I'm going to do a double proof of the fact that I've created my own project. And when it's final year, I'm going to bring back my spring. What are you doing? I'm going to put that one down. That'd be my name. That'd be my name. That's good. I have a red, and a black. That's what I'm going to do. I thought you'd model the process. I thought, and I hope, that's what people need to know. I want to see the painting where I do a hyperbolic painting. Yeah, that's fine. That's fine. But you're beautiful, mate. You're a buster, I mean. Just kind of like what I do when I'm in London. I don't know what I'm going to do. So I'm going to do a one-room painting. Except for the only one-room painting. It's not too dark. That's kind of fun. It is. Yeah, I just want to do a double proof. frame without detection other transmissions, it's done, right? This transmission is successful, so great, we are done. However, if the NIC detects another transmission while it is transmitting an existing frame, that means it's a collision. So, NIC will abort and send some jam signals to say, okay, I start transmission. So after aborting, when should NIC retransmit, right? When should the NIC retransmit? Should the NIC retransmit immediately? Should it? No, because if everyone retransmit immediately, then what happens? Another collision, okay? So you can't do immediate retransmission. So you need to do some kind of like a back-off, all right? And so to back-off, the NIC will choose a random number from 0 to 2 to power m minus 1, so I will explain it later, and wait the time that equals to send k times 512 bits, which is a half kilobyte, sorry, no, no, no, I'm wrong. So it's a half kilobyte, okay, we'll choose a number, k, random k, and wait for that random number proportional to k, and then return step 2, which means retransmit. So actually when you have one collision, when you have one collision, right, it will choose a random number between 0 and 1. When it has two collisions, it will choose a random number among 0, 1, 2, 3. When it has three, when it detects the third consecutive collision, so that means for the first collision, it will choose a random number, right, between 1 and 0, and then retransmit again. If that is another collision, okay, it should choose a random number between, I mean, among 0, 1, 2, 3, then retransmit again. If that's another collision happens, then a random number from 0 to 7, then a random number from 0 to 15, okay, so every transmission, the expected waiting time will grow exponentially, okay, each time the average or expected waiting time grows by 2, times 2, okay, so it's exponential growth, and this is called a binary, or it's called an exponential backoff, okay, and this was designed by the inventor of Ethernet, Metacoff, who got the Turing Award in 2023, but this random backoff was first invented by my PhD advisor, Simon Land, which he wrote a paper, okay, that's a Pew theory paper in 1970s, and in that paper, he proposed a random backoff, and then Metacoff used the idea, of course, he cited the paper, but Simon criticizes that Metacoff used the method but didn't use it correctly, so he said in the original paper, he explicitly proved that linear backoff, which means you grow the number of waiting time linearly is sufficient to avoid collisions, while Metacoff, from whatever reason, probably to be more conservative, he chose exponential backoff rather than linear backoff, okay, and Simon said that's definitely an overkill, you know, exponential, of course, that's you need to wait longer time, right, and linear, you wait shorter time, but he proves that linear is sufficient to avoid collisions, but Metacoff still uses exponential backoff, okay, and then Ethernet uses exponential backoff, yes, it probably depends on a lot of things, but is 512 bit times, like, usually longer than an Ethernet cable? No, no, no, Ethernet frame is usually like around a 1k byte, right, but that, like, is 512 bit times enough for, for the, I guess, for you to realize that someone else is saying, like, if you choose 0 and the one person said, and you can hear that, right, even though you can hear other other people are sending.

It's hard to say. And does that, like, put an upper bound on the length? Because you can assume even that this the length is not a problem, I think, the length is definitely not a dominant component in the latency of Ethernet, and the length is short, of course, if you use Ethernet, it's not a, you know, optical fiber, yeah. It's not used because that's a theory paper, that's Pew theory, yeah, yeah, he did, I think 1970s people, people don't have, didn't have personal computers. So what they do is just like, write formulas and prove the theoretical bounds or a certain method without actually trying it, right. But if it's proved, then it definitely work.

But when, when, when later people design more complex systems, and it becomes impossible to prove that the system can converge or stop or whatever in a limited number of time, then you need to do experiments, at least in simulations or in real implementations to prove that your, your, your method works, rather than theoretical proof.

Theoretical proof usually cannot be done for very complex systems. Like DHT, right? It's very hard to, oh, like DHT, and there are failures. It's impossible to use, you know, theoretical proof. Yes. So this is assuming that the average length is vital.

Yeah, no, I don't think that's just the standard in Ethernet. If every length is less, then probably I would do better by not having to multiply by not waiting for that long. Uh, yeah.

That's right. That's right. But, you know, when you, when you define a standard, you always choose a conservative number.

I guess like also zero to one is acceptable when one to two to two to three also be acceptable. Like you just have to, if zero and one are okay values. Yeah. It seems like linear would be like the first thing I would grab. Yeah.

Okay. You're right. Okay. So the efficiency of CSMA-CD can be modeled in the following same formula, which we have a T-probe as a maximum propagation delay between two nodes in the length and the T-trans is a time to transmit maximum size, size frame.

So, uh, we don't know which is bigger. Usually I think this is definitely bigger.

So, uh, the efficiency is around one over one plus five T-probe over T-trans.

Um, and if the prop goes to zero, then the efficiency will, um, almost zero.

And if the transmission delay is large, very large, then the efficiency also goes to one.

Okay. So you're either one, the propagation delay is short or the tran or transmission delay of to sending a maximum size frame is much longer than the propagation delay between two nodes.

Okay. So, um, we still have another part of CSMA that's CSMA-CA and, uh, we will discuss it in the chapter of wireless networks. So now let's focus on the Ethernet. Ethernet uses, uh, network address, another type of network address called MAC addresses. And MAC addresses is different from IP. We know that IP is a, we can consider IP is an identifier of your computer on the internet, but in fact it's an identifier for an interface. It's not a computer, but sometimes if you say that's an identifier on a computer, then people will say, okay, yeah, you're not wrong.

Um, it's used for layer three, which is a network layer, routing and forwarding. Right.

So in MAC, in Ethernet or, uh, layer two or the link layer, that's another layer, that's a layer below the network layer. Okay. The devices like Ethernet switch use a MAC address of a host.

MAC address is used to identify the sender and the receiver of a frame. A frame is an Ethernet frame, which means a link layer packet.

All right. And MAC address is 48 bit for most of the network. Okay. And it is burned in the NIC memory.

So sometimes you can set the MAC address, but most of the time the MAC, when you buy a computer and the computer comes with a NIC, then the MAC address associated with that NIC will not change. So we all, we usually assume the MAC address doesn't change. Although in extreme cases you can still change. Yes. I was going to ask about like MAC address. Yeah. You still can change it using some techniques, but most of the time people won't change it. Okay.

Okay. So a MAC address looks like this. It's, it's, it's include, it includes, uh, 12 hexadecimal numbers and each hexadecimal number is four bit long, right?

long, right? So in total it's 48 bit. And you, there's a analogy that you can, you can think about the MAC address and IP address. So you can think IP address is your home address. Okay. When you move from one resident home to another, you change your IP, sorry, you change your home address. You will need to inform the DMV that you have a new address. So please use this new address for me and you need to inform the USPS that you have a new address. Right. Um, but MAC address is just like your social security number. No matter where you move, okay, as long as you move inside the United States, your social security number doesn't change. Okay. It's like your MAC address. When you have a laptop, you move the laptop around and from your home to the classroom and the IP address of your laptop definitely change. And it will, uh, when it's in the classroom, then it will get an IP address from the UCSC campus network. So it will has a, it will have a prefix of the UCSC network. When it's go back to your home, your home use, I don't know, AT&T, then it will has AT&T prefix. So the IP address could be very different, but the MAC address of your computer will not be different. We'll still keep the same way when you move your laptop. Right.

So for some applications, for some applications, people would like to use my MAC address to identify, uh, each computer, for example. Okay.

Okay. If you are, and use a campus network to do something that is illegal, for example, you are downloading copyright protected content, right?

content, right? And it is detected by the, uh, campus network and the campus network want to list to your computer in a blacklist, then should you add, should they add your IP address to your back? Or MAC address to your back? MAC address definitely, because that's making nonsense to add the IP address. IP address is theirs. It's UCSCs. It's just a temporary assigned to your computer. Next time when you, if they add the IP address to the blacklist, next time you connect to the network, it will be assigned to another, with another IP address. So that makes no sense. You are still not blocked, but when they add a MAC address, your computer is temporarily blocked. Okay. You need to, okay, maybe apply to, uh, on the suspend of your computer, or you can do some, you know, technology stuff to burn another MAC address to your NIC. I don't know.

Probably that's doable. Um, okay. So for some applications, people want to use MAC address. And also for Ethernet switches, they look at the MAC address.

Um, in an Ethernet, it's connect, uh, it's connected by a either, even a switch or a wireless access point. And everyone use the MAC address to identify a computer, more precisely to identify an interface or adapter.

And then MAC address is allocated and administrated by IEEE.

and administrated by IEEE. That, that's the Institute of Electrical and Electronic Engineers. It's, uh, one of the biggest, um, organization of EE and CS and CE, uh, people to, you know, organize the conferences, journals, and, uh, memberships, and also administrated. And the manufacturer of computers, for example, Dell, HP, uh, Apple, right?

Dell, HP, uh, Apple, right? They are manufacturer of the computers. They buy a portion of MAC address space so they can make sure that MAC address, they assign it to those computers are unique. Okay. And they should not repeat. If they repeat, then some trouble will be caused. So you also cannot assume in a local area network, people will share the same prefix on their MAC addresses. For example, in this local area or in the classroom, right? We have so many mobile devices connected to the, uh, wireless access, access point, and they are from a different manufacturer and they are from, they are bought in different time, right? So their MAC address, they definitely are different.

We cannot assume they share the same prefix and they actually do not share the same prefix. So MAC address cannot be aggregated in most of the time. All right. IP address is assigned by the campus. They share the same IP prefix so they can be aggregated. MAC address cannot.

So it's just like MAC address is your social security number and IP address is your postal address. When you move to a new place, you are assigned to assign with a new postal address and people actually can see some hierarchies from the postal address, just like the prefix. So for example, you and your neighbors share the same street number and the city name and the county name, right? And probably the same post post code.

Okay. That's, that's just like prefix.

And for MAC address, we, we can call it splat addresses and it's portable.

A portable means it can move with the network card from one network to another network and it will not change. So it's called portable.

And IP has a hierarchical address.

address. Hierarchical means you can look at the prefix and you can determine what kind of subnet it is, right?

it is, right? You can guess which location the IP address is in because the IP address contains the network location information.

That's why when you search a certain IP address on the internet, you can, some website will probably will return on approximate location of you. Although they cannot precisely identify your location, for example, you use AT&T and the user IP address, people probably can't locate you in, you know, Monterey Bay area, but they cannot, you know, precisely locate into a resident house. But maybe in Santa Cruz, maybe in Monterey, maybe in Salinas, I don't know.

That's because, that is because your IP address contains some information of the network locations.

But MAC address definitely develop.

develop. Okay. All right. So the next question is when I want to send some, a message in the LAN, local area network. And I know the destination IP address, but doesn't know it's MAC address. How could I know it's MAC address?

address? First, let me ask another question. How do I know the destination IP address? There are multiple ways.

Can you name one way to know the IP address of the one I want to talk with? Yes.

No, no, I'm just a normal user. I'm not a network administrator. So how can a normal network user knows the destination IP address? Yes, right. This is a way to know that. And there are other ways.

But you only need to know DNS. And okay. So now suppose you know the IP address, then what is the way to know the MAC address? The way is called address resolution protocol, ARP, or ARP.

So each node maintains an ARP table, and ARP table contains the entries like the IP address, MAC address, TTL. Okay. The entries look like this. The entry, each entry contains three fields.

IP address is a search field.

MAC address is a return value field. And TTL is the field to manage this entry. Okay. And basically, this entry means when you have the MAC address, you can search the MAC, sorry, when you have the IP address, you can search the IP address in the ARP table. And when there is a match, the ARP table will return the corresponding MAC address to you. So you know the MAC address that corresponds to one certain IP address. Okay.

And what is TTL? TTL is time to live. It's actually a counter.

It's counter time. And after this amount of time, the address mapping, the entry, will be removed from the ARP table.

So ARP table will not grow infinitely. Okay. It will grow where you have new mapping stored, but after some time, it will be deleted.

And it's typically, it's 20 minutes. So then you ask, what if I need this mapping, you know, after 20 minutes? In fact, if you use this mapping frequently, okay, frequently means you will need to send some traffic to this MAC address.

So when you use this MAC address or search this entry, the TTL will be reset to 20 minutes. That means if that is a mapping you use frequently, it will keep refreshed, refreshed, refreshed, and it will not disappear. But if that is a mapping that you use that for once and never use that again, it will be removed after 20 minutes.

Okay. And this mechanism is called soft state.

Soft state means when you don't use that, it will be automatically removed. And hard state means no record will be automatically removed unless you explicitly ask it to remove it. It's just like a database record, right? Your student record is stored in UCC and it will be there forever unless someone explicitly removed your record. But it will not automatically disappear.

Okay. So here is a process to run the opt protocol. Suppose a host A wants to send a datagram to another host B and it was first to search using B's IP address.

And now after searching, B's MAC address is not in R's opt table, which means A doesn't have this entry, right? So if A doesn't contains the entry of B, A will broadcast an ARP query packet containing B's IP address. And it will use a destination MAC address as FF. So the F means 1111, right? A hexadecimal F is 1111. So all Fs means all ones. All ones, 48 ones. 48 ones means this is a broadcast destination.

A broadcast destination means everyone received this message will look at this message. So all nodes on the local area network will receive the ARP query and all nodes will look at it, which includes B. So B will be one of the receivers of this broadcast message.

And B received the ARP packet and replies to A with B's MAC address. And when B sent back, okay, the message A, I mean the datagram A sent to B contains AC information.

So B knows A's address clearly. And this time B doesn't need to send a broadcast message.

It can simply send a unicast, unicast message to A.

message to A. And the destination of this unicast message is A's MAC address. Is it clear to you?

you? All right. So when A got the reply from B, A will save IP to MAC address pair of B in its ARP table on two timeouts.

So we can call this saving as cache.

Because cache is to save temporarily. And cache is a memory space that will automatically disappear, okay, or be removed.

So you can call it cache. So A actually cached the information in the ARP table.

And this is the concept of soft state. All right. So now let's look at our more comprehensive process, which involves both the Ethernet and a router. So we assume there the router R connects two networks.

And inside each subnet, there is an Ethernet switch or wireless access point connected to machines. So now A is a user, and A wants to send a datagram to B.

And the datagram will travel through R. Okay.

So assume A knows B's IP address. So A will know B's IP address. That is two, two, two, two, two, two, two, two. Okay. You can know this IP address using a few ways. And one possible way is DNS, as you said.

And also A knows the IP address of the first hop router. And here, the first hop router is R. And you may find that this R contains two interfaces.

And each interface has a different IP address and MAC address.

Right. So for a router, different ports has different addresses.

So this router has this address in the left port and this address on the right port, which they share the same prefix with the subnet. And their MAC address is completely, R completely different. So A knows B's IP address of the first A knows IP address of the first hop router. And how does A know that? Because if you still remember what I taught in 150, there is a protocol called a DHCP, dynamic host configuration protocol. When a new node, a new computer joins a network, it will run this DHCP protocol and ask a DHCP server to get a new IP address to the new computer. Right. And the information included in the reply, DHCP server from the server does not only contains the new IP address. It also contains the IP address of the first hop router. So it can, the new computer can use that information, save it, and use that information in the future to send a message to an external address.

So then we, A should also know R's MAC address, which is this E6E9.

How does A know R's MAC address? Yeah.

From R? From R. A can first send an R broadcast, and R will receive it. So R will reply with its, the R broadcast contains R's IP address, and R knows that A is requesting its MAC address. So it will reply with its MAC address, and then A will save the IP address, MAC address, mapping to the R table. So next time when A wants to send another message to R, it can use unicast, right? So A knows R's MAC address. So A will generate a package or more precisely an IP datagram, and it will add an Ethernet header to the IP datagram and make it an Ethernet frame.

So the Ethernet frame contains the MAC, the source MAC address, and destination MAC address. And the IP datagram also has an IP header. IP header contains the IP source, the source address, and destination address.

So the IP source address is certainly A's IP address. And IP destination is B's IP address.

Make sense? The MAC source address is A's Ethernet address.

But the the destination MAC address is not B's MAC address.

address is not B's MAC address. So who's MAC address?

MAC address? It is the routers. Why this is routers? Because when A wants to send a packet to B, right, the router is the last step, last stop in the Ethernet, in the subnet. After the router, it's not in the same Ethernet. It's not in the same subnet, right? Okay, so the MAC address is just the destination address in your subnet.

Okay, all right, so this packet will arrive at the router, and the router will drop the Ethernet packet header and recovers the IP packet.

Because the router operates on IP, so router will extract the IP packet, right, or IP datagram.

extract the IP packet, right, or IP datagram. So the router use the IP datagram and generate a new Ethernet on the other side of the network, which is another subnet, another Ethernet. Okay, so who should be the MAC source MAC?

Okay, so who should be the MAC source MAC? Who should be the destination MAC? The source would probably be the router, and the destination would be B. Yes, the source is the router itself, but be aware it should be the port on the B side, not A side. Okay, a router has multiple MAC addresses, and the destination MAC is B's MAC.

Okay, a router has multiple MAC addresses, and the destination MAC is B's MAC. So, and outside of Ethernet, the Ethernet header will be removed to replace with another one, and another one is on their Ethernet, okay, which is completely different than the original Ethernet. Yeah?

Is there a reason why all the ports on routers have different MACs?

Yeah, I think so. Is there like a simple reason, what's the reason?

Is there like a simple reason, what's the reason? The simple reason is that they use different interfaces, and every interface needs to have a MAC address. So since they use different interfaces, there are multiple MAC addresses.

Same thing applies to your laptop, Ethernet port, and Wi-Fi or Wi-Fi. They all have different MAC addresses. Oh, yeah, definitely. You can have your computer, your computer has two different LANs.

Sure, you can.

But I think most of the operating system only allows one network connection.

You can connect both to both networks, but your operating system probably only will maintain the network stack for one of them. I can confirm that.

Yeah. All right, so it arrived at the B, and it will again arrive to the IP layer or network layer, and the drop the MAC header, and the B will eventually receive the destination, the datagram. Okay, so the Ethernet switch is a link layer device, and even that switch will look at the frame's MAC address, and selectively forward the frame to one or more outgoing links when the frame is to be forwarded on the segment, then use the CSME CD to access.

However, nowadays, different ports, they have different collision domains. So every port will have a separate collision domain. So nowadays, that still runs CSME CD, but you don't expect them to experience collisions. Collisions from multiple ports, no.

Collisions from multiple ports, no. And Ethernet is plug-and-play and self-learning. So what is a plug-and-play?

So what is a plug-and-play? That means when you plug the Ethernet cable to your computer and another side to the Ethernet switch, both of them, they do not realize the existence of, sorry, let's see, oh, your computer actually do not realize the existence of the Ethernet.

So when your computer connected to a router, definitely it knows the router, because it needs to know the router's IP address. But when it connects to the Ethernet, it doesn't realize there is an Ethernet switch. And it may think that I probably directly connected to a router, I probably connected to one Ethernet switch, I probably connected two Ethernet switches, they are multi-hop, but they don't know by looking at your computer. And they don't need to do extra steps to set up the network, like DHCP, those kind of things. They don't need to run DHCP. The Ethernet address, they don't change. They will always be there. And the ARP table, you don't need to set up the ARP table, because at the beginning, all ARP tables will be empty, and you can self-learn the other MAC addresses on the ARP table, right? So, even a switch is plug-and-play is self-learning, and it's transparent to the computers. Hosts are aware of the existence of switches.

And the modern switch supports simultaneous transmissions.

In this example, they have four interfaces, and each interface connects to one computer, and there's no collision on different links.

So, it's kind of like full duplex, because you can send and receive at the same time on each link, and multiple links can operate on the same time, and there's no collision.

The switch can simultaneously support the packets transmitting from A to A' and B to B', right? So, we have six ones, A, B, C, D, sorry, A, B, C, and A', D', D'.

and A', D', D'. So, we say that an EvenS switch is self-learning. That means you don't need to configure anything on it. Then, how does a switch know A is reachable from 1, and A' is reachable from 4, and C is reachable from 3?

from 4, and C is reachable from 3? How does it know that? Okay, it also can be done by self-learning. Each switch has a switch forwarding table, and each entry will look in the in the switch table, in the forwarding table, will look at it like this. It's a MAC address and the interface to reach that post of that MAC address and the timestamp.

post of that MAC address and the timestamp. So, timestamp is the last time they visited the use of this entry, okay?

It looks kind of like similar, like an IP router forwarding entry, but just like an IP router, the matching field is IP address, right?

right? But now, here is a MAC address. Looks like different, because IP needs to need some configuration, but now, switches, they don't configure their tables.

So, how does they know new tables, new table entries? How do they remove the table entry, old table entries? Okay, do they actually run something like a routing protocol? Actually, not.

So, switch will learn the posts and their corresponding interfaces. So, when A wants to send a frame to A', A will send a frame to the switch, right? It will first arrive at the switch, so the switch will know that A is reachable from interface 1.

Does it make sense? Okay, it will immediately remember A1, and then the destination, the switch will realize the destination is A', right? But it doesn't know which port to go to A', which link to go, so it will simply broadcast the frame to all network, or host in the network, or we call it flood.

And one of the receivers should be A', so the A' also is included in the receivers, and A' get the frame, so the A' will send back a frame to the switch.

Now, the switch received a packet from A' and to the destination A, and the destination A is actually included in its MAC table, right? And also, it will put A' and a 4 into the MAC table.

So, since it knows the exact port number to reach A', it will simply use a unicast to send a frame to A' instead of flooding.

Okay, and Ethernet actually can be connected together, and we call it a bridge, Ethernet bridges, because Ethernet can be used to, you know, go multiple hops.

go multiple hops. And how does A' know how to go to H or G, right?

go to H or G, right? How does A' know? It also can be done by self-learning.

It also can be done by self-learning. So, A' can first send a packet to S1, and S4 receives the flood information, and it will also flood to the other network. And S3 will also flood to the other network, and the other hosts in the network, and eventually G will receive that. And G will give a reply to S3, and then S4, then S1, then back to A, and then S3, S4, S1 will know how to get G. Okay?

Okay? Yes? So, what this means, you send a signal to send the frame to everyone.

Basically, yeah, not until G, because G won't ask others to stop, so it will send to everyone. So, the first time you send it, it sends to everyone, and then after that, the sender makes note of who received it, and then... Yes, yes, yes. Yeah? Yes.

Yeah, S2 will send to DEF as well, because S2 doesn't know who are inside its network. Like, after S2, then it's going to record which one to go to, and then A wants to see it. Mm-hmm.

Okay. So, an actual network, including institutional network, yes. Back on the previous slide. Yeah. So, would S2 not know where to find G still, or would G... Yeah, S2 still doesn't know where to find G. S2 would have to send what to DEF and S4, but then S4 would know where G... Some of its own host send a message to G and its backwards S2 will know. And then S4 already knows. Yeah. Okay.

So, an actual institutional network will consist of the routers and the switches, and sometimes it says switches only, sometimes router only.

I don't know. Is there a... Is it router? But it maybe comes with extra switches or not.

Okay. And it's possible that just one router and everyone is connected to that router, and that's it, right? But if that is for a large organization, it may use switch to extend the network a little bit.

And also, an institutional network probably also includes a mail server, a web server, because the organization will have its web page or have its mail, email servers, so they can... The users in this network can receive the emails, but we know that UCSC outsourced our email to Google, right? So, we actually do not have a UCSC mail server. We use Google's mail server instead.

Yes, question. What if having one router would be sort of like a single point of vulnerability, right? So, wouldn't you have usually have multiple routers connected to the outside world forward? Yes, we usually have multiple routers.

Yeah. And after router there are switches. Okay.

So, when we compare switch with routers, they both are devices that store and forward packets. But one is one forwards network layer packets, which is a datagram, and the other forwards the link layer packets that are Ethernet frames. They both have a forwarding tables, but one is IP to port mapping, and the other is MAC to port mapping. Okay. And one is use routing algorithm to configure the IP table, and the other use flooding or self-learning to configure the MAC table.

And these days, actually, people do not pay a lot of attention to these days when you buy a router, usually that includes a function of switch. So, when you buy a wireless router, that's a single box, but actually that includes the function of a wireless access point, right? And a switch, Ethernet switch and a router.

So, it's like three in one box. And so, nowadays, when you call a box a switch, it may also include the router functionalities, like it can look at the IP address, it can look at the IP addresses, the protocol type, the IP you can look at MAC addresses. So, nowadays, we don't, you know, do a lot of separation of routers and switches.

So, in our next presentation, we will learn data center networks. In data center networks, the developers and operators, they usually call their boxes and switches.

But even when they call their boxes as switches, the switch also includes a router functionality. It can look at IP address. It actually works on layer three. And if you have experience working on this open flow switch, open v switch, right? Some of you have the turns. Open v switch can look at a lot of headers, including the Ethernet header, an IP header, and even the TCP header. You can look at the TCP port number and the TCP sequence number. So, it's a very powerful switch, although the name is switch, but it's just not limited to the classic switches. Okay.

So, let's talk about data center networks. Data center networks is a super important component of today's internet.

So, you actually talk to data center network every day.

When you use Google Drive, when you use Gmail, when you use Facebook, when you use chat GPT, right, deep seek, whatever internet server, you are actually talking to a data center network because the service is hosted in a data center network. And the environment of the ecosystem that data center network hosts the services is actually called a cloud computing, right? Cloud computing.

Or you watch videos on YouTube, Netflix, on other, you know, stream video, web pages, or you play a game on Steam, and you download a game from Steam, and then you also talk to data center network. And in a data center network, it includes thousands, tens of thousands, hundreds of thousands, or millions of hosts.

Okay. They are deployed close to each other, very close in physical proximity.

For example, here, you can see this is Microsoft data center container, and it includes the, you can identify a rack, and each rack includes multiple servers. The servers are running there.

Each server are connected to a cable, and cable is connected somewhere else. Okay. So, this is a data center network. How to organize data center network is a big problem. And actually, the challenges are, there are many challenges.

For example, a data center network runs multiple applications. Each serves massive number of clients, and the network needs to support so many hosts.

So, it's very easy to cause congestions, to cause data bottlenecks.

So, we have to manage the traffic load, how to balance the traffic load to avoid the network bottleneck. And also, the bottleneck is not just on the network. It can be on CPU, it can be on memory, it can be on disk, right? It can be multidimensional.

So, the performance is very important of the data center network. And it's not just a performance. Yes. Isn't causal to the huge concerns? It's huge, it costs definitely. Yeah. Because it's very different to cool the servers. So, I noticed a lot of data centers are moving towards ARM processors, because that architecture assumes less electricity than equity. Exactly. So, in addition to the performance, electricity and the cooling are also challenges in a data center network. Because you run those computers together, their CPUs will generate heats. Then you need to cool those servers down, otherwise the CPU will burn out or crash. And that's a big problem, right? And you probably can also, to be more environmental friendly, you probably want to use the heat to, for example, heat waters or do something useful, right? Rather than just directly send the heat to the outdoors. So, that's a very large engineering work of building a data center. And usually you want to build a data center in some cool places, like, I don't know, Washington, Oregon, or Canada, Norway, those places. They put the containers. How does the cooling, the cooling is done by the ocean water? Usually, data centers are also set up in places where there's electricity. Yes, that's right.

Exactly. The real question is why we don't have, take the, use liquid cooling to cool CPUs, take the water and use the correct scheme. Interesting.

Yes, yes, that's a great idea. I believe people have thought about the idea. I don't know if someone has used the idea. Okay, so there are some other basic components in a data center, and one is called a load balancer.

What is a load balancer? A load balancer means there could be multiple servers in a data center network that serves the same type of customers. For example, if you go to like 3w.facebook.com or amazon.com or youtube.com, right? Not everyone goes to the same server.

Not everyone goes to the same server. If everyone goes to the same server, the server definitely crashes because at the same time, there could be millions of requests from the different parts of the world who want to visit those services and those large popular services. Okay, and that's not even, not considering the cases like Black Friday, something like a lot of people would like to go shopping, right?

a lot of people would like to go shopping, right? So the service provider needs to deploy multiple servers to serve one purpose or one webpage. And in this situation, a load balancer is necessary that when you receive a million requests, they are both targeting on the same type of service. But actually, they should be distributed onto different servers. And this is the task of a load balancer.

So when you, for example, when you open the page of Facebook, right, you get the IP address from DNS. And you go to the Facebook data center, and you arrive at it.

So the IP address returned by the DNS is actually the IP address of the load balancer. It's not the IP address of the event server. And at the load balancer, the load balancer will revise your packet and replace the destination IP address to be the actual server destination IP address.

actual server destination IP address. And for different users, they actually will replace with different IP, different servers, right?

IP, different servers, right? And one challenge is that they need to ensure that no server is overwhelmed. And another challenge, if you can think about, is that if there are a sequence of packets from a same user, right, they should go to the same server. You should not break a sequence of packets from the same user to different servers, because other servers don't have the context of the previous conversation. So there are also multiple challenges in designing a load balancer.

And we see that a rack contains multiple servers.

And all servers in the rack may connect to a so-called or switches. It's a top of rack, top of rack switches. And I think at the beginning, it is put on the top of the rack. So it's called a top of rack switch. And the top of rack switch are also connected by other switches. And other switches are also connected. And eventually, they will form something look like a tree, right? Okay, looking like a tree. So this is a tree data center.

Yeah, let me switch to another set of slides. Okay, let's talk about data center networks. We will discuss a paper that is written, oh, we will discuss a paper that was written more than 15 years ago by a group of UC San Diego people on how to design a data center network architecture or the data center network topology.

architecture or the data center network topology. And they were later hired by

Google to deploy their Google data center network. And I think the professor, Amin Vahada, he is now the vice president in Google, if I remember correctly, right?

right? Okay. Yeah. For managing their Google cloud infrastructure, which is data center network.

Okay. And at that time, he was a professor at the UCSD.

So Google simply used the idea presented in the paper to build their data center network. But it actually not exactly the same.

But it actually not exactly the same. Because, you know, when you, when you propose an idea, and you, you write a paper, and you want to deploy it in practice, then you will meet, you will see some, you know, practical problems, and you need to keep changing your original plan. So the right now school's data center networks is quite different. Okay. But the key idea is still like this. All right. The idea is still like this. Hey, which why, why are we repeat that again? Okay, so when you look at this, this, this topology, or this organization, that this is tree like topology, right? So there is a potential problem is that suppose each server will communicate for a maximum bandwidth of 10 gigabytes per second, right?

gigabytes per second, right? Then we have 10 servers per rack, the top of rack switch needs to support 10 times 10, which is 100, right? The gigabyte per second. So the tier two switch supports four racks.

So it needs to support 400 megabyte, sorry, gigabyte per second.

megabyte, sorry, gigabyte per second. And the upper layer switches need to support even more. And even with that, so many layers, they can only support like how many?

only support like how many? 16 times 10. That's only 160.

Okay. And in fact, Google contains like millions, several millions of the servers. So that means when you add keep adding layers to a tree, right? That means you need to you, you have two choice, and you must choose one of them.

One is that you will experience congestion on the top layers. Second, you must buy expensive switches or routers on top layer to support higher bandwidth.

Because higher bandwidth requires the silicon inside the switch to be more kind of like carefully designed and built.

And that requires high cost.

So more expensive, a more powerful switch, that means more expensive switches. And that can be much more expensive than an ordinary, for example, the 100 gigabytes switch.

Okay. So you have two choice, either you have over subscription ratio.

Over subscription means if the servers under this switch, they all send at this full bandwidth, the switch will be congested. Okay, this is called over subscription.

So the hope is or the expectation is that not everyone will send at the same time. But in fact, there are some traffic or there are some workload will require or no stand at the same time. For example, a MapReduce. I don't know if you heard MapReduce.

MapReduce is used to, you know, for example, a machine learning training. You want the data to go from one server to another server to train a machine learning model. Okay, they use a framework called MapReduce. And it may cost, you know, everyone send at full bandwidth.

And if there's over subscription, then the high layer net router will be congested and basically crashed. Okay, yes.

Over subscription is basically using the entire bandwidth of these servers.

Over subscription means your hope is that not everyone will speak at the same time. Okay, but if everyone sends at the full speed, full bandwidth, then the router will or the switch will not be able to support this bandwidth.

bandwidth. Yes. Doesn't it matter if it's only interact communication?

Oh, yeah, so yeah, you can you probably will can limit the if you want, if you can only send it to the other neighbors in the rack, you can reduce the communication. But there are only nine other users in the same rack, or there are only like 20 other hosts in the same rack. Majority are still cross rack, right?

right? Yes. So you're saying that you would take you have you assign more users to a server than usual in the hopes that they won't overwhelm the server. What do you do? It's not user, you can think that they are owned by a same user.

One user deploys those virtual machines or software on the servers.

And they run a same task together. For example, training a machine learning model.

Okay, that's that's that's simply one user. For example, Facebook status and there's just one user, which is Facebook. Yeah, I just say that you sort of over some current you sort of assign more.

Yeah, assign it. They don't send at the same time. Oh, yeah. But in fact, there are some cases they will send at the same time.

Okay, the so that means if you use, you know, at that time, at that time, okay, the majority bandwidth is one gigabyte. So you use a one gigabyte top rack switch. And the top rack switch will connect it to end of row switch, which is you have a row, right? You have a rack, you have a row, and the so called end of row switch, and then end of row switch should be more powerful than top rack switch.

And then the core switch could be even more powerful than end of row switches. And if not, use the same capacity, then it will have an oversubscription ratio, certain ratio, for example, to 2.5 to one or eight to one.

Okay. The current bandwidth, of course, is much higher, it can be 10 or even 100 gigabytes per host.

Okay. So the design goal of the paper, I mean, the old paper is that they want to find a interconnection of switches that can support the host to communicate at the support bandwidth without oversubscription.

Okay. And they don't want to use super powerful switches, they want to use commodity switches, that is a normal switches you can buy from market. Okay.

And they want to be backward compatible to Ethernet and IP. If that's not compatible to IP, which means they build their own routing, I mean, the network identifier, build their own network stack, which means what? It will cause a lot of troubles.

You need to install a new OS on every computer, right? Because the network stack is included in the OS. So you don't want to install a new OS on every computer, so you want to keep an IP and Ethernet. All right.

So the topology they use is called a factory, but factory is from the paper. Okay. But it's a wrong name. It shouldn't be called a factory. Factories, they also made a mistake, and which actually misled a lot of others to call this factory.

The factory was originally invented for supercomputing, and it was another type of topology, not this one. This one should be called as CLOS. C-L-O-S. Okay. And if you find the later papers from Google, they switch the name back to CLOS.

Okay. It's also an interconnection, but it's slightly different from this one. Yeah. You can Google it. I remember factories more in the context of operating systems, how operating systems arranged storage in the DIC.

Also, that could be totally different. Yeah, that can be different. Okay.

Okay. So in this one, we have a few hosts, right?

hosts, right? And they are divided into pods. So several hosts, several servers, several hosts are in the same pod, and each pod contains a few switches, and those switches are top-of-rack switch and aggregate switch.

Okay. And there are a group of switches there called core switches. So there are only three layers. They won't add more layers. Even if they have millions of hosts, they will increase the size of number of pods or how many hosts contained in the same pod, but they will not increase the levels. So there are only three levels, core layer, aggregation layer, edge layer.

Okay. Or top-of-rack Okay. So all computers, they divided into k ports, and in each rack, there are sorry.

Okay. There are k over two servers. So in this example, k equals to four. K equals to four.

Okay. So each rack has two servers, four over two. That is two. Okay. And each port also contains two top-of-rack switches or edge switches, and each port also includes two aggregation switches.

And there are k over two square core switches, which is two square, which is four. All right. Four switches. So there are four switches. And k can be an integer number that is 10, that is 12, that is 16 or, you know.

But please be aware that if you choose a k, then each switch needs to support k links.

Okay. So k cannot be infinity because you have nowhere to buy a switch that supports like, I don't know, 1,000 ports.

Probably there are 1,000 that exist, but people don't use that. And that would be very expensive. The majority, the switches we can buy is like 32 ports, 48 port or 128 ports.

128 are already very high-end switches, 48 maybe. So the k is also the biggest number of ports that or links a switch can support. So now we basically find that it's a small network, and we use very, very simple switches, and each switch only

needs to connect the four links.

And there are 16 servers. So that's a small data center network.

Okay. And the factory topology actually is equivalent to a tree that for every layer, the bandwidth will increase or basically will times k.

Okay. Every layer this link will, there are like, we have multiple links, right? We have multiple links to the upper layer, but it can be equivalent to we have one link to the upper layer, but the link bandwidth will be k times bigger than the lower layer.

Okay. So, and wait.

and wait. Okay. We still have a few minutes. So in the, in this network, give an arbitrary standard and arbitrary receiver. How many paths are there in the, in the network?

are there in the, in the network? So we have one pass. Okay. And another one and another one and another one.

Okay. Four. So it has a k over two square shortest pass on the topology and each shortest pass will pass through one core switch.

So there are that many core switches. So there are that many shortest paths and there were multiple paths. Yes. But the slide says, what either the slide, it didn't be k squared over four, not k squared over two. Right. So there were multiple paths and you can, the server can choose one of them for the routing.

So actually it's possible that we can do multi-path routing. Okay.

But we needed to do some, we first want to run a process called Addressing. So what is Addressing? That is, we want to assign an IP address to every host and every router and every switch, sorry, every switch.

And so now we don't make a difference between a router and a switch. Then this, even this is switch, we still use an IP address to, to, to it.

Okay. So we use an IP address, like 10 dots, something that's internal IP address, which means you can use that for every internal network. And they do not talk to the external network. Let's assume they only talk to the other host inside.

other host inside. So we don't, we don't consider external traffic. The external traffic, it probably can be handled by load balancing. And how can we do multi, multiple paths?

And how can we do multi, multiple paths? There is a protocol called ECMP.

is a protocol called ECMP. So if we have some equal cost multiple paths. So for example, in the previous factory, we have four shortest paths and those four paths, they are equal cost, right? They are all, they are, they all are shortest paths. So they have equal cost. And for this ECMP, we will schedule the packets based on the hash result of the CRC16 of the, the CRC16, which is, can be considered as a hash function. And that's a very fast hash function that's actually implemented on the network card. Why is CRC16 is implemented on a network card?

Why is CRC16 is implemented on a network card? Yes. Yeah, it's used to compute the checksum.

Yeah, it's used to compute the checksum. Yeah. So we can use, we can reuse the CRC function and we use destination and source address, right?

function and we use destination and source address, right? And use the CRC16 to calculate hash value. And we will divide the hash value into four spectrons.

And once it falls to one of them, it will be assigned to one of the paths. Okay. It's for four different paths.

It's for four different paths. So why we want to do a hash on the source and destination address, rather than just on the source address?

Yes. Yeah. So we want to make sure that the traffic from each source and destination is equally balanced. Yeah, we want to, the traffic to be balanced. And we also want to the same sender to the, the package from the same sender to the same receiver will always keep on a same path. So if the source and destination are the same, they will always be assigned to a same path because the hash value is fixed for a same input, right?

because the hash value is fixed for a same input, right? It's pseudo random. And why we want to make the packets from the same destination and the source on the same path? Yes. That way, that way they get to, I don't know how you say it, but it is a consistent path to sending between two pairs.

Uh-huh. All right. We'll have to recalculate the normal path. What if, what if they are sent on different paths? They send on different paths. It might take a bit longer because it's not working.

Okay. Yes. The different packets could be received in different order. In different order.

In different order. Uh, because some paths can be slower, they had some congestion, right? Yeah. So, so the, in this way, um, some packets sent earlier can arrive later. And what's the problem of reordering?

Yes. You need larger buffers, right?

You need larger buffers, right? Yeah. Yeah. Larger buffers. And what, what, what, what, what else? Well, TCP might actually, if it's TCP, when your session is on. They use TCP and TCP would definitely be a problem if you have packets that are reordered. Why? Because TCP has a mix name. Yes. You want to say it? Oh, I was going to say like, TCP has this thing where it's like, if it keeps getting the correct order of packets, it can like streamline the whole process. Yeah. It's called a triple duplicate act. If it has a triple duplicate at a TCP, we know we will reduce the congestion window to two. I mean, this is, this should be an undergrad stuff, but I didn't, I didn't have the time to review it, but TCP performance will be significantly hurt when you have out of order packets. So we want to make the packets stick on the same path. All right. So we don't have time. We should stop here and see you on Thursday. At least when you run a same flow, like you are downloading something, I would assume the password is the same. Okay.