + audience, tosses club. CLIENT LOOKS at club, says “This is how cryptography really works” Repeat for “You have to negotiate a secure connection” Repeat for “Then encrypt the data for each and every packet” Repeat “The importanta of cryptography is, that a man in the middle cannot understand what’s going on ” Throw pin, intercepted by DAVE, who repeats the string of nonsense, dubiously. Then tosses club to server. “It’s a LOT more complicated than this!”	
Reset	This brings in yet another complication. Because in the case of packets, the client and servers… are separate by an unknown distance… they might be really close together… one might be in the center of the stage – call that MIT – and another off to the left – call that harvard – and another to the right – call that the texas instutute of technology and science.
Whip out blind folds.	Worse, the clients and server are… blind. While you can measure the amount of time it takes for a reply to come back, you don’t know if that was the network, the server, the cryptotgraphy or the queuing delay that took the time.
	There’s ways that handle that over time.
	There’s actually three laws for how to manage this sort of thing – BDP, srqrt BDP flows, Power. And despite 50 years of internet development we still

	don’t know which of these laws is correct. So instead of using these things for the purpose of analogy, we’re going to use an entirely different law.
Two jugglers are separate by enough feet	The LAW of gravity. You need to have enough force upwards to compensate for the distance and speed you are sending stuff. Packets are sent at something close to the speed of light, so that is actually a constant for packets, but we’ve slown this down a bit for our purposes today.
Throw a ball too short	The packet might not have enough hopcount to make it to the cloud...
Throw a ball and miss	It might be directed at the wrong server...
Throw it at “just the right height”	It has to be thrown at the right height for the distance.
Move closer together, throw back	
	The height of the flow here represents the minimum amount of “buffering” required to ensure we defeat the law of gravity.
Move apart	Now I have to inject yet another a dose of reality into this. What actually happens on a web download is two GIANT data packets are sent, over “TCP”,
Throw two clubs Dave walks between the jugglers for the dialog	and then, a really tiny packet, called an acknowledgement, is sent back, containing a list of what was successfully received, and what was not.
Client pulls out duckie from lab coat	Dave: What’s that? CLIENT: It’s a quacknowlegement. :audience boos: Dave takes the duckie and walks back to SERVER That… quackknowlegment says YES I received the data, please send more.
Catch those big pins but put them down	Now when that, quacknowlegement comes back, something interesteting happens, to probe for both bandwidth and distance… we send not 2, but 4 more BIG packets
C: Throw two duckies back	And we get 2 more quacknowlegements back
Dave + Server send 8 packets	And then we send 8 more packets!!!
And we go boom dropping stuff on the floor, the dns guy starts picking those up and juggling them,	The SERVER guy is happy. The client on the other hand is not, his object didn’t load, so he

the client walks over and quacks a packet DNS goes picking them up with feet somewhat inobtrusively	sends back acks, saying that “we lost these three packets”
Switch to using pins. If at all possible lose (mabye grab out of the air) a packet now and then and find a way to throw that back while varying the height somewhat.	This idea is **fundamental** to the structure of the internet. We probe for more bandwith by throwing ever more packets, and when one is lost, the sender is signaled to **slow down**, and to fill in the missing data.
	Watch for a while
	If you were to imagine the internet consisting of trillions of jugglers, all obeying these two simple rules – again, lose a packet, slow down, retransmit, slowly increase the speed, lose a packet slow down, retransmit… sending these not through the air but throw a series of tubes, and funnels designed by a madman, you wouldn’t be too far off.
	I really want to hammer this concept home.
Setup – three jugglers 3 balls, 4, 5 facing audience…	Pay no attention to the balls themselves, ok? Use a ruler. Watch the hands. They are moving stuff from hand to hand at roughly the same rate, no matter if it’s 3 balls or 5, or seven. That is the amount of needed “buffering” for this trick, but in general the same amount of data is being transferred. It’s just going higher every time. You just have more balls in the air.
Then moving apart to throw at each other.	You need to have more balls in the air to traverse a istance. In a real world scenario – where we are separated by a distance, throwing 3 balls is very inneficient, And the crux of the problem I’ve beent trying to explain is how hard and yet how important that is, in the face of web/movie/file transfer traffic trying . Too fee packets in flight is bad for the web/movie/file transfer traffic, and too many packets in flight, is bad for voice and videoconferencing traffic.
	And the network is constantly changing up and down, not just with flows entering, probing and

	leaving the link, but with wifi and wireless where if you move a fraction of an inch, the available bandwidth can change by a lot. I entered this mad scene in roughly 2008, and applications that I relied upon, like skype, were acting up periodically, and I didn't know why. It wasn't until jim gettys identified the root cause – really exececessive amounts of buffering – that I understood.
	And it turned out that the various means we had developed until that time to figure out the "right amount of buffering" weren't working. And the amount of buffering we were observing was so much in excess of what actually needed, sometimes measured in seconds – which was totally ok for web traffic, but the impact of that, and things like bittorrent and netflix, were harming the network for the interactive applications we'd also designed it for, like voice, and videoconferecing.
	It's been a very long 14 years since. We've developed better end to end congestion control algorithms such as BBR and packet pacing, and we've also made great strides in increasing bandwidth, and in improving the routers to be more intelligent as to how to manage the flows of different kinds of packets.
	In the time remaining, I'd like to try and describe how two of the algorithms we've developed actually work to assist the network – enforced at the router – work to make it possible for
Blow-off Enter on unicycle with rings	
Close	And a huge thanks to our MIT jugglers – Jaime! Vasu! Joshua! For coming out today!

…

Dave: “But there should be three takeaways:

Is an envelope, and inside the envelope

Is a layer in the data. Details about the stream are sent in the clear.

Now I’m going to talk about Congestion Control

I’m going to take a huge step back in history now, because it’s important. Broadcast media.

You’ll note from the previous demo, that when I sent data – for most of human history, media was “broadcast media”, you wrote a pamflet, a broadside, put a song on the radio

In the case of electronic broadcast media – you had channels, or stations, and ultimately measurement means like the Nielsens arrived to determin market share, but the users were left with running up antennas on roofs, changing the channels.

Once upon a time we thought that the underlying components of the internet would go down that path, where you had that “channel selector”,

But the broadcast paradigm is something that haunts us in every thing we do on it, and to a increasingly horrifying extent, the underlying bi-directionality of the internet is being lost.

You got all this? The internet is a string of tubes, connected by funnels,

Factoid: Most web traffic is governed by Round Trips

Drops in half, if you have another flow

“lupanov stability” – can you say that? Great!

If you look at the speed his hand are moving at, he’s not actually transferring data much faster – he just has more balls in the air.

An internet that is optimized for web traffic is not ideal for doned esign

The way that humans think about mbits/sec, which is kind of ok for “web” traffic which takes 3 seconds to load, but voip, gaming, and videoconferencing traffic, you have to think about good behaviors at kilobits/ms. Reliable, smooth packet delivery on sub-16ms intervals is about the highest bound, and research into VR/AR suggest that something lower than 8ms, closer to 1ms, is needed.

For starters 1ms is about 1foot in air, so the real physical distance starts to really matter.

How many of you have ever had zoom fatigue?

I could also see (a cuttable joke) trying to explain how wireless had trouble with full duplex because of how the energy splatters, tossing a water balloon eagerly in my hand, aiming at a volunteer... beginning to wind up... and getting stopped by one of those folk "because the hotel won't allow it" and trying to explain that visually and talking about how fiber should be be called "fibers".

I think we have to incorporate a "ball person" to chase after stuff, but that person would be best in on the plot as well. I do have a couple sweet jokes for that person.

Milk in the balloon