**جلسه اول:**

**جزوه اول:**

انتقال داده ها باید به صورت فیزیکی در قالب موج ارسال و دریافت بشود مثل صدا یا هر چیزی دیگری مثل پیوسته در زمان یا گسسته در زمان (رقمی شده دیجیتال). انتقال شامل رسانه های هدایت شده یا با سیم هستند مثل زوج سیم های به هم تابیده و فیبر نوری و ... ، و رسانه های غیر هدایت شده یا بی سیم. مثل هوا یا آب و تو خلاء منتشر میشن. داده ها دو دسته هستند پیوسته در زمان و گسسته در زمان.

امواج رادیویی بهترین موج برای انتقال پیام هستند نور فقط مستقیم میرود. تا 1 ساعت اول دیده شد.

امواج رادیویی نویز کمتری دارد و طول موج های بزرگتری دارد، چون امواج رادیویی را گرفته بودند ما مجبور شدیم از امواج مایکرویو زمینی استفاده بکنیم.

ما هاست ها و end system ها که شامل هر چیزی هستند مثل لپ تاپ و ... را به هم میخواهیم وصل کنیم با چه وسیله ای توی اقیانوس؟ با اون interface ها یا واسط ها مثل روتر ها سوییچ ها تکرار کننده ها، هاب ها و پل ها، روتر ها همان مسیر یاب ها هستند. gateway ها همه لایه ها را درگیر میکنند هاب ها و تکرار کننده ها فقط برای لایه فیزیکی هستند پل و سوییچ برای لایه دوم هستند و روتر ها برای لایه سوم هستند و gateway ها که پیچیده تر هستند همه لایه ها را درگیر میکنند.

اگر شبکه مثل گراف باشد گره ها یا node ها میشوند واسط ها و لبه ها میشوند لینک های ارتباطی که همان رسانه های انتقال داده هستند مثل سیم ها و ... ، مدارات داخل کارت شبکه هم جز لینک ها هستند. با این گراف یک زیر شبکه ارتباطی میسازیم تا همه هاست را به هم متصل کنیم.

تقسیم بندی شبکه از دید همبندی یا topology: میان گراف ها رو بررسی میکنند که چه شکل هایی میسازد دو نوع هم داریم نقطه به نقطه یا point to point و broadcast یا همه بخشی، تو اولی به طور مستقیم از سمت مبدا به سمت ماشین مقصد هدایت میشود. در مقابل رادیو یا broadcast هست چون داده را به همه ماشین ها پخش میکند مثل رادیو، مثلا همه ماشین ها رو اون فرکانس خاص میگیرند، شبکه های با سیم و بی سیم از همه بخشی یا broadcast استفاده میکنند. تو همه بخشی به همه میگیم ولی با یک نفر کار داریم ولی نقطه به نقطه فقط به یک نفر میفرستیم و میشنود. نقطه به نقطه با همتا به همتا فرق میکند دومی یک معماری سیستمی هست اولی یک توپولوژی هست. معماری همتا به همتا یا peer to peer دیگه مثل client/server نیست بلکه همه با هم برابر هستند. از این 2 تا معماری استفاده میکنیم.

خود نقطه به نقطه ها نوع های مختلفی دارند، ستاره: یکی وسط بقیه بهش وصل هستند. مش کامل: همه به هم وصل هستند. درخت: مثل درخت هست، مش معمولی: همه به هم وصل نیستند. دقت کن توی همه بخشی همه میشوند ولی چون ما فقط یکی کار داریم فقط اون برمیدارد. توپولوژی مانع از انتقال داده به شکل تک بخشی چند بخشی و همه بخشی نمیشود. انتقال داده هم همه بخشی آدرس دارد. در استاندارد اترنت وقتی آدرس مقصد 6 تا ff بگذاریم یعنی همه بخشی یعنی واسه همه میفرستیم.

جلسه دوم:

اینترنت یک شبکه واحد و یا یک استاندارد یکتا نیست بلکه مجموعه ای از زیر شبکه های مختلف و استاندارد های مختلف هست. زیر ساخت شبکه میشود لایه فیزیکی و روی آن لایه دیتا لینک یا پیوند داده هست. لایه های فیزیکی کلا با هم فرق میکند با استاندارد های گوناگون با سیم و بی سیم و ... .

وب همان اینترنت نیست، وب یک سیستم توزیع شده هست. یک سیستم توزیع شده یعنی ، اینترنت با پروتکول های IP و TCP به وجود اومد و بعد UDP، سه کاربرد مهم داشتند FDP برای انتقال فایل، Ethernet برای لاگین به سیستم راه دور و ESMDP برای دریافت و ارسال ایمیل اومد. حالا چرا وب اومد؟ شبکه پیچیده شد و انواع پروتکول، سیستم ها و ... داشتیم. مشکل پیچیدگی هست. مشکل معماری های ناهمگن و زیاد هست. گوناگونی سیستم عامل ها هست. و مشکلات خرابی ها، برخورد ها، رقابت ها مثل همون که تو سیستم عامل داشتیم. و سازگاری تمام نسخه ها از یک داده که مثلا وقتی یکی را تغییر میدهیم بقیه هم تغییر کنند. پس رفتن دنبال وب چون ساده بود و همه چیز یک جا بود یعنی به نوعی یک وحدت بود یعنی نحوه بیان داده یکسان هست. و واسه همین دنبال سیستم های توزیع شده هستیم ولی ما این پیچیدگی و زیاد بودن را میپذیریم. یعنی برای کاربر ها یک کامپیوتر واحد به نظر میاد یک تصویر تک سیستمی ایجاد میکنیم ولی در واقعیت انواع و کثرت سیستم هست. و فکر میکنه همه چی تو لپ تاپ شما هست و فقط اسم رو بگو و همه پیچیدگی ها از دید کاربر پنهان هست و یک تصویر تک سیستمی ایجاد میشود تا سادگی ایجاد شود یعنی حتی ip ها از دید کاربر مخفی هست. این شاهکار سیستم های توزیع شده هست. این ویژگی را شفافیت میگوییم. یا نامرئی سازی پیچیدگی های حاصل از فراوانی هست در واقع شفافیت به این معنا هست. یعنی مثلا میگوییم google ما دیگه اصلا نمیدانیم ipش چی هست چه کامپیوتری هست چه سیستم عاملی هست تو کدوم کشور هست ما فقط یک نام میگیم و به بقیه کاری نداریم. وب یک نمونه از سیستم های توزیع شده هست. یک نوع دیگر مثل cloud computing هست یا مثل دیتابیس management سیستم هست این ها همه جز سیستم های توزیع شده هستند بلکه به صورت موازی پردازش داشته باشند و بار سیستم را پایین بیاورند. وب یک سری اسناد با یک استاندارد واحد هست روی هر سیستمی، مثل HTML، و از طریق نام URL و این ها مثل تار عنکبوت همه این ها به هم پیوند خورده و هایپر لینک هستند. چطوری نام به یک IP تبدیل میشود؟ با DNS سرورای دنس نام میگیرند و آدرس میدهند. یعنی مجموعه سرور های DNS نام را که گرفتند یک آدرس به ما میدهند. کی این کار را انجام میدهد؟ یک زمانی میگفتند سیستم عامل بیاد این کار ها را انجام بدهد در واقع ما به ازای هر معماری یک کرنل واحد بسازیم مثلا 50 تا معماری و 50 تا کرنل واحد که اون خودش برود همین کار دنس را بکند ولی این کار خیلی زیاد و بزرگی بود یعنی 50 تا سیستم عامل مختلف رو 50 تا معماری مختلف هر کدوم واحد باشند. چی جا این را گرفت؟ middleware یا میان افزار یعنی ما میپذیریم انواع سیستم عامل ها را و میپذیریم معماری های مختلف را، و روی همه این سیستم عامل یک لایه نازک میگذاریم به نام میان افزار که فقط پنهان کند پیچیدگی ها را و یکپارچه نشان بدهد و میان افزار زیر اپلیکیشن و روی کرنل قرار میگرفت. و روی یک محیط ناهمگن بشیند. و به این میگویند disrepute system. اپاچی سرور وب هست. نگاه به اینترنت از دو منظر مختلف هست. 1. توصیف اجزای اصلی اینترنت شامل سخت افزاری مثل دروازه ها و روتر ها و ... ، و نرم افزاری. 2. به عنوان یک شبکه زیر ساخت که سرویس های مختلفی را برای برنامه های کاربردی توزیع شده فراهم می آورد و چطوری بتوانیم سرویس بگیریم و ازش استفاده کنیم مثل socket programming.

نگاه از بالا به اجزای اصلی اینترنت: یکی از اجزا ماشین هایی هستند که کاربران باهاش به اینترنت متصل میشوند. و بخش دیگر ماشین ها و هاست ها هستند که قرار هست به اینترنت متصل شوند. به همه اینها میزبان میگوییم چه هاست باشد داخل یک دیتا سنتر، این هاست میزبان یک برنامه وب سرور مثل دیجی کالا هست یعنی مثلا اپاچی میخواهیم وصل شویم و ما با برنامه کروم هستیم و اون میزبان ما هست و یک end system هست نگاه اقیانوس و ساحل به سیستم های انتهایی داشته باش.

لینک ها پیوند های ما همان کارت های شبکه هستند که روی دو میزبان قرار میگیرند و انواع گوناگونی از لینک های ارتباطی داریم و مثلا از یک media فیزیکی متفاوت و انواع پروتکول ها و قوانین متفاوت.

رسانه های گوناگونی مثل کابل کواکسیال یا فیبر نوری و طیف رادیویی که میتوانند لینک های متفاوت داده ها را با نرخ انتقال های متفاوت و واحد انتقال بیت بر ثانیه را میتوانند انتقال بدهند.

FDP سرور و MySQL سرور به این ها برنامه میگوییم به اون ماشین هایی که توی رک ها هستند میگوییم و توی دیتاسنتر ها هستند و در واقع اون برنامه ها روی این ماشین ها اجرا میشوند. روتر ها وظیفه مسیر یابی و وظایف اصلی هسته اینترنت را در لایه سوم هستند و به آنها سوییچ لایه سوم میگوییم. سوییچ های لایه پیوند یا سوییچ لایه 2 هستند و ساده تر از لایه 3 هستند. در واقع به این شکل هست که ما از طریق لپ تاپ به یک نقطه دسترسی یا access point وصل میشویم که خود آن به سوییچ ها وصل هستند و این سوییچ اترنت ها به روتر وصل میشوند و بعد به اینترنت متصل میشوند. پروتکول مثلا LAN هست که با یک سیم به اون access point وصل میشود. Video stream به صورت جریانی از بسته ها میاد برای شما میفرستد و تو لایه انتقال TCP میاد این تکه به تکه میکند و به لایه ها زیر میفرستد مثلا IP به این لایه میفرستد. و با یک هدر میفرستد که این هم TCP نوشته مثلا این قسمت چندمین قطعه از اون هزار تا قطعه هست و به کجا باید برود بعد IP هم میاد یک هدر مینویسد. و وقتی به مقصد رسیدند و از اون هدر استفاده میشود تا بتوانیم از داده ها استفاده کنیم و بعد میندازیم دور و در واقع توسط اون هدر داده ما دوباره بسته بندی میشود. حالا برویم سراغ هسته، اگر سوییچ های بسته تا لایه نتورک را ساپورت کنند بهشون روتر میگوییم یا سوییچ لایه سوم را میگویند و اگر فقط لایه 1 و 2 را ساپورت کنند بهشون میگیم سوییچ لایه پیوند که همون سوییچ میگوییم و در لایه سوم به سوییچ روتر میگوییم.

سوییچ های لایه پیوند در سطح access point یا access network قرار دارند و ما را به اولین روتر میرسانند. مثلا به اولین روتر ISP، و روتر ها در هسته اینترنت استفاده میشوند. و توسط لینک های ارتباطی و سوییچ های بسته که توسط سیستم های انتهایی طی میشوند تا به مقصد برسند و البته یک سری الگوریتم داریم که بهترین مسیر را پیدا کند به همه این مسیر یک route میگوییم. سوییچ های بسته در واقع همون تقاطع های چهار راه هستند. ISP ها در سطح شهری و کشوری و قاره ای هستند و اون غول ها کارشون این هست که ISP های کشوری را به هم بچسبانند. و سوییچ های آن ها در همه جا پخش شده اند حالا چه کشور باشد چه شهر. دقت کن سرور ها هم باید به اینترنت متصل شوند و اینترنت کلا چیزی نیست جز اینکه همه میزبان ها و سیستم های انتهایی را بهم وصل کند. ISP ها اینترنت را برای شما تهیه میکنند و مبین نت میاد مثلا در سطح کشوری عمل میکند و رده بندی دارند.

یا به ISP های بزرگ تر وصل میشوند و یا به CDN ها وصل میشوند.

سیستم های باز: اول سیستم های بسته بودند مثلا چند تا کامپیوتر یا چند تا شرکت رو مثلا IBM وصل میکرد و بسته بود و مثلا SON نمیتواند به IBM وصل شود چون همه چیز متفاوت هست بنابراین برای اینکه این ها بتوانند با هم ارتباط برقرار کنند باید به سیستم های باز وصل شیم، و پروتکول های منطقی و استاندارد و باز باید داشته باشیم تا بتوانیم با بقیه ارتباط برقرار کنند. باید قواعد استاندارد داشته باشد، چه کسی این کار را انجام میدهد؟ شرکت های مختلف به نام حامل های عمومی مثلا در ایران مخابرات سیم کشی این کار را انجام میدهند. کار مودم این هست که سیگنال دیجیتال را به سیگنال آنالوگ و سیگنال آنالوگ را به سیگنال دیجیتال یا گسسته در زمان تبدیل میکند. و همه این تجهیزات باید استاندارد داشته باشند تا با هم بتوانند کار بکنند.

تا 1:40 دیده شد.

شرکت های بزرگ دنیا میان این استاندارد ها را ایجاد میکنند، بعد اومدند شبکه های دیجیتالی را پدید آوردند یا PSDN بدون نیاز به مودم جا به جا میشد، برای انتقال داده های دیجیتالی، بعد از آن ISDN اومدند یعنی تلفن با دیجیتال یکی شود و صدا را هم دیجیتال کردند و از یک خط صدا و داده بدون نیاز به مودم ارسال میشد. ترکیبی از PSDN بود، پل واسه وقتی هست که 2 تا زیرساخت متفاوت را بخواهی به هم وصل کنی خود این پل ها توسط ISDN به هم وصل شده اند و یک ترکیبی از PSDN تلفنی و PSDN داده ای را در یک خط منتقل کنند بعدا گفتند هر نوع media را بتوانیم انتقال بدهیم، مثل فیلم و عکس، براد بند یعنی جاده وسیع میشود و به جای داده میتوانیم فیلم هم بفرستید یعنی اونجا جا بزرگتر شد. پهنای باند آنالوگ هست و اون نرخ انتقال یا transmission rate هست که برای سرعت اینترنت دیجیتالی گفته میشود، البته این 2 با هم نسبت مستقیم دارند. یعنی اگر W را دو برابر کنی R هم دو برابر میشود یعنی پهنای باند را دو برابر کنی نرخ انتقال هم 2 برابر میشود، IETF جز مهم ترین شرکت ها هست و انواع پروتکول ها را این ها درست کرده اند و IEEE هم تجهیزات را استاندارد کرده اند. IP یک پروتکول در لایه سوم هستند IP دو کار میکند یکی addressing و دیگری framing دومی یعنی فرمت هدر اون بسته های یک ویدیو را اون IP این کار را انجام میدهد و اولی هم کار IP هست، استاندارد ها باید دقیق و کامل باشند و بی طرف باشد، استاندارد های IEEE به صورت توپولوژی همه بخشی بودند و تو لایه هایی پایین سخت افزار و مدارات الکتریکی و اتصالات بی سیم را استاندارد میکرد.

اپلیکیشن ها و کلاینت ها و سرور ها فقط روی هاست ها نشسته میشوند و روی هسته و سوییچ ها قرار نمیگیرند. اپلیکیشن ها اصلا سوییچ ها را نمیشناسند، اپلیکیشن ها همون سیستم های انتهایی هم میتوان گفت، اپلیکیشن ها API ها را صدا میزنند و آنها فراخوانی سیستم میکنند و در لایه های پایین تر به کارت شبکه میرسد و اجرا میشود.

QOS یا کیفیت خدمات تضمین شده یعنی اون سرویس ها را تضمین کنیم مثلا از یک سرعتی کمتر نشه؟ JITTER از یک مقداری بیشتری نشه؟ و اگه این ها را توانست ارائه بدهد میگوییم این تضمینات را دارد.

پروتکول چیست؟ ما میخواهیم سیستم های باز را استاندارد کنیم اون مجموعه قوانین استاندارد میشود پروتکول، یعنی چه پیامی باید ارسال و در جواب این پیام چه چیزی باید ارسال شود یا پاسخ داده شود، مثلا وقتی ما میزنیم گوگل پروتکول TCP میرود به سرور درخواست میدهد چرا چون وب ما HTTP پروتکولش هست و در لایه زیرش روی TCP مینشیند، TCP نیاز دارد اتصال برقرار کند، بعد تایید میده و یک پیام GET میاد توسط TCP ارسال میشود، و یک فایلی میاد که مثلا عکس و نوشته ها میاد. پروتکول میتواند مثلا واسه دو تا کارت شبکه باشد و پیام منتقل میکنند، پس پروتکول سخت افزاری و پروتکول های نرم افزاری داریم. پروتکول های مسیریابی هم داریم در واقع در همه بخش ها هستند.

**پایان جزوه دوم.**

**جزوه سوم:**

ویدیو سوم:

لبه شبکه یا اینترنت: در لبه شبکه هم کلاینت ها هم سرور ها در لبه شبکه قرار دارند، سرور ها همان سرویس دهنده ها هستند هر چیزی مثل وب سرور یا ایمیل سرور و ... ، همه در لبه شبکه یا لبه اینترنت قرار دارند، چرا میزبان؟ چون اینها میزبان یا هاست اجرا کننده برنامه مشتری هستند مثلا مرور گر ما برنامه مشتری هست، مثل کروم، یا مشتری پست الکترونیک یا برنامه های سرور مثل اپاچی یا وب سرور و ...، اینا نمونه ای از وب سرویس ها هستند، و همه اینها میزبان هستند، وقتی میگیم برنامه تاکید ما مثلا MySQL server sql server Microsoft server هستند یا مرورگر ها کلاینت هستند، البته ما در این درس به اون ماشین هایی که اینها را اجرا میکنند هم کلاینت میگوییم، مثلا میتوانیم بگوییم سرور ما میزبان وب سروری مثل دیجی کالا هست، در اون شکل بیضی ها روتر یا مسیر یاب ها هستند و روتر به هسته شبکه وصل هست در واقع روتر ها هسته شبکه هستند، چهارگوش ها سوییچ لایه دو یا ethernet هستند که access point یا access network هم گفته میشود، به اولین هسته شبکه edge router نیز گفته میشود، در کامپیوتر ما مرور گر ما میزبان آن هستیم و اینور سرور میزبان صفحات وب و میزبان اپاچی و سیستم سرویس دهنده هست. تجهیزات لایه network روتر ها هستند که routing یا forwarding و switching که بسته ها به سمت مقصد هدایت میشوند برای اینکه از بهترین مسیر بروند توسط الگوریتم routing انتخاب میشود که در فصل چهارم میخوانیم.

حالا سراغ access network ها میرویم و معرفی میکنیم، دقت کن access point میتواند بی سیم مثل وای فای باشد یا با سیم مثل Ethernet، و این ها با براد بند ها مثل ADSL به ISP وصل میشوند یا مثلا P2P یا FFTP، همه این براد بند ها به صورت سیم نیست مثل TDLT یا P2P. اگر در خانه باشیم معمولا ADSP کار میکنند یا HFC یا FTTH که جدید هست یا dial-up و یا ماهواره که برای تو کوه و دشت اینا هستند، پس پنج سرویس داریم مثل DSL، HFC تا سر کوچه فیبر نوری هست و تا تو خونه میاد میشه کواکسیال، FTTH یا فیبر نوری تا تو خونه میاد. خطوط ADSL از همان زوج سیم های به هم تابیده که واسه تلفن بوده استفاده میکند دیگه دوباره سیم کشی نمیکند، چه مشکلی با DIAL UP بود؟ اینترنت میومد از اون پهنای باند صفر تا 4 کیلو هرتز به صورت تمام استفاده میکرد و واسه همین تلفن قطع میشد یا تلفن بود یا داده روی اون خطوط بعدا دیدن اون سیم های به هم پیچیده هرتز های خیلی بیشتری دارن که استفاده نمیشن بنابراین اومدن تقسیم بندی کردن یه مقدار واسه صدا یه مقدار واسه ارسال و یه مقدار واسه دریافت اینترنت. چرا down stream؟ چون بیشترش رو واسه دریافت گذاشتیم تا ارسال، و ISP ها میان روتر های خودشون درون دفتر های مخابرات تلفن میگذارند، و اینجا مالتی پلکسر میخواهیم چون 100 تا خط از خونه مشتری ها میاد و یک خط به ISP با سرعت چند گیگ بر ثانیه میفرستد، کامپیوتر ما با شبکه دسترسی به DSLAM وصل میشود و DSLAM ما را به روتر های لبه میفرستد و توسط سوییچ ها و روتر ها به هر جایی که خواستیم میرویم.

در واقع یک لینک تلفن به 3 خط مجزا تبدیل شد و اون خط چند گیگ بر ثانیه رو مالتی پلکس کردیم به 100 ها خطوط واسه مشتری ها، که بهش تقسیم فرکانسی میگوییم، همینکه مثلا بین هرتز ها تقسیم فرکانسی کردیم بعد این میتواند متقارن یا نا متقارن باشد، splitter وظیفه دارد که down stream و up stream و صدا را در خانه مشتری از هم جدا کند، در سمت مخابرات هم وظیفه DSLAM هست که به کدوم سوییچ ها منتقل کند، کی باید این ها را استاندارد کند؟ شرکت های tele communication اومدن استاندارد کردند.

هر چی فاصله دورتر شود سیگنال ضعیف تر میشود و نسبت به نویز تاثیر پذیر تر میشود در واقع signal to noise ration طبق قاعده شارنون کمتر میشود. و از یک پهنای باند مشخص بیت کمتری میتوانیم برداشت کنیم.

تا دقیقه 54 دیده شد.

گره فیبر: از اون مرکز تا سر خونه ما هست با فیبر نوری بعد تا در خونه کابل کواکسیال هست و داخل خونه هم مودم کابلی هست. DSLAM و CMTS میان سیگنال دیجیتالی رو به انالوگ و برعکس تبدیل میکنن مثلا سیگنال انالوگ را میگیرن و به دیجیتال تبدیل میکنند و میفرستن تو خونه. تصادم یعنی اینکه روی این سیم ها کواکسیال که ما فرستادیم ممکنه همزمان هم ما بفرستیم هم همسایه پس دیتا قاطی میشه که و تصادم میشه که پس ما باید جلوی آن را بگیریم و تو فصل 5 میخوانیم.

تکنولوژی جدید: فیبر تا خانه هست، و اینجا هم 2 حالت دارد یا یک فیبر میاد و مثلا برای 40 خونه به 40 انشعاب کشیده میشود و یا 40 تا فیبر میاد و برای هر خونه یک فیبر اگر اشتراکی باشد یا حالت 1، دوباره همان مشکلات تصادم را داریم، ما یا فیبر مستقیم داریم یا شبکه توزیع نوری یا غیر مستقیم به نوعی direct و indirect، و توسط یک splitter مثلا اون یک فیبر را میگیرد و مثلا به 40 تا تقسیم میکند مثل سوییچ ها نیست که بداند این بسته واسه کی هست و روی هر 40 تا خط هر بسته را تقسیم میکند و میفرستد.

Indirect: دو نوع هستند فعال و غیر فعال بودند، هاب مثلا یک نوع دستگاه برای شبکه نوری غیر فعال یا منفعل هست یعنی همه به آن وصل میشوند و کار خاصی نمیکنیم و ممکن هم هست تصادم رخ دهد، ساده هست. در فعال سعی میکنیم مثلا بافر بگذاریم یا از اون تصادم جلوگیری کنیم یا جدا سازی کنیم یا از سوییچ ها استفاده کنیم که مثلا فقط به یک خونه اشاره کند. اون فیبر پایانه ای که در خونه شماست میشود ONT و اونکه تو اون مرکز مخابراتی هست OLT نام دارد.

سیگنال هایی که از طریق روتر لبه به هسته یا سیگنال های کامپیوتری ما همه الکتریکی هستند ولی سیگنالی که روی خطوط هستند نوری هستند و این وظیفه OLT و ONT هست.

در شرکت ها معمولا از شبکه محلی استفاده میکنند، و سوییچ های اینترنت باشعور جایگزین هاب های بی شعور شده اند که تو فصل 5 بیشتر میخوانیم. و از طریق LAN به هم وصل هستند و سوییچ در نهایت به روتر وصل شده است که اصلا یک ISP واسه خودشون دارند، مثلا ممکنه چندین روتر داشته باشیم و اونها به چندین سوییچ وصل شده باشند، میتوانند به صورت بی سیم هم باشند و یک نقطه دسترسی داشته باشیم و حتما باید base station و access point دارد و بعد از اون نقطه دسترسی به شبکه وای فای وصل میشوند و اونجا وای فای به روتر وصل میشوند.

اون 802.11 واسه وای فای و 802.3 هم واسه اترنت هستند. حفظ کن.

3g و ... برای شبکه بی سیم شهری هستند. تا الان هشت تا access network گفتیم: ADSL, FFTH, HFC، AON,PON، مثل اترنت و وای فای LAN، شبکه های شهری، ماهواره، DIAL UP.

دقت کن طیف رادیویی مایکرویو زمینی استفاده میکنیم تو فرکانس های بالاتر، ما زوج های فرستنده-گیرنده داریم یعنی هعی یکی میگیرد از اونطرف میفرستد بعد روتر بعدی از اینطرف میگیرد از اونطرف میفرستد. و پالس های نوری از رسانه فیزیکی استفاده میکنند.

چرا زوج سیم مسی را بهم میتابند؟ چون اگر یک نویزی بیاد رو یکی بخورد روی اون یکی هم میخورد و اثر آن از بین میرود یا کاهش میدهد چون هم این نویز میگیره هم اون یکی. هر سطح فلزی بسته نویز الکترو مغناطیس وارد نمیشوند، STP واسه وقتی هست که نویز زیاد هست مثلا از کنار دکل برق میکشیم یک لایه فلزی روش میکشند به ما میدهند و طبق اون قانون برق داخل اون نویز الکترو مغناطیس نمیاد.

کابل کواکسیال یا هم محور چون دو تا استوانه هم محور دارند بهش میگویند هم محور و حالت محافظ عمل میکند اون استوانه داخلی همان مغزی هست. جدیدا استوانه ها را رشته به رشته میسازند و عایق هم دارند و یک پوشش پلاستیکی دارند و BIT RATE بالایی دارند.

**پایان ویدیو سوم.**

پایان جزوه سوم.

**شروع جزوه چهارم:**

شروع ویدیو چهارم:

وقتی ازدحام شود در شبکه، میتواند باعث تاخیر شود، بحث پهنای باند هست. چون ما بیت ریت محدود داریم و محدودیت داریم مثل سرعت پردازش و ... ، و ما باید خطا و تلفات را قبول کنیم. ما باید تضمین کنیم و گرنه مثل ip هستیم که میشود best effort یعنی بهترین تلاش خود را میکنیم.

معیار های کیفیت سرویس: نوسان تاخیر: یعنی بیشترین میزان تاخیر منهای کمترین میزان تاخیر که همان جیتر هست لرزش یا نوسان تاخیر میزان بازی تاخیر هست. جیتر میتواند صفر باشد ولی تاخیر نمیتواند صفر باشد جیتر صفر یعنی نوسان تاخیر صفر باشد. نرخ یعنی در واحد زمان مثلا بیت ریت یعنی چند واحد در زمان میتوانی بیت بفرستی نسبت کسر هست یعنی اصلا واحد ندارد مثلا 100 تا بسته میفرستیم یکی تلف میشود پس میشود 1%. گذردهی چند بیت مفید در واحد زمان به مقصد میرسد. قابلیت اطمینان یعنی در یک واحد زمان سیستم درست کار کند و بسته های ارسالی دچار خطا نشوند و لینک ها قطع نشوند و ...، این 5 معیار کیفیت سرویس بودند.

تاخیر: انواع تاخیر داریم بسته به اینکه چه نوعی هست واحد آن ثانیه هست. مثلا تاخیر انتها به انتها.

نوسان تاخیر یا لرزش: که همان جیتر هست، مثلا اگر تاخیر یکسان باشد باید بسته ای که میفرستیم مثلا با 1 ثانیه تاخیر میرسد بسته دوم هم با همون ثانیه تاخیر برسد که در واقعیت به شکل نیست. ما تاخیر را نمیتوانیم صفر کنیم ولی جیتر را میشود کم کرد. همان کاری که یوتیوب و نتفلیکس میکند اول آن چند ثانیه صبر میکند و یک بافر میگذارد و بعد پخش میکند چون اول هست کاربر را اذیت نمیکند. پس یک بافر در گیرنده میگذاریم و چه مدت بافر کنیم؟ دقیقا به مدت جیتر.

اکثرا باید بایت را به بیت تبدیل کنیم. تاخیر قابل جبران نیست جیتر قابل جبران هست و تلفات هم قابل جبران هست مثلا شماره گذاری کنیم. و میگیم این رسید ولی مثلا 3 نرسید چون بر اساس شماره ها میفهمیم کدوم ها رسیدن.

گذردهی: چند بیت در واحد زمان مفید میرسد به مقصد، همه بیت ها مفید نیستند غیر مفید ها میشوند هدر ها یا بیت ها و پارامتر های کنترلی، مثلا سیگنال های کنترلی لازم هستند ولی مفید نیستند مثلا بگیم چه داده ای میخوای بفرستی یا بگیم داده رسید.

واحد آن بیت در ثانیه هست، ما معمولا میانگین گذردهی را کار داریم. فرمول: سایز فایل تقسیم بر زمان. زمان همان تاخیر هست. بیشتر از این فرمول استفاده خواهیم کرد یک رابطه ای دیگم هست که بهره وری ضربدر گذردهی که اون واسه یک جا های دیگه کاربرد دارد.

گذردهی بخش مفید بیت ریت هست. r با w رابطه مستقیم دارد.

قابلیت اطمینان: شبکه مطمئن شبکه ای هست که خطا نداشته باشد یا اگر داشت بتواند آن را اصلاح کند. خطا نداشتن یعنی همه بیت ها سالم برسند و بدون تکرار برسند و به ترتیب برسند. تکرار خطرناک هست همچنین به هم خوردن خطرناک هست.

نکته مهم این هست که در برنامه ما شاید همه این معیار ها برای ما مهم نباشد مثلا فقط تاخیر مهم باشد پس ما یک میزان حساسیت کاربرد ها به معیار ها را داریم پس دو نوع داریم حساس به تاخیر هستیم یا تحمل پذیر تاخیر هستیم. Tolerant به اصطلاح.

عوامل تاخیر: نکته خارج از بحث لینک ها edge یا لبه هستند و سوییچ ها گره یا نود هستند. در هدر آدرس مقصد هست و اون روتر میفهمد به کدوم روتر باید هدایت کند تا مثلا برسد کرج و روتر چک میکند که خطا دارد یا نه جدول مسیر یابی هم داریم به این کار ها میگوییم پردازش. حالا اومد داخل پردازنده بعد باید تو صف وایستد تا نوبتش شود به این میگوییم تاخیر صف که بسته به اینکه ترافیک سنگین یا سبک هست. تاخیر انتقال این هست که مثلا 1000 تا توپ داریم و هر دقیقه 1000 توپ میتوانیم بفرستیم این تاخیر ما 1 دقیقه هست ولی مثلا اگر 500 توپ در دقیقه بتوانیم بفرستیم دو دقیقه تاخیر داریم این فقط زمانی هست که من دارم میگذارم روی صف و لینک. تاخیر انتشار این هست که مثلا توی هوا هست یا روی سیم هست و اون بسته ما media را طی میکند و به مسیر یاب بعدی برسد به این میگوییم تاخیر انتشار.

تا 1:30 دیده شد.

خیلی وقت ها برای تبدیل بایت به بیت یک ضربدر هشت نیاز داریم دقت کنید.

**جزوه پنجم:**

قسمت پنجم:

Store and forward: بسته میاد به هدرش نگاه میکنیم مثلا میگه تهران به جدول نگاه میکنیم تهران باید بره به فلان روتر forwardش میکنیم به خروجی اون روتر. بر اساس ادرس مقصد میگیم مثلا فلان پورت برو. Store یک بسته که رسید 300 بیت مثلا رسیده هست اجازه ندارد دونه به دونه بفرستیم به پورت خروجی چون باید تمام بسته برسد در شبکه های پکت سوییچ تا بتواند بفرستد.

چون اخرین ماشین دقیقه دوم ارسال میشود و وقتی برسد 60 دقیقه دیگر میرسد و 62 دقیقه طول میکشد. تاخیر انتقال با تاخیر انتشار جمع میشود.

تاخیر کل در این جزوه برای یک بسته هست و برای چند تا بسته فرق میکند. دقت کن توی فرمول تاخیر صف باید با سیگما بری و مقدارش برای 9 تا بسته هست. اگر 10 تا بسته بود که اخرین بسته جلوش 9 تا بسته هستند که دارن ارسال میشن یعنی تاخیر انتقال همه رو با هم جمع بزنی و به قطعی بدست میاد ولی تو سوالات کنکور و دنیای واقعی تصادفی میدهند که به این سادگی نتوانی جواب را بدست بیاوری.

در صف ها اولین m برای نشان دهنده این است که ماهیت بسته ها چیست مثلا اگر d گذاشتند یعنی قطعی هست داده ها، ولی m میگذاریم تصادفی هست و بدون حافظه هست یعنی اتفاقات آینده ارتباطی به گذشته ندارد. چرا گسسته پیوسته کردیم ؟ اگر راجب ریت بسته ها صحبت کنیم با گسسته کار داریم اما اگر راجب فاصله زمانی بخواهیم صحبت کنیم چون عدد حقیقی هست میشود پیوسته. حالا زیاد این قسمت ها مهم نیست. تابع توزیع نمایی مقدار آن عکس پواسون هست. از نظر میانگین. پس m اول برای ورود بسته به اون صف پس اگر ریت نگاه میکنی پواسون نگاه کن. نرخ عکس زمان هست. همین رو فقط یادت باشد. دومین حرف m نماد زمان سرویس یا زمان انتقال هست. همون زمانی هست که داریم به اون بسته سرویس میدیم و براش وقت میگذاریم. میانگین نرخ ورود را لاندا گرفتیم و میانگین زمان سرویس را با میو نشان میدهیم. که این 2 باید بی حافظه و مایکروین باشند. اون 1 به چه معناست؟ نشان دهنده تعداد سرور ها هست یعنی فقط یک گیشه در بانک کار میکند. البته در بانک مثلا m m 3 هست چون 3 تا سرویس دهنده موازی داریم که از یک نوع هستند. اون بینهایت هم ظرفیت صف هست که معمولا بینهایت هست یا در نظر نمیگیریم روش نوبت بندی هم مثلا FCFS هست. به صورت دیفالت همین صف M M 1 بگیر. ظرفیت بی نهایت ظرفیت جمعیت بی نهایت روش نوبت بندی هم که گفتیم دقت کن مایکروین باشند اون دو تا M بعد فقط فرمولش را کار داشته باش. لاندا زمان وارد صف میشوند بسته ها و یو مقدار بسته ها ارسال میشوند و یو باید از لاندا بزرگتر باشد و گرنه صف میترکد و شلوغ میشود. معمولا لاندا را میدهند ولی میو را نمیدهند تاخیر انتقال را میدهند که طول بسته بر روی نرخ انتقال بود و میو را باید بدست بیاوری پس دقت کن.

پس میو میتواند r/l ام بشود یعنی نرخ انتقال بر روی طول بسته .

شدت ترافیک: نسبت لاندا به میو هست. یعنی مثلا 9 مریض در ساعت بیاد ولی 10 تا مریض در ساعت سرویس بده میشود 9 دهم یا 90 درصد که خوب هست. حالا اینجا هم ممکن هست میو را ندهند. و فقط نرخ انتقال را بدهند که باید حساب کنی. در شبکه نرخ بیت بر ثانیه دادند میتوانیم تقسیم کنیم اگر بسته بر ثانیه هم دادند میتوانیم تقسیم کنیم پس فرقی ندارد. شدت ترافیک باید زیر 1 باشد. حتی به 1 هم نباید نزدیک شود چون 1 شود بی نهایت میشود. چرا ؟ چون وقتی شدت ترافیک به 1 میل کند مجانب باشد یعنی لاندا دارد به میو نزدیک میشود و وقتی نزدیک شود تاخیر صف مخرجش که منها این 2 بود صفر میشود و میشود 1 بر روی صفر که میشود بینهایت.

میانگین تعداد بسته های درون صف: لاندا ضربدر تاخیر صف. بدیهی هست چون هر کدوم یه تاخیری دارن با یه انتقالی هم دارن میان چند نفرن خوب ضرب اینا دیگه.

اون جمع تاخیر فقط برای یک بسته هست و اگر چند تا بسته بود نباید ضربدر چند تا بسته کنی چون اشتباه هست.

تاخیر رفت و برگشت: همان پینگ هست که میخوان ببینن کسی هست یا نیست و 2 تا تاخیر انتها به انتها دارد چون یه بار میره یه بار میاد دیگه. البته همیشه 2 برابر نیست یک تخمین هست. همان سلام و علیک سلام را فکر کن. RTT نام دارد.

نسبت تلفات و احتمال خطای انسانی: تلفات همون پکت لاست هست. نسبت تلفات و احتمال خطا پارامتر ضربی هستند چون هر پیشامد ما مستقل هست. پس نسبت تلفات به این شکل هست: احتمال اینکه تو گام اول لاست نشود؟ 1 منهای لاست. ضربدر احتمال اینکه در گام دوم لاست بشود؟ 1 منهای لاست. و .... تا اخر. چرا منفی هست؟ چون احتمال اینکه گام اول لاست نشه و گام دوم لاست نشه و گام سوم لاست نشه در شبکه واسه همین منفی میکنیم چون بگیم گم نشه دیگه پس اگر تو هیچ گامی رخ نداده هست انگار گم نشده هست و چون این گم شدن در هر گام مستقل هست باید ضرب کنیم. عین همین برای احتمال خطای بسته ها هست.

تاخیر تصادم هم برای اینکه اون مسیر آزاد شود و ما باید صبر کنیم جز تاخیر های ما هستند. تاخیر بسته بندی که واسه صدا هست یعنی باید صدای آنالوگ ما بسته بندی بشود و عدد بشود و بفرستد این هم تاخیر هست.

پایان جزوه پنجم.

**جزوه ششم:**

انواع روش سویچینگ، مهم ترین بحث برای ما پکت سوییچینگ هست چون TCP این ها استفاده میکنند. سوییچینگ واسه سوییچ های شبکه اینترنت هست. هدف این هست که این کامپیوتر ها تبادل پیام با هم بکنند. دو دسته پیام داریم یک پیام های داده و پیام های کنترلی که مثلا واسه وقتی پیام های پروتکول هستند که مثلا این پیام ها به چه ترتیبی باید ارسال و دریافت بشه و چه جوابی بدهیم این ها با پیام های کنترلی انجام میشود. در بعضی پروتکول ها دست تکانی داریم یعنی ابتدا طرفین رضایت بدهند بعد ارتباط برقرار شود. پس ببین ما یک پیام بزرگ رو قطعه به قطعه میکنیم بعد در لایه IP قسمت شبکه به هدر آن یک چیزی اضافه میکنیم و تبدیل به پکت میکنیم پس پکت با قطعه دقیقا برابر نیست. سوییچ های مربع شکل سوییچ های اترنت یا لایه دوم یا محلی هستند و سوییچ های بیضی شکل سوییچ های IP یا روتر هستند.

دقت کن که اول باید کل بسته در بافر ورودی کامل ذخیره شود تمام اون بسته بعد هدایت پس اول ذخیره میکنیم آن هم به صورت تمام بسته ها یعنی در بافر ورودی اینقدر صبر کنیم تا همه بسته ها برسند بعد شروع میکنیم به هدایت کردن نکته بعدی این هست که این ها دیفالت برای سوییچ های بسته هستند، هر سوییچ به این شکل نیست برای پکت سوییچ و سوییچ بسته هست پس دیفالت ما پکت سوییچ و دیفالت پکت سوییچ، سوییچ های بسته هستند که به شکل ذخیره و هدایت عمل میکنند. وظیفه روتر هدایت کردن آن هست که ما به کدوم لینک خروجی برویم. چطوری میفهمیم کجا باید بره؟ به هدر آن نگاه میکنیم که آدرس مقصد کجاست بعد تو جدول خودمان نگاه میکنیم که این آدرس باید به کدام روتر برود و هدایت میکنیم. اون آدرس همان IP هست. منتها ما بخشی از آن IP را کار داریم. ای پی ها مثل آدرس ها از کل به جز هستند یعنی اینکه مثلا میگوییم ایران، کرج، مصباح، بعد همینطوری سلسله مراتبی هستند یعنی مثلا IP از سه قسمت تشکیل میشود. و یک پیشوند دارد که مثلا اگر از تهران بخواد بره آمریکا دیگه کاری به ایالت و آدرس خیابون ندارد. مثلا میگه هر کی آمریکا هست بره روتر دبی.

تمام ای پی را ذخیره نمیکنیم چون خیلی بزرگ میشود صرفا کشور ها باشد کافی هست دیگه به آدرس جزئی کاری نداریم که. کی جدول ها را ساخت؟ الگوریتم های مسیریابی که بهترین مسیر ها را پیدا کردند. نگاشت کردن یعنی اینکه میخوای بری آمریکا برو فلان روتر.

در اون مثال پمپ بنزین ها روتر هستند. مثلا پمپ بنزین چالوس دیگه کاری نداره شما کجا میخوای بری تهران میگه عه میخوای بری تهران میگه فلان جا برو. بعد کل آن را نمیگوید تا بعدی را میگوید پس همینطوری باید بپرسیم. دقت کن ترتیب رسیدن بسته ها در پکت سوییچینگ منظم نیست و ما بعد از رسیدن به مقصد باید مرتب کنیم.

سوییچینگ مداری: در این شبکه ما منابع مثل نرخ انتقال لینک ها برای اینکه به thruput مورد نظر برسیم و بافر ها چه قدر حافظه برای بافر کردن نیاز داریم و پردازنده، در طول مدت نشست ما رزرو میکنیم. در صورتی که تو سوییچینگ بسته ای ما چیزی رزرو نمیکنیم چیزی بخواهیم درخواست میدیم اگر ندادن در صف منتظر میمانیم. ولی توی مداری ها ما میتوانیم از قبل یک سری منابع رزرو کنیم تا کیفیت سرویس حفظ شود. در شبکه های تلفنی مثل PSTN از سوییچینگ مداری استفاده میشود و یک اتصال فیزیکی ثابت با پهنای باند ثابت 4 کیلوهرتز برقرار میشود. مهم ترین ویژگی نرخ ثابت تضمین شده هست. اگر رو روتر ضربدر باشد علامت پکت سوییچ هست. و شارپ علامت سوییچینگ مداری هست. داخل لینک ها چندین چنل هست که پهنای باند آن ها فعلا یکسان هست به صورت پیش فرض هم مساوی بگیر. چه قدر باید رزرو کنیم؟ حداکثر نرخ مورد نیاز هست. و همین بدبختی ما خواهد بود.

TDM ایستا یعنی زمان رو به یک سری قطعات یا برش تقسیم میکنیم به صورت مساوی و در هر فریم برش زمانی را به یکی از اتصالات اختصاص میدهیم. در FDM پهنای باند را تقسیم میکنیم. FDM برخلاف TDM هست یعنی توی برش اول کل پهنای باند را به یکی میدهیم. اما در FDM در تمام زمان ها به طور پیوسته هست ولی پهنای باند را تقسیم میکنیم مثلا 4 کیلوهرتز رو میدیم به فلانی 4 تای دیگه رو میدیم به اون یکی دیگه و دیگه زمان معنی نداره و برای همیشه به آن میدهیم تا وقتی که ارتباط برقرار هست.

WDM: یا تقسیم طول موج هست که در فیبر های نوری استفاده میشود و یک بخشی از طول موج را به یک کانال اختصاص میدهد. مثلا یک مقداری رنگ قرمز رو میدیم به یک کانال.

CDM: همه کانال ها در همه زمان ها پهنای باند رو میدهیم بهشون و خوب قاطی میشود که عجیب هست که این ها جدا هستند و با هم قاطی نمیشوند.

چون اینها عمود هستند و میتوانیم یکی رو گوش بدیم یکی رو گوش ندیم مثلا فقط فارسی رو چرا چون روسی و فرانسوی و آلمانی اینا ها عمود بر هم هستند و ما میتوانیم از یکی از اینها استفاده کنیم.

گفتیم سوییچینگ مداری مزیت تضمین کیفیت را دارد ولی هدر دهنده منابع هست که باعث میشود ما در شبکه استفاده نکنیم. چرا چون مثلا یک مقداری به ما دادن ولی ما نمیتوانیم استفاده کنیم پولش را باید بدیم و اون بخشی که استفاده نکردیم از دست رفته است. مثلا تو به عمت زنگ بزنی و نیم ساعت سکوت کنی درسته کاری نکردی ولی چون بهت اختصاص دادن پولش را میگیرند. اتلاف زیاد میشود. پیچیدگی هم داریم چون هعی باید رزرو کنیم و سخت هست. اگر سادگی رو میخوای ولی نمیتونی تضمین کنی باید بپردازی چون بهای سادگی هست در پکت سوییچینگ.

تعداد لینک ها یکی بیشتر از تعداد سوییچ ها هست. در شبکه سوییچینگ مداری ذخیره و هدایت نداریم در پکت سوییچ داشتیم. دیفالت پکت سوییچینگ ذخیره و هدایت هست. و دیفالت سوییچینگ مداری ذخیره و هدایت ندارد و تا بسته میرسد میفرستد پس زمان انتقال مستقل از تعداد لینک ها هست یعنی چه 1 لینک باشد چه 100 لینک. و مستقل از تعداد گام ها هست.

پایان جزوه ششم.

جزوه هفتم:

**جلسه هفتم:**

شبکه سوییچینگ مداری در تلفن استفاده میشود. سویچینگ بسته ای کیفیت خدمات را تضمین نمیکند. پروتکول TCP وقتی میبیند شبکه دارد ازدحام میشود میگوید نرخ ها را پایین بیاورید و با سرعت کمتری بفرستید. ولی در کل شبکه سوییچینگ بسته ای کنترلی روی تعداد کاربران نمیگذارد. 10 تا یوزر برای ما مشکلی ایجاد نمیکند.

یک نگاه دیگر از شدت ترافیک نرخ بیت ریت ورودی با بیت ریت سرویس دهی هست. همان لاندا به میو هست صرفا نسبت به بیت ریت نوشتیم.

کارایی در شبکه سوییچینگ بسته ای 3.5 برابر داریم بیشتر استفاده میکنیم از تعداد کاربران ولی کاربران 0.996 که تقریبا صد در صد هست کاربر ها همون حس سوییچینگ مداری را دارند و میخواد بهینگی نسبت به اتلاف را به ما نشان دهد. در سوییچینگ مداری وقتی 9 تا کاربر دیگه غیر فعال هستند فقط 100 کیلوبایت به شما میرسد حتی اگر بقیه غیرفعال باشند. ولی در سوییچینگ بسته ای کل مقدار کانال یعنی 1 مگابایت به ما میرسد. فقط بهینگی نیست سرعت لحظه ای هم بیشتر میشود.

مدار مجازی: گفتیم سه تا شبکه داریم سوییچینگ بسته ای و سوییچینگ مداری و ترکیبی از این 2. چرا ترکیب کردیم؟ مزایای این 2 را میخواهیم استفاده کنیم ما هم بهینگی را میخواهیم و ساده باشیم از سوییچینگ بسته ای و هم میخواهیم رزرو منابع و کیفیت سرویس را از سوییچینگ مداری میخواهیم. چون میخواهیم یک مدار برقرار کنیم نیاز به یک برقراری اتصال داریم. آخر هم بعد از استفاده باید آزاد کنیم منابع را. ما یک مسیر مشخص و ثابت داریم مثل مداری و تا وقتی که قطع نشده هست. همه مسیر ها از بسته ثابت میروند و ترتیب بسته ها نیز حفظ میشود و شروع کندی دارد چون باید برود ارتباط را تنظیم کند و منابع را باید رزرو کند همه اینها مثل سوییچینگ مداری هست. خیلی شبیه به مداری هست ولی دو تفاوت عمده دارد اول اینکه بر خلاف مداری رزرو منابع به اندازه حداکثر نیست. ولی اینجا اندازه ظرفیت معادل هست. یک پارامتری آماری شما فکر کن مثل میانگین هست. و به اشتراک میگذارد. مثلا درسته ظرفیت معادل به ما داده اند ولی اگر یک موقعی ما کمتر استفاده کردیم میدهیم به بقیه و اگر یک موقعی بیشتر شد از ظرفیت بقیه به ما میدهند و ما به بهینگی نزدیک میشویم. البته چون ممکن هست همه همزمان بیان و بار بگذارند ترافیک بالا رود و بافر ها شروع کنند به پر شدن و سر ریز و تلفات داشته باشیم. ولی طبق محاسبات آماری پیچیده میایم تضمین میکنیم از یک حدی بیشتر نشود. در مدار مجازی تا حد بالایی تضمین میدهد. درست هست هیچ کدوم صد در صد نیست. فاز برقراری سوییچینگ بسته ای نداشت یهو شروع میکرد ولی مداری مجازی چون منابع باید رزرو کنند و مدار باید تنظیم کنند.

پایان جزوه هفتم.

**جزوه هشتم:**

فقط دقت کن اونجا که مینیموم میگیرد. مثلا میگوید این بین store که هست مثلا 50 و 100 تا میخواد برای ارسال و بسته 1 که داره میفرسته قطعا بسته 1 کمتر از بسته 2 هست دیگه پس میگذاریم بسته 1 بفرستد. در واقع ماکس گرفتیم بین 50 و 140. صف کجا تشکیل شد؟ اون لینکی که کم تر شد پس عین جاده هست دیگه جایی که جاده باریک شود صف تشکیل میشود و هنگام که از باریک به گشاد معلوم هست تشکیل نمیشود. از گشاد به باریک صف تشکیل میشود.

در روتر لینک اول اثر لینک قبلی از بین رفت زیرا درست هست بسته ها در ثانیه 50 میرسند ولی تا 140 باید صبر کنند در نتیجه دیگر آن را نمینویسیم.

در تست اول ذخیره و هدایت غلبه کرد بر تاخیر صف. فرمول نهایی Rmin صرفا فقط میخواد واسه گلوگاه را بگوید که چند بار اتفاق میفتد.

تا ساعت 1:09 دقیقه از این قسمت دیده شد.

دقت کن در ستون دوم اینقدر سرعت بالا هست که اصلا اون سرعت انتقال تاثیری ندارد چون دیر میرسد. توی سوییچینگ مداری فقط لینک گلوگاه توی تاخیر انتقال تاثیر میگذارد.

گذردهی انتها به انتها یعنی با چه نرخ مفیدی بسته تحویل مقصد داده میشود. کلا سرعت را گلوگاه مشخص میکند چون بقیه که سریع رد میکند میرود. R min همان نرخ گلوگاه هست.

و گذردهی گلوگاه میشود f تقسیم بر مینیموم R ها. دقت کن بعضی اوقات گلوگاه داخل هسته شبکه هست یعنی درست هست لینک ما کمتر از میزان لینک کل هست ولی لزوما اون به عنوان مینیموم انتخاب نمیشود. اگر R اصلی ما خیلی بزرگ بود میانگین R سرور و R کلاینت اون نرخ انتقال هست ولی اگر نبود باید R اصلی را تقسیم کنیم بر تعداد کانکشن ها و اگر کمتر از R سرور و کلاینت بود اون R ما هست. اون internet همون لینک گلوگاه هست. به علائم روی روتر ها دقت کن که پکت سوییچینگ هستند یا سوییچینگ مداری. اگر نگفت بیت یا بایت بگیر بیت نسبت به بقیه چیز ها باید ببینی. دقت کن r ها را در فرمول میتوانی یکی بگیری اگر گفته باشد نرخ انتقال همه یکی هست. اگر نرخ انتقال بیت اول کمتر از بیت دوم بود اصلا صف تشکیل نمیشود چون قیف نیست. در نتیجه اونجا queue صف رو تعیین نمیکند بلکه هدایت و ذخیره تعیین میکند.

**جلسه نهم:**

لایه های پایین تر عملیات های گام به گام و یا هاپ بای هاپ دارند بر خلاف بقیه لایه ها که عملکرد شبیه آینه ای دارند. ارتباط بین لایه ها به این شکل هست که هر لایه به لایه بالاتر از خودش سرویس میدهد و از لایه پایین تر از خودش سرویس میگیرد. هر لایه فقط با لایه بالاتر و پایین ترش فقط ارتباط دارد و بقیه ارتباطی ندارد.

در معماری لایه ای هر باکس ما یک لایه هست. چون 3 تا چیز در هر معماری مهم هست که دو تاش رو بالا گفتیم دومین این هست که ارتباط بین اجزا چطور هست؟ نظم خاصی دارد؟ سومین نکته چه ارتباطی یا اتصال کننده ای بین لایه ها داریم مثلا فراخوانی یک تابع در معماری شی گرایی. در معماری لایه ای به این شکل هست که ما از لایه پایین تر درخواست سرویس میکنیم.

پروتکول بین لایه های متناظر هست. یعنی یک جز اینطرف با همون جز اونطرف و در یک لایه قرار دارند و یک پروتکول استاندارد دارند و گفتگو میکنند و توسط هدر هایی که به بسته های اضافه میشود هست. چون درست هست ما فقط مثلا با لایه پایینی کار داریم ولی یک هدر به اون بسته میگذاریم و اون بسته از صد ها سوییچ میگذرد و اونجا که رسید مثلا اونور اون لایه متناظر میگیرد و اون پیام را میخواند. ارتباط مستقیم نیست بلکه از طریق هدر های بسته هایی که جا به جا میشوند صورت میگیرد. این پروتکول هست و ارتباط بین لایه ها با پایین بالایی خودش به صورت پروتکول نیست بلکه به صورت سرویس استاندارد هست. مزایا رو حفظ کن. در مقابل این همه مزیت یک عیب وجود دارد. این هست که وقتی ما این همه مولفه داریم به جایی اینکه به یک سیستم بگیم یک کار انجام بده اول به اولی میگوییم اولی یک کار انجام میدهد بعد به دومی میگوید و دومی به سومی و ... ، این مبادله کردن ها و اینکه هر سری یک محاسبات داریم و هر لایه باید یک هدر طراحی کنند و بگذارند و علاوه بر سر بار زمانی سربار داده هم رخ میدهد. و هر چه تعداد لایه ها بیشتر و مولفه ها بیشتر میدهیم هزینه زیاد تر میشود سربار بیشتر کند تر و performance میاد پایین البته ارزش دارد. البته بهره وری هم کم میکند. سرویس به لایه بالاتر از دو بخش تشکیل شده است یک بخشی که شامل خدماتی هست که خود اون لایه ارائه میدهد و دو بخشی که از لایه های پایین تر گرفته است چون از عهده خودش بر نمیاد. ممکن است یک سرویسی تا قبل از یک لایه مطمئن باشد و وقتی به اون لایه رسید نامطمئن شود مثل ip در لایه شبکه ولی وقتی همون ip در TCP بشیند مطمئن هست. یعنی پیام هایی که میفرستد را یک کپی میگیرد و اگر بسته ای نیومد دوباره میفرستد مثال همون خدمتکار.

سیستم های باز یعنی سیستم هایی که پروتکول ها را رعایت میکنند مثل سیاست کشوری که پروتکول ها را رعایت میکند با کشور های بیشتری ارتباط دارد و باز هست بر خلاف کشور های بسته یا سیستم های بسته.

لایه ارائه: سرویسی فراهم میکند که از طریق آن همه برنامه های کاربردی ارتباطی قادر به تفسیر و یا ترجمه مفهوم داده های در حال تبادل با یکدیگر هستند. مثلا یک کد را رمز نگاری کردیم هکری نفهمد و چطوری تو لایه متناظر این پیام را متوجه میشود؟ با همین لایه.

این وظیفه لایه ارائه بود. کلا هر تغییر شکلی از داده و هر شکل از داده باشد و تغییری در محتوا باشد فقط در لایه ارائه هست. امروزه اگر استفاده شود باید لایه اپلیکیشن مدیریت کند. البته لایه های دیگری هم امروزه دخالت میکنند.

لایه نشست: وظیفه این لایه تعریف سرویس های مربوط به همزمانی یا همگام سازی synchronization و عدم محدودیت مبادله داده ها هست مثلا یک سری چک پوینت تو یک فایل 1 گیگ میگذاریم که اگر به مشکلی خورد نیایم از اول 1 گیگ را ارسال کنیم و این رو هم لایه اپلیکیشن به طور مجزا باید انجام بدهد نه به این معنی که مهم نبوده اند. مهم ترین مشکل سربار پشته پروتکلی هست. و این سربار مثلا در مبدا برای ارسال یک پیام لایه به لایه میره پایین و این هعی درخواست کردن سربار زمانی دارد و هر لایه یک سری محاسباتی دارد مثلا مبدا کجاست مقصد کجاست n تا field دارد و میچسبند به هدر اون پیام و بفرستد به لایه پایین و خود این محاسبات و دست به دست کردن زمان انتقال بیشتر میشود.

تا 65 دقیقه اول از جلسه نهم دیده شد.

چرا میگویند پشته؟ چون هر چیزی که تو مبدا اتفاق میفتد توی لایه ها توی مقصد برعکس اتفاق میفتد یعنی مثلا اگه اونور از لایه اپلیکیشن بود اونور مثلا از لایه اول هست. یعنی آخرین هدر که چسبیده شد تو مبدا اولین هدر هست که تو مقصد برداشته میشود. دقت کن لایه ها هدر خود را میچسبانند میدهند به بعدی. مدل 5 لایه ای سربار کمتری دارد. یکی از مشکلات لایه ای به نوعی تکرار هست یعنی مثلا کنترل خطا هم تو لایه دیتا لینک هست هم تو لایه انتقال تازه اونور هم تست میشود. و همچنین کاملا جداسازی انجام نمیشود. دیتا لینک لایه سخت افزاری هست. پروتکول های لایه پیوند درون کارت شبکه هست و همچنین داخل سوییچ ها و دو سر هر لینک یک سوییچ کارت شبکه وجود دارد. جز دو لایه آخری 3 تا لایه اولی تو هر سوییچ وجود دارد البته سوییچ های لایه سوم منظور ما هست. HTTP برای وب هست FTP برای فایل و SMTP برای سرویس ایمیل هست.

لایه کاربرد: نرم افزار های شبکه و پروتکول های مربوط به آن اونجا قرار دارند و لایه کاربرد یک بخشی از نرم افزار هست تمام آن نیست. و هر چیزی که درون اپلیکیشن هست لایه کاربرد نیست. در لایه کاربرد علاقه داریم با نام کار کنیم نه آدرس 32 بیتی. یعنی مثلا گوگل را میگیرد به آدرس ای پی گوگل تبدیل میکند در سیستم نام گذاری دامنه.

لایه انتقال: مهم ترین پروتکول آنها TCP و UDP هستند. بعضی ها از اولی و بعضی ها هم از دومی استفاده میکنند. مثلا در HTTP SMTP FTP همگی از TCP استفاده میکنیم هر جا که فایل باشد از پروتکول مطمئن TCP استفاده میکنیم. TCP سرویس مطمئن اتصال گرا هست. اولا تضمین میکند بدون خطا و تلفات تحویل بدهد. دوما کنترل جریان میکند یعنی وقتی احساس میکند بافر مقصد داره سر ریز میشه از این حجم از جریان ورودی سرعت فرستنده را کم میکند و شبکه را شلوغ نکند. سومین قطعه بندی و سرهم کردن قطعات هست یعنی یک پیام را قطعه به قطعه میکند هدر میگذارد تو مقصد هم سر هم میکند. چهارمین مورد کنترل ازدحام هست که وادار میکند فرستنده ها نرخ شبکه را کاهش بدهند. UDP هیچ کدام از اینها را ندارد و ضمانتی برای تحویل و ... ندارد. دقت کن لایه انتقال هیچ چیزی اون وسط اتفاق نمیفتد فقط انتها به انتها رخ میدهد. مالتی پلکسینگ یعنی اینکه پیام ها را از لایه اپلیکیشن بگیریم و پلکسینگ یعنی تحویل به کاربرد مورد نظری که داریم مثلا به FDP کلاینت بدیم یا بروز کروم. UDP فقط همین 2 تا را انجام میدهد و بقیه را انجام نمیدهد.

لایه شبکه: از اینجا به بعد نه تنها تو انتها به انتها بلکه تو سوییچ های لایه سوم هم این لایه انجام میشود. اینکه بسته ها از کدام روتر عبور کنند چگونه آدرس دهی کنیم بسته را هدایت کنند و ... وظیفه لایه شبکه هست به این بسته ها دیتاگرام هم گفته میشود. همون پکت هست منتها تو این لایه دیتاگرام نام دارد. مهم ترین وظیفه همان هدایت بسته ها یا دیتاگرام ها هست. آدرس دهی و ساخت بسته هدر به نام ip وظیفه لایه شبکه هست مهم ترین آنها البته. این جدول مسیریابی که مثلا از تهران میخواهیم بریم نیویورک از کجا باید بریم وظیفه این لایه شبکه هست این اطلاعات را لایه شبکه در جدول هدایت گذاشته هست. پروتکول ip برای آدرس دهی و پکت سازی هست و سایر پروتکول ها هم هست مهم ترین پروتکول در لایه شبکه ip هست به لایه شبکه لایه ip هم گفته میشود ip همان چسبی هست که همه را بهم چسبانده است. وظایف لایه شبکه را در خاطر داشته باش.

لایه پیوند یا داده: از حرکت یک روتر به دومین روتر مثلا از هاست به اولین روتر کلا هر گره ای که بود وظیفه لایه پیوند هست که تشخیص بدهد بسته را از کدام لینک خارج کند و روی لینک بفرستد و تحویل بدهد رو روتر مقصد یا کارت شبکه مقصد. مثلا فیبر نوری هست پس بسته یا دیتاگرام را تحویل کارت شبکه مخصوص میدهیم. کنترل خطا بین دو روتر هم داریم بین دو لینک. درست هست تکرار هست ولی چون همون جا تشخیص میدهیم خطا را دیگه منبع صرف نکنیم تا مقصد برسد بعد بفهمیم خطا هست.

به بسته ها یا دیتاگرام فریم گفته میشود در لایه انتقال. مسیج میاد تو لایه انتقال میشود قطعه تو لایه network میشود دیتاگرام و تو لایه پایین تر فریم گفته میشود. لایه فیزیک هیچ درکی از هدر ندارد فقط بیت به بیت منتقل میکند.

لایه فیزیکی: فقط بیت ها را جا به جا میکند کاری ندارد برای کی هست چی هست بی شعور هست و درکی ندارد.

کپسوله سازی: جمع بندی همه حرف ها تا اینجا هست. دقت کن که توی عکس توی سوییچ لایه سوم وقتی شبکه بسته را به لایه پایین تر میدهد یک هدر جدید پیوند تولید میشود زیرا اصلا فرق میکند چون لایه شبکه تشخیص داده به کدام قسمت باید برود. هر گره تشخیص خطا دارد و مثلا به گره قبلی میگوید فلانی دوباره بفرست این خطا داشت. هدر ها بعد از استفاده دور انداخته میشوند. مفهوم کپسوله سازی کجاست؟ قطعه میرود داخل یک بسته کپسوله میشود یک بسته داخل یک دیتاگرام کپسوله میشود و دیتاگرام توسط یک فریم کپسوله میشود. MMS برای جاهایی که احتمال هدر رفت بسته مثل ارتباط بی سیم زیاد هست. بیا اون مسیج را تکه به تکه کن بزرگترین تکه را بردار یک هدر اضافه کند میشود MSS. البته تکه را بدون هدر بزرگترین را بردار هدر جز آن نیست. Port یعنی اینکه وقتی به اون هاست توسط ip رسیدیم به کدام قسمت باید تحویل بدهیم مثلا 5 تا برنامه دارد اجرا میشود مثلا HTTP روی پورت 80 هست پورت ادرس اپلیکیشن هست که ادرس اون برنامه کاربردی در هاست هست. به ازای هر بسته هر فریم داریم یعنی ده تا بسته دادیم ده تا فریم هم داریم. حداکثر داده درون یک قطعه MSS هست. پیام جز لایه کاربرد هست. 70 بایت در تست آخر جمع سرآیند باقی لایه ها هست. دقت کن که ما میخواهیم 1000 بایت پیام بفرستیم میخواهیم ببینیم کوچک ترین بخش چه قدر هست که بتوانیم توی هزار تا بفرستیم حالا یک قسمتی هدر هم داریم. مسیر مشخص هست دیگه اول از اون MAX FRAME SIZE سایر هدر ها را کم کن تا MSS بدست بیاد بعد ببین توی اون مقداری که میخواهیم بفرستیم که 1000 تا هست چند تا پیام جا میشود بعلاوه اینکه خوب باید سربار هم بفرستیم دیگر.

**جلسه دهم:**

تا دقیقه 5 دیده شد.

همتا شدن به این معنی هست که مثلا 50 درصد ترافیک ما داخل کشور هست و به جای اینکه ما به خارج وصل شیم تا بتوانیم از یک ISP به ISP دیگر وصل شویم خود ISP های منطقه ای را به هم وصل کنیم و به نوعی همتا شویم.

با حمله داس سعی میکند منبع را در اختیار بگیرد مثلا در سیلابی نمودن پهنای باند کل پهنای باند را در اختیار میگیرند تا کاربران چیزی نتوانند منتقل کنند جاده بسته هست ولی سرور بیکار هست. حملات آسیب رسان به این شکل هست که خود سرور یا سیپیو را اینقدر مشغول میکنند که وقت نمیکند کار های شبکه را انجام دهد. سومین داس حمله به TCP هست همینطوری هعی اتصال برقرار میکنند و ظرفیت full میشود و کسی نمیتواند وصل شود چون مثلا سیستم میتوانست ده هزار اتصال برقرار کند ولی الان همه پر شده هست و کسی نمیتواند وصل شود.

**فصل دوم:**

**جلسه یازدهم:**

در این فصل قرار هست با سرویس ها آشنا بشویم و استفاده بکنیم اینکه TCP یا UDP چیکار میکند به ما مربوط نیست. صرفا حدودی آشنا میشویم. دقت کن برنامه های ما در سیستم های انتهایی فقط اجرا میشوند و سوییچ های هسته شبکه اصلا این لایه های کاربرد و انتقال را ندارند. این برنامه ها اصلا در روتر ها عمل نمیکنند. فقط 3 تا لایه اول را کار دارند و 2 لایه دیگر ربطی به هسته شبکه ندارد گفتگو ما مربوط به لایه کاربرد هست. سوییچ های لن دو لایه و سوییچ های هسته روتر 3 تا لایه دارند. از دیدگاه برنامه نویس شبکه اصلا کاری به 5 لایه شبکه و لایه انتقال و پایین تر ندارد سرویس میخواهد بداند چه سرویس هایی میخواهد بگیرد و کجا ها میخواهند اجرا بشوند. معماری کلاینت سرور: یک میزبان به عنوان سرویس دهنده میزبان هست و سرویس میدهد و دائم به درخواست مشتری گوش میدهد و همیشه فعال هست و مشتری ها درخواست میدهند. فرض کنید ما آپاچی هستیم یک مشتری میخواد مثلا فایل اچ تی ام ال صفحه اول دیجی کالا و ما میفرستیم براش. دقت کن کلاینت با کلاینت هیچ موقع ارتباط ندارد فقط با سرور. سرویس دهنده IP ثابت دارد اگر خواستند تغییر کند در سرویس های DNS دنیا عوض میکنند تا همه آدرس جدید را بشناسند. TELNET برای لاگین کردن به ماشین از اونور دنیا هست. همه معماری دهه 70 کلاینت هستند و معماری دهه 90 همتا به همتا هست. گوگل فیبر های نوری خودش را از زیر دریا عبور داده و سرویس اینترنت خودش را دارد و از کسی سرویس نمیگیرد. درست هست یک کامپیوتر هست ولی این ظاهر هست به صورت خوشه ای هستند در همتا به همتا همه به هم سرویس میدهند فقط از سرور سرویس نمیگیریم. یعنی کلاینت ها هم به هم سرویس میدهند.

کلاینت سرور مثل اسپاتیفای هست ولی همتا به همتا مثلا تو تلگرام موزیک اشتراک میگذاریم مثلا اون ساسی مانکن دارد من زدبازی. مثلا تورنت نگاه میکند کیا این موزیک رو دارند دنبال میکند ببیند کیا فعال هستند میگوید بفرستید واسه فلانی. معماری ترکیبی هم داریم یک بخشی همتا به همتا یک بخشی کلاینت سرور مثلا ارسال پیام فوری برای اینکه IP دوستمون رو پیدا کنیم از سرور کمک میگیریم ولی وقتی مستقیم به دوستمون پیام میدهیم همتا به همتاست دیگر سرویسی اون وسط نیست. در کلاینت سروری خود مقیاس پذیر نیست ما باید خرج کنیم که پهنای باند و مقاومت بالاتر برود خودش نمیتواند باید پول خرج کنیم سرور بیاریم معماری همتا به همتا ما پول خرج نمیکنیم چون وقتی هزار تا مشتری داشتیم هزار نفر هم جواب میدهند خوب اگه 1 میلیون مشتری بیاد اونا هم 1 میلیون مشتری هستند که موزیک دارند پس پهنای باند هم 1 میلیون برابر میشود. قدرت پاسخ دهی خودش، خودش را مقیاس پذیر میکند. گاهی اصلا ممکن هست یک تیکه فایل را یکی میدهد یکی دیگرش را دیگری درست هست که همشون دارد ولی برای پهنای باند بیشتر 3 نفر اپلود میکنند. در ADSL همین هست. درخواست جدید: ظرفیت جدید. مورد پسند ISP نیست چون قبلا دانلود فقط بود الان باید اپلود دانلود برابر باشه و باید عوض شود بحث امنیت هست. چون معماری باز و توزیع شده دارد و یک حفره ایجاد شود و نفوذ کند یک هکر و اطلاعات شخصی را بدزدد و ما میترسیم. ایجاد انگیزه هم باید بکنیم.

ارتباط فرآیند ها: تبادل پیام مهم ترین راهکار هست. در یک ارتباط ما هر دو نقش کلاینت یا سرور را در همتا به همتا داریم یا میگیریم یا میدهیم ولی در کلاینت سرور ما همیشه برچسب مشتری داریم. در همتا به همتا همزمان میتواند سرویس و مشتری باشد مثلا یک موزیک دارد دانلود میکند و یک فیلم دارد میدهد. دو برنامه در حال اجرا با استفاده از سوکت با هم ارتباط برقرار میکنند. سوکت در خونه هست یکی یک پیام بخواهد بفرستد باید از در خونه برود بیرون و اونکه میخواهد بگیرد باید از در برود داخل بین اینها هم شبکه پستی هست. این فرآیند ها در لایه کاربرد هستند زیر آن لایه انتقال هست که TCP هست. TCP اول یک دست میدهند چون نیاز به ارتباط دارد. هر در خونه یک پلاک دارد به نام port number. API ها یک واسط بین برنامه ها و لایه های شبکه هستند و برنامه های تحت شبکه را به کمک آن مینویسم. برنامه نویس در سمت لایه کاربرد سوکت کلا کار میکنند و کنترل ناچیزی در لایه انتقال دارد و بقیه آن را برنامه نویس لایه انتقال کنترل میکند صرفا حداکثر اندازه بافر و ... رو مشخص میکنند. آدرس IP میشود آدرس ماشین ولی اینکه با کدوم فرآیند کار داریم میشود port number. مثلا اون سوکت رو چطوری پیدا کنیم با اون شماره پورت. 0 تا 1023 اینا از قبل رزرو شده است همینطوری نباید به کسی بدهیم. 1024 به بالا میتونیم تصادفی بدهیم.

**جلسه دوازدهم:**

حالا باید ببینیم کی از TCP, UDP باید استفاده بکنیم پس باید ببینیم چه خدماتی اینها به ما میدهند تا نسبت به نوع اپلیکیشن ما یکی را انتخاب کنیم. گذردهی یعنی یک حداقل سرعتی تضمین میکند. هماهنگی زمان یعنی یک حداکثر تاخیری برای ما تضمین میکند یا نه.

دقت کن بسته ها باید به ترتیب درست و بدون خطا و بدون تلفات برسند. انتقال داده مطمئن با تصحیح یا ارسال مجدد بسته ها تلفات و خطا را جبران میکند البته همه اپلیکیشن ها به این نیازمند نیستند. توی برنامه ای که فایل جا به جا میکند نیاز هست چون یک بیت جا به جا شود کل فایل بدرد نمیخورد. لایه شبکه تلفات دارد لایه انتقال باید جبران کند. تصحیح خطا هم داریم. لایه شبکه همه نواقص گفته شده را دارد یعنی ترتیب حفظ نمیکند تکرار و تلفات و خطا دارد و همه این ها باید در لایه انتقال جبران شود. بعضی ها مثل TCP میکنند بعضی ها UDP نمیکنند.

تا دقیقه 17 دیده شد.

گذردهی: چند بیت مفید میتوانیم به مقصد تحویل بدهیم نرخ موثر تحویل هست. گذردهی را دقیقا نمیتوانیم بدست بیاریم بخاطر عوامل مختلف مثل تاخیر های مختلف. طبیعتا برنامه های که نیاز به سرویس بلادرنگ دارند یا برنامه کاربردی تلفن اینترنتی خوب نیاز داریم که تضمین کنیم با یک نرخ ثابت بفرستیم و حساس به پهنای باند هستند. گذردهی مفید بیت ریت هست. مهم ترین معیار های مربوط به هماهنگی زمان تاخیر و نوسان تاخیر یا همون جیتر هست. و باید ببینیم برنامه ما نسبت به تاخیر و جیتر حساس هستند یا نه برای سیستم های بلادرنگ تعاملی مثل بازی ها مثل رینبو تاخیر و نوسان تاخیر خیلی مهم هست. و بافر کردن سخت میشود چون مثلا نمیتوانی بگویی که اون وسط 9 ثانیه همه صبر کنند تا من بسته ها را بافر کنم. بحث امنیت: ما با استفاده از جامعیت داده میتوانیم بفهمیم که داده ها دستکاری شده اند یا نه یا احراز هویت داریم که ببینیم این شخص همان شخصی که ادعا میکند هست یا نه. حالا باید ببینیم پروتکول ها این 4 سرویس: امنیت، هماهنگی زمان و گذردهی و قابلیت اطمینان در اختیار ما میگذارند یا خیر. جدول نکته برداری شود.

TCP: این یک سرویس اتصال گرا هست یعنی اینکه حتما باید فرستنده و گیرنده یک کانکشن مستقل ایجاد کند و همه بسته ها را از طریق اون ارسال و دریافت کند ربطی به سوییچینگ مداری و بسته ندارد و لایه انتقال هست. یعنی چی کانکشن یعنی یک بافر در مبدا و مقصد ذخیره میکند رزرو میکند و یک سری متغیر های کنترلی دارد مثل کنترل جریان که سر ریز رخ ندهد یا کنترل ازدحام یا کنترل خطا مثلا تو کنترل ازدحام میگوید که فرستنده سرعت را کم کند تا شدت کاهش یابد. UDP اصلا این چیز ها را ندارد و کنترل جریان و خطا ندارد. این یک مفهوم انتها به انتها هست. مطمئن و اتصال گرا هست. دقت کن HTTP, WEB روی TCP هستند یعنی اول باید کانکشن برقرار کنند یعنی همینطوری نمیتوانی بفرستی تا کلاینت و سرور با هم صحبت کنند برای این باید بین کلاینت و سرور یک سری متغیر های کنترلی جا به جا شود که به این میگویند دست تکانی. 3 دست تکانی دارد. اول درخواست ارتباط سرور جواب این 2 تا بعد کلاینت میگوید دریافت کردم صفحه اول دیجی کالا را بده این شد 3 پیام. این اتصال کاملا دو طرفه هست یعنی همزمان که میفرستیم میتوانیم دریافت کنیم. دومین نکته این هست که سرویس انتقال مطمئن دارد یعنی کل داده بدون خطا و تلفات و تکرار و با ترتیب صحیح تحویل گیرنده بدهد و جریانی از بایت ها را همونطوری که میگیرد، تحویل میدهد. همچنین کنترل ازدحام دارد که همه فرستنده ها سهم عادلانه داشته باشند. امنیت: خود TCP سرویس امنی را تضمین نمیکند و بین لایه کاربرد و انتقال یعنی در لایه کاربرد SSL میاد این امنیت را برآورده میسازد. در UDP هم همین شکل هست ولی UDP پشتیبانی از SSL نمیکند. ما تو لایه انتقال فقط 2 پروتکول داریم. SSL در لایه کاربرد هست. باید در طرفین ارتباط نرم افزار و کتاب خانه های مربوط در کلاینت و سرور SSL نصب شود. SSL، API خودش را دارد واسه سوکت های مخصوص خودش و نیازی به TCP ندارد و کدی که نوشته میشود باید متفاوت باشد از سیستمی که استفاده نمیکند. البته دقت کن واسه برنامه هایی که ما نیاز به امنیت داریم. در لایه کاربرد ما بسته را میدهیم به SSL اون رمزنگاری میکند و میفرستد پایین TCP نمیفهمد این چی هست صرفا از لایه کاربرد دریافت و میفرستد اونور لایه TCP کلاینت دریافت میکند و این بسته را تحویل SSL میدهد و به خود اپلیکیشن تحویل میدهد. گیرنده مستقیم TCP را صدا نمیزند. UDP: قابلیت اطمینان ندارد، اتصال گرا نیست، کنترل جریان و ازدحام ندارد و SSL را اجرا نمیکند چون سرویسی نمیدهد پس سبک و ساده هست. هیچ کدام از این 2 پروتکول هماهنگی زمان و گذردهی را هیچکدام تضمین نمیکند چرا چون لایه پایینی آنها تضمین نمیکند. معلوم هست وقتی شبکه تضمین نمیکند ما هم نمیتوانیم تضمین کنیم در لایه انتقال. حالا اگر کاربرد ما حساس به زمان باشد چی؟ ما اینها را داریم چون اغلب ستون فقرات اینترنت اینقدر پهن هست درست هست که IP میگوید انشاالله میرسد ولی این احتمال بالا هست و تلفات خیلی کم هست. یعنی اینقدر خوب هست حالا یکم هم از دست رفت نگرانی وجود ندارد. تضمین نمیکند ولی بهترین تلاش را میکند. TCP سنگین هستند ولی موردی ندارد. حتی یوتیوب اینها هم از TCP استفاده میکنند.

پروتکول های لایه کاربرد: دو دسته هستند عمومی و اختصاصی عمومی ها استاندارد بین المللی دارند و همه دنیا میتوانند استفاده کنند. مثل HTTP که قلب دنیا وب هست جز دسته عمومی هستند. پروتکول اختصاصی هم مثل skype داریم. فرق هست بین برنامه کاربردی مثل ایمیل یا اسکایپ یا تلگرام ولی این به معنای یک پروتکول لایه کاربرد نیست و پروتکول یک بخشی از اون برنامه هست. در واقع قالب اسناد اینجا HTML هست و روی هر سیستم عاملی که ببینیم یکسان هست چون HTML هستند و در کنار اینها ما یک پروتکول داریم که قالب و ترتیب پیام های تبادل شده بین کلاینت و سرور را تنظیم میکنند و کار های دیگه که داخل جزوه و یک جزئی از دنیای وب هست.

**جلسه سیزدهم:**

از مهم ترین تغییرات on demand بودن هست در مقابل تلویزیون که همینطوری broadcast میکند. صفحه وب مجموعه ای از اشیا هست شی یک فایل ساده هست مثل HTML، یا تصاویر یا یک ویدیو هست که به وسیله یک آدرس URL واحد قابل آدرس دهی میباشد. یعنی HTML یک پایه و یک شی و بقیه عکس ها و ویدیو ها یک شی هستند و در مجموع مثلا 11 شی هست ولی هر کدوم از این یک شی ها یک آدرس مخصوص دارند. خود HTML یک شی هست. ما فقط URL، HTML را میدانستیم و داخل آن میگوید ما چه سایر اشیایی داریم و آدرس هر کدوم چیست یعنی تا وقتی که سند HTML دانلود نشود ما آدرس اشیا درون اون صفحه را نمیدانیم. مرور گر برای تک به تک اشیا یک پیام درخواست HTTP میدهد و سرور نگاه میکند اگر معتبر بود ارسال میکند. دقت کن برای محاسبات زمان اول TCP را باید تاخیر آن را حساب کنیم چون اتصال گرا هست. انتقال پیام ها به واسطه سوکت هست و بین میزبان و مهمان توسط سوکت ارتباط میگیرند و مستقیم با فرآیند در ارتباط نیستند در لایه انتقال بلکه به واسطه سوکت مرتبط هستند. ما مطمئن هستیم که میزبان دقیقا همان پیام را دریافت میکند TCP هم همینطور هست وقتی میدهد به IP و اینکه ای پی از چه سوییچ های عبور میکند و اینکه چه آدرس دهی میشود دیگر به TCP مربوط نیست هر لایه فقط به فکر خودش هست و به لایه پایین فکر نمیکند. مثلا اینکه چرا با سیم میفرستد یا بی سیم یا فیبر نوری چه انتشاری چه موجی اصلا ربطی به TCP ندارد. هر قسمت فقط به لایه خودش فکر میکند. برای طبقه بندی پروتکول های لایه کاربرد هم عین لایه انتقال بررسی میکنیم مثلا پروتکول ها از UDP استفاده میکنند یا TCP، یا حالت یا وضعیت اطلاعات مشتری ها را نگهداری میکنند یا نه. که سه حالت دارد پروتکول های بدون حالت، نرم و با حالت اگر هیچی ذخیره نکنیم بدون حالت هست اگر همه چیز را ذخیره کنیم با حالت هست و اگر یک بخشی را ذخیره کنیم نرم حالت هست.

آیا سرور ها اطلاعات مشتری ها را نگهداری میکنند؟ اگر بگیم اره درست هست اگر نه بگیم هم درست هست. پروتکول HTTP ذاتا بدون حالت هست اما اگر ما را شناختند مثلا از کوکی استفاده کرده اند. در حالت نرم برای مدت محدودی نگهداری میکند مثلا تا وقتی که session برقرار هست و اگر log out کنیم دیگر هیچی یادش نیست. از منظر حالت دیدیم حالا از نظر ماندگاری و غیر ماندگاری ببینیم: آیا ما بسته ها را دوره ای یا متناوب میفرستیم یا پشت سر هم میفرستیم، پس ماهیت تبادل داده متفاوت هست اگر قرار هست رگباری بفرستیم بهتر هست همون اتصال TCP را نگه داریم یا به ازای هر درخواست یک اتصال بزنیم؟ جواب این هست که نمیتوانیم بگوییم کدوم خوب هست کدوم بد. چون اگر قرار هست پشت سر هم بفرستیم و هعی بخواهیم قطع و وصل کنیم سربار زمانی دارد پس بهتر هست همیشه اتصال باشد. یک حالت دیگر این هست که درخواست ها هر نیم ساعت یکبار میفرستد خوب چرا نیم ساعت الکی اتصال را باز نگه داریم. از هر دو در HTTP استفاده میکنیم و باید توجه کنیم چون تاخیر های زمانی متفاوت هست. پیش فرض ماندگار هست یعنی اگر هیچی نگوییم ماندگار هست.

کپسوله کردن که یادت هست بسته میاد یک هدر توش میگذاریم میشود کپسوله سازی. TCP وقتی مطمئن شود که گیرنده بسته را گرفته اتصال را قطع میکند. اگر غیر ماندگار باشد برای اشیا دیگر جز سند 4 مرحله اول تکرار میشود یعنی آدرس هر کدوم رو درخواست بفرستد. توازن درخواست هم داریم مثلا 5 تا 5 تا هست مثلا 5 تا اشیا میتوانی انتخاب کنی اگر ده تا باشد باید یکبار این 5 تا بیان بعد یک درخواست دیگر واسه 5 تا بدهی. وقتی میگوییم غیر ماندگار یعنی هر درخواست یک زوج پیام را انتقال میدهد یعنی ارسال و دریافت فقط. توازی یعنی ما 5 تا ارتباط برقرار میکنیم نه داخل یک اتصال 5 تا درخواست بدهیم. پس برای 11 تا اشیا چه موازی چه غیر موازی 11 تا درخواست باید بفرستیم. کلا دو تا رفت و برگشت طول میکشد بعلاوه تاخیر انتقال که L/R هست. اتصالات ماندگاری: غیر ماندگاری برای درخواست های رگباری خوب نیست اصلا چون به ازای هر درخواست باید یک اتصال جدید برقرار کند و بافر ذخیره کند. درجه توازی در ماندگاری بی نهایت هست. یکی خط درخواست حتما داریم بین صفر تا چند خطوط سرآیند یک تا چند تا داریم. حضور host اختیاری هست که در سرآیند باشد چون یکبار درخواست دادیم و IP و آدرس را میدانیم، علتی دیگر این هست که چون ما از کش استفاده میکنیم یک حافظه نهان هست و میخواهد از اطلاعات درون اون field، URL بدرد آن میخورد که ببیند درون کش خودش هست یا نیست.

تا 120 دقیقه اول دیده شد.

متد HEAD: برای دیباگ کردن استفاده میشود یعنی عین GET هست فقط چیزی ارسال نمیشود فقط میخواهیم ببینیم درست هست یا نیست.

ذخیره سازی نهان: ما میتوانیم یک GET شرطی درست کنیم که با استفاده از سرآیند آخرین تغییرات مشخص میکنیم آیا فایلی که ما فعلا در حافظه داریم با همونکه میخواهد سرور بفرستد یکی هست یا نه اگر یکی بود میگوییم دیگر نفرست ولی اگر تغییر کرد بفرستیم که خوب خیلی خوب هست دیگه چون الکی چیزی مصرف نمیشود. پورت 80: وب سرور هست. TELNET لاگین از راه دور است.

**جلسه چهاردهم:**

HTTP ذاتا بدون حالت هستند و سبک هستند و اطلاعات مشتری را نگهداری نمیکنند و کارایی بسیار بالایی دارند بخاطر همین، ولی بعضی سرور ها این اطلاعات را نیاز دارند بنابراین کوکی ها را به وجود آورند تا کاربران را ردیابی کنند. 4 جز داخل پیام پاسخ HTTP قرار میگیرد که مربوط به کوکی ها هست که یک پایگاه داده و فایل و مخصوص و سرآیند مخصوص دارد. روند درخواست به همین شکل هست یعنی اول HTTP سرور نگاه میکند یک درخواست آمده است ولی درون اون درخواست کوکی نیست پس سرور او را نمیشناسند پس یک شماره کوکی برای کاربر میفرستد و یک شماره بهش اختصاص میدهد. و یک پایگاه داده درون کامپیوتر کاربر هست که یک فایل دارد و شماره اون سرور که فرستاده هست مرورگر ما در یادش میماند که کوکی مثلا واسه سایت آمازون چه شماره ای هست و هر موقع خواست به آمازون درخواست بدهد این شماره هم میفرستد. و بعد از طریق سرور یادداشت میکند که این شماره کوکی چه کارهایی این کاربر کرده است و یک جا ثبت میشود. مرورگر هست که شماره کوکی را ذخیره میکند و از اون استفاده میکند و برای سرور میزبان میفرستد. اگر فایل کوکی را پاک کنیم ناشناس میشویم و دیگر سرور ما را نمیشناسد. و بعد دفعه بعدی دیگر نیازی به دریافت اطلاعات نیست مثل کارت بانکی و رمز و آدرس و ... چون سایت آمازون دیگر ما را میشناسد و این اطلاعات را نگهداری میکند. ذخیره ساز نهان وب: کش میتواند در یک ماشین کاربر یا در LAN یا در ISP باشد. و این موجودیت میانی شبکه ای هست که میان درخواست را اول کش میگیرد و میبیند که میتواند جواب بدهد یا نه اگر داشت که خوب به ما میدهد و ما دیگر پهنای باند را اشغال نمیکنیم و مشکلی نداریم اگر نداشت میرود از سرور میگیرد یک کپی به ما میدهد یک کپی هم خودش نگه میدارد. کش کپی اشیا اخیرا را نگهداری میکند که همان local reference سیستم عامل بود که ارجاع محلی بود میگفتیم یه چیز برداشتیم تو آینده نزدیک هم بهش نیاز داریم. دقت کن کش ما دو نقش دارد یکبار سرویس دهنده برای خود کامپیوتر ما و یکبار مشتری به عنوان درخواست برای سرور در صورت نداشتن فایل مورد نظر. پهنای باند گلوگاه بین مشتری و سرویس دهنده خیلی کمتر از مشتری و ذخیره ساز نهان وب هست برای سازمان ما هم بهتر هست چون نیاز به اینترنت را کاهش میدهد و هزینه کمتر میشود برای ستون فقرات اینترنت هم خوب هست چون بار کمتری روی شبکه قرار میگرد و آمار بین 20 تا 70 درصد نشان میدهد که از کش ما اطلاعات میگیریم. دقت کن که شدت ترافیک به نوعی برابر هست نرخ ورود تقسیم بر نرخ سرویس دهی. اگر نرخ دادند این فرمول استفاده شود.

GET شرطی: ممکن است کپی یک شی که وجود دارد قدیمی شده باشد و HTTP این امکان را دارد که بررسی کند که چک کند قدیمی هست نسخه موجود یا جدید. حتما باید از GET استفاده بکنیم.

**جلسه پانزدهم:**

نهان نشده است یعنی کش نداریم اصلا وقتی گفتند DNS و گفتند آدرس IP یعنی کش نداریم. DNS روی UDP میشیند در نتیجه UDP زمان کمتری دارد چون اتصال گرا نیست و اون RTT اضافه را ندارد. و هر کدوم یک RTT دارند پس اگر 5 تا بود میشود 5 تا RTT ولی همین در TCP چون یک RTT دارد میشود ضربدر 2 یعنی 10 RTT. RTT0 برای HTTP هست که تاخیر انتقال هم توی این فرمول دخیل هست. مسائلی که روی HTTP میدهند سه نوع هست: 1. غیر ماندگار بدون توازی 2. ماندگار با توازی که اگر درجه توازی ندادن بینهایت بگیر 3. ماندگار که با خط لوله از درجه بینهایت. چون غیر ماندگار هست: هر شی یک TCP جدا یک اتصال جدا چون توازی هم نداریم باید سریالی بریم جلو. اول باید فایل HTML پایه را دریافت کنیم تا ببینیم چند تا شی هست همون اول نمیفهمیم چند شی داریم و همون اول نمیتوانیم توازی کنیم. دقت کن تاخیر انتقال برای هر شی هست یکی نیست. دقت کن در ماندگار برای کل درخواست های رگباری فقط یک RTT0 طول میکشد. دقت کن تو تست 6 هر کدوم از اون خطوط اتصال یک مگابایت دارد پس اگر 8 تا اتصال موازی داریم پس نباید جمع کنیم با هم و فقط باید زمان انتقال بزرگترین فایل را در نظر بگیریم در قبلی ها یک لینک 1 مگابایتی بود و L توی محدوده باریک حرکت میکردند. در ماندگار کلا یک اتصال داریم و دیگر بحث موازی نیست و خط لوله هست و باید سیگما l/r را بگیرند. یعنی فرقی نمیکند هر خط باشد یا کل خط کلا یک اتصال هست.

HTTP/2: یک مشکل داشتیم به نام HOL یا head of line مثلا اینکه یک خانواده حتما میخواد دختر اولش اول ازدواج کند در صورتی که دختر دوم الکی باید صبر کند. HTTP/2 میاد مثلا یک ویدیو را به صورت قطعه هایی کوچک به نام فریم تقسیم میکند. و لابلای هم ارسال میکند. و با حالت round robin مثلا چند فریم از ویدیو بفرستیم چند فریم از عکس که میشود درهم آمیختن این فریم ها. فیلد سرآیند هم 1 فریم و باقی پیام هم چندین فریم هست. زیر لایه فریم بندی علاوه بر این فریم بندی میاد کدگذاری دودویی هم میکند که باعث کار آمد تر شدن میشود چون فریم ها کوچک تر میشوند همچنین کمتر در معرض خطا قرار میگیرند. همچنین امکان اولویت بندی پیام پاسخ و درخواست ها را داریم. همچنین سرویس دهنده میتواند اشیا دیگر پوش کند تا تاخیر کمتر شود. HTTP/3: در لایه کاربرد میایم زیر HTTP/3 میایم یک پروتکول میگذاریم به نام QUIC که روی UDP میشیند و در لایه کاربرد هست و چندین ویژگی اضافه میکند. اولی همان مالتی پلکس کردن پیام ها هست دوم کنترل جریان مجزا به ازای هر جریان و برقراری اتصال با تاخیر کم.

**جلسه شانزدهم:**

پروتکول FTP بر روی TCP میشیند. FTP بر خلاف HTTP دو اتصال موازی ایجاد میکند یک اتصال کنترلی ماندگار برای انتقال داده های کنترلی و یک اتصال داده، اتصال کنترلی مثلا برداشتن فایل از یک دایرکتوری تعویض دایرکتوری و انتقال فرامین و شناسه کاربری و رمز عبور بفرستیم موقعی که خواستیم انتقال فایل کنیم یک اتصال داده غیر ماندگار ایجاد میکنیم فایل جا به جا میشود و بعد قطع میشود اگر دوباره یک فایل دیگر خواستیم به همین شکل هست یعنی یک اتصال جدید میدهیم. و به شکل خارج از باند هست یعنی یک باند داده های کنترلی یک باند خود فایل. ولی HTTP یک پروتکول داخل باند است. سرویس دهنده FTP نیاز دارد که اطلاعات را حالت مربوط به کاربر را نگهداری کند تا بداند که مثلا در کدام دایرکتوری هست و بخاطر همین یکم سربار ایجاد میکند چون سنگین میشود باید حالات کاربران را نگهداری کند. ایمیل ها در صف پیام ها میماند تا وقتی که سرور دوباره اجرا شود و از طریق پروتکول SMTP این ارتباط صورت میگیرد پس بر روی TCP میشیند. در پروتکول SMTP یک دست تکانی اضافه علاوه بر اونکه در TCP هست داریم. و در این دست تکانی مثلا آدرس ایمیل فرستند و گیرنده را مشخص میکنند و بعد پیام را میفرستند و خودشان را بهم دیگر معرفی میکنند. چرا ایمیل سرور روی کامپیوتر ما نیست و ما فقط یوزر هستیم؟ چون باید کامپیوتر ما همیشه روشن باشد تا بتواند پیام ها را بگیرد و اگر مثلا بعد چند روز روشن نباشد پیام ارسال نمیشود و از بین میرود پس بهتر هست روی یک سرور یا ISP باشد تا صندوق پستی شکل بگیرد. حتی خود فرستنده هم نمیتواند سرور رو خودش داشته باشد چون ممکن است تحت شرایطی اون سرور دهنده خراب باشد و پیام ما ارسال شده باشد ولی نرسد به اون طرف و خوب وقتی یوزر کامپیوتر را خاموش کرده دیگر ارسال نمیشود. SMTP پروتکول تزریقی هست. و یوزر نمیتواند برای خواندن از ایمیل خود از پروتکول SMTP استفاده کند یعنی برای گیرنده پروتکول دیگری داریم چونکه مشخص هست SMTP پروتکول تزریقی هست ولی خواندن پروتکول استخراجی هست به جای این 2 پروتکول دیگر معرفی شد POP3 و IMAP. اولی قدیمی تر هست دومی بروز تر و با امکانات بیشتر. POP3 برای این طراحی نشده است که بتواند از هر ماشینی ورود کنی و به ایمیل ها دسترسی داشته باشی و فقط مربوط به همون session هست و خارج شی پاک شده و دفعه بعد بری چیزی وجود ندارد ولی در IMAP میتوانی فولدر بندی کنی و بعد از خروج و بازگشت فایل ها دست نخورد هستند. یک مورد جدیدی هم آمده است که یک ایمیل را به چندین بخش میتوانی تقسیم کنی مثلا میتوانی به طرف بگی ایمیل را بفرست ولی مثلا ویدیو را نفرست چون پهنای باند این اجازه را نمیدهد. ولی امروز از جیمیل و Hotmail استفاده میکنیم. دقت کن درست هست ما با HTTP با ایمیل کار میکنیم ولی خود اون سرور ها از SMTP برای رد و بدل کردن استفاده میکنند.

**جلسه هفدهم:**

DNS عین یک سرویس دایرکتوری هست که یک سری نگاشت به ما میدهد که مهم ترین آن نگاشت بین آدرس URL و IP هست. سیستم نام گذاری دامنه هست. سرور DNS یک ساختار سلسله مراتبی دارند به شکل پایگاه داده داریم و توزیع شده هست و مثلا ما از کل این سیستم پرس و جو میکنیم که مثلا دیجی کالا آدرس IP معادلش چی هست؟، دقت کن از UDP استفاده میکند با پورت 53 حتی اگر خطا هم رخ داد موردی ندارد چون بلاخره اگر خطایی رخ داد دوباره میزنیم. DNS یک نسخه کلاینت و یک نسخه سرور دارد. ماشین کاربر مثلا از طریق مرورگر میاد برنامه کاربردی نسخه مشتری را اجرا میکند و میگوید مثلا من دنبال دیجی کالا هستم بعد مشتری جواب دریافت میکند که شامل آدرس IP هست و بعد مرورگر ما میرود و یک اتصال TCP برقرار میکند. دقت کن در حین برقراری اتصال به آدرس IP نیاز داریم بعد اتصال دنبال IP نیستیم. تو لایه انتقال پورت 80 را میگوید و در هدر شبکه آدرس IP را قرار میدهد و مسیر را طی میکند تا به آن برسد. یک اسم کانونی داریم که اسم کامل ما هست و یک نام مستعار داریم که مثلا ساده شده اسم اصلی هست و دقت کن میتوانیم چندین اسم مستعار داشته باشیم. دقت کن از سمت کاربر فقط اسم مستعار درخواست میشود. کاربر با برنامه DNS مستقیم ارتباط ندارد و یک سرویسی در لایه کاربرد است. به سرور ایمیل هم نام مستعار اختصاصی میدهیم تا کاربر به جای اسم کانونی صرفا نام مستعار ساده را صدا بزند. توزیع بار: وقتی سر یک سرور خیلی شلوغ هست مثلا دیجی کالا که با یک وب سرور کارساز نیست و باید چندین وب سرور داشته باشد پس باید یک مکانیزمی باشد که ما بدون اینکه آدرس هر 50 تا را بدانیم بتوانیم به وب سرور دسترسی داشته باشیم و همچنین این مکانیزم همه بار را به یک سرور اختصاص ندهد و یک توازن باری وجود داشته باشد. پیچیدگی شبکه ها به ما ربطی ندارد و ما شفافیت میخواهیم همه این کار ها توسط DNS صورت میگیرد. دقت کن وقتی جواب DNS میاد مرورگر ما معمولا اولین آدرس معرفی شده را برمیدارد دقت کن این آدرس معرفی شده را هعی چرخیده میشود، نسبت به افراد مختلف که دنبال دیجی کالا هستند و همه به یک سرور اشاره نمیکنند و بار پخش میشود و کاربر ها اصلا متوجه این نمیشوند. شرکت های توزیع محتوا یا CDN ها، مثل ابراروان. دقت کن این انتقال پیام بین DNS و مرورگر ما توسط UDP شکل میگیرد. دقت کن DNS متمرکز در یکجا نیست مشکلات هم در جزوه هست. دقت کن هر موقع یک برنامه در مقیاس بزرگ کار نمیکند یک ساختار سلسله مراتبی پیشنهاد میکنیم. DNS به صورت عموما 3 لایه دارد: ریشه، دامنه سطح بالا و مرجع یا معتبر. مشتری اول سراغ یکی از سرور های ریشه میرود تا بفهمد دیجی کالا کجاست، دقت کن یکی از سرور های ریشه، متمرکز نیست، بعد میگوییم دنبال ای پی فلان هستیم اگر داشت که هیچ اگر نداشت مثلا میرویم سراغ دامنه سطح بالا .COM، سراغ یکی از آنها باز میرویم اگر داشت که هیچ، اگر نداشت سراغ سرور های مرجع مربوط به دیجی کالا میرود. از بالا به پایین حرکت کردیم تا به جواب رسیدیم. اول سراغ سرور DNS محلی میرویم تا ببیند در کش خودش دارد یا نیاز هست که ما به DNS سطح بالا تر برویم اگر در کش خودش داشت که هیچ.

دقت کن DNS محلی میاد در کش خودش ذخیره میکند مثلا فلان آدرس IP را بعد به میزبان درخواست کننده میفرستد و درون خود ذخیره میکند. از DNS سطح بالا نداشت میگوید من IP دیجی کالا را ندارم ولی مرجع آن را میشناسم و ما را هدایت میکند به مرجع. 2 نوع پرس و جو داریم، بازگشتی و تکرار شونده. رفتار DNS محلی با DNS های ریشه و سطح بالا و مرجع از نوع تکراری هست یعنی میگیم فلانی این را داری؟ نه بعد دوباره تکرار میکنیم. اما رفتار میزبان با DNS محلی به صورت بازگشتی هست. دقت کن میتوانیم کلا بازگشتی داشته باشیم یعنی به صورت زنجیر وار در ادامه همدیگر درخواست میکنند. یکی از مزایا همه بازگشتی این هست که هر لایه میتواند کش کند یعنی مثلا ریشه در کش خود دارد و دیگر نیاز نیست برای نفر بعدی چیزی خواست برود تا سطح مرجع، این مزیت اول. مزیت دوم این هست کاهش تاخیر به خاطر پرس و جو های راه دور هست.

بدی این روش این هست که سر DNS ریشه و سطح بالا خیلی شلوغ میشود. کش باعث بهبود کارایی میشود و کاهش تاخیر و کاهش حجم ترافیک داریم واسه همین دوست داریم در DNS محلی نگهدارد. دقت کن وقتی DNS ها بروز میشوند باید کش هم بروز شود دقت هم کن کش کردن 2 روز معتبر هست و بعد از 2 روز باید درخواست بزنیم از مرجع که ببینیم فلان IP هنوز معتبر هست یا نیست. رکورد هایی که در پایگاه داده DNS ذخیره میشوند RR نام دارند، مثل TTL که زمان حیات اون رکورد هست یعنی مثلا آدرس IP دیجی کالا را به شما میدهیم منتها برای 2 روز بعد 2 روز دوباره بیاد درخواست بده. اگر یک سرور DNS برای یک نام میزبان خاص سرور مرجع باشد آنگاه شامل یک رکورد نوع A برای نام میزبان خواهیم داشت. دقت کن حتی اگر سرور مرجع هم نباشد ممکن است این A در ذخیره ساز نهان خودش باشد. وقتی در حین ترجمه به آدرس IP به یک سروری میخوریم که مرجع نباشد، یک رکورد از نوع NS دارد که نام سرور هست برای اون دامنه که متعلق به اون میزبان هست همچنین یک رکورد A دارد که آدرس اون NS هست دقت کن نه آدرس اون چیزی که ما دنبالش هستیم. دقت کن شناسه به این درد میخورد: ما یک پرس و جویی میکنیم و سرور پاسخ میدهد شناسه ای که فرستادم را برمیگرداند در پیام جواب و ما میفهمیم این برای کدوم پرس و جو ما بوده است. پرچم ها به سرور اعلام میکنیم مثلا میخواهیم به صورت بازگشتی باشد یا تکراری. دقت کن یک پرسش میتواند چندین پاسخ داشته باشد. 6 تا فیلد هدر داریم. 12 بایت اول بخش سرآیند هست. یک فلگ داریم به نام پرس و جو/ پاسخ که اگر صفر باشد پرس و جو و اگر پاسخ باشد 1 میشود. پرچم مرجع هر وقت یک باشد یعنی برای اون جستجو که کردی من خود مرجع هستم و دیگر لازم نیست بری زیر مجموعه های پایین تر. پرچم درخواست بازگشتی وقتی که کلاینت از سرور بخواهد این درخواست به صورت بازگشتی باشد و به صورت تکراری نباشد که ریشه بگوید سلام من ریشه هستم برو سراغ دامنه سطح بالا و ... ، البته اگر در دسترس باشد بازگشتی.

2 ساعت اول دیده شد.

اگر نام میل سرور را خواستی از MX استفاده کن. وقتی نام کانونی را خواستند آدرس ای پی را هم نیز در بخش اضافی میفرستیم چون وقتی نام را خواسته قطعا IP هم میخواهد. DNS ها این اطلاعات را از کجا آورده اند؟ چطوری اطلاعات درج میشود؟ در یک data base مرجع. دقت کن وقتی سراغ TLD برود که او را بشناسد یعنی در کش خود داشته باشد اگر نداشت باید برود سراغ ریشه. بعد TLD میبینه منبع مورد نیاز و نوعA را دارد و برای فرستند میفرستد یعنی آدرس مرجع DNS، دیجی کالا را به ما میدهد و DNS محلی مشتری میداند که باید برود سراغ DNS مرجع و اون چون مرجع هست میداند.

**جلسه هجدهم:**

معماری همتا به همتا: این معماری خود مقیاس پذیر هست. توزیع فایل داریم که یک برنامه کاربردی یک فایل را از یک مبدا واحد به تعداد زیادی همتا توزیع میکند از طریق یک پایگاه داده توزیع شده، و به دنبال زمان توزیع هستیم یعنی یک کپی از فایل را به N تا همتا میخواهیم ارسال کنیم و هیچ گلوگاهی در هسته اینترنت نیست و فقط در شبکه های دسترسی گلوگاه داریم همچنین همتا ها تمام پهنای باند خود را به همین کار اختصاص داده اند. 2 نکته در معماری کلاینت سرور باید در نظر گرفته شود؟ 1. کی میفرستد؟ سرور به N تا مشتری و جمعا N\*F انتقال میدهد F طول اون فایل هست. زمان دانلود اگر از حداقل کوچک تر بود جواب دقیقا برابر با حداقل همین هست اگر بزرگتر بود جواب میشود همان زمان دانلود چون ما دنبال بزرگترین نرخ دانلود هستیم، زمان تاخیر کران پایین هست. برای N ها بزرگ همون فرمول اول هست و مقیاس پذیر نیست و به صورت خطی افزایش پیدا میکند یعنی 1000 تا مشتری بیاد 1000 برابر تاخیر بیشتر میشود. بر خلاف کلاینت سرور همتا به همتا تاخیر کمتری دارد، بر خلاف قبلی 3 نکته داریم: 1. کی باید اپلود کند؟ اول سرور حداقل یکبار باید بفرستد چون دفعه اول هیچ کسی ندارد، در فرمول همتا به همتا چرا بزرگتر مساوی گذاشتیم؟ چون شبیه به ذخیره و هدایت داریم و اول تمام بیت ها باید برسند بعد به سمت اون یکی همتا اپلود شود اگر این شکلی نبود و همون لحظه ارسال میشد این تاخیر اضافی به وجود نمی اومد. چرا مقیاس پذیر هست؟ N تا F داریم و N تا هم اپلود کننده داریم که با همدیگر میروند و مقیاس پذیر میشود. همتا به همتا خالی از سرور نیست، ما اول به سرور تورنت وصل میشویم بعد او ما را میبرد سراغ یک ردیاب بعد ردیاب لیست همتا ها را به ما میدهد و همتا جدید به صورت مداوم باید به ردیاب اطلاع دهد که حضور دارد، همتا جدید باید تلاش کند با همتا های معرفی شده توسط ردیاب اتصال TCP برقرار کند. دقت کن کاری به دانلود و اپلود ندارد ولی به همه اتصال برقرار میکند. یک سری تصمیم گیری نیاز داریم؟ 1. کدام یک از تکه هایی که نداریم را از همسایه ها درخواست کنم؟ 2. به کدام از همسایگان تکه های درخواست شده را بفرستد؟ جواب اولی میشود ابتدا کمیاب ترین را از اون همسایه ها بخواهیم چون بقیه که همیشه هستند ولی این کمیاب ها شاید همیشه نباشند. دوم به کدوم ها باید بفرستیم؟ 4 نفر فقط اونا کی هستند؟ اونایی که بیشترین سرعت ارسال بیت به من را دارند و به اصطلاح به آنها نا مسدود گفته میشود. و هر 10 ثانیه یکبار در این لیست تجدید نظر میشود. دقت کن هر 30 ثانیه یکبار من یک همتا را به صورت تصادفی میاورم مزیت این کار این هست که کسی که تازه وارد هست و هنوز چیزی اپلود نکرده است هم بیاد توی بازی، در واقع میشود 5 تا همسایه. بیت ریت های بالا با هم و بیت ریت های پایین هم با هم و این عدالت برقرار میکند. جدول درهم سازی توزیع شده: سوال اصلی این است که کدام فایل ها را چه کسانی دارد؟ دیتا بیس متمرکز مقیاس پذیر نیست و باید توزیع شده باشد و داده ها باید بین همتا ها توزیع شود. کل دیتابیس در همه ماشین ها ذخیره نمیشود در واقع هر همتا زیر مجموعه کوچکی از این رکورد ها را دارد و تمام رکورد ها اصلا در ماشین ما جا نمیشود که بخواهیم قرار دهیم. دقت کن مالک کسی هست که اون رکورد مورد نظر را دارد و مسئول کسی هست که میداند چه کسانی هستند که مالک هستند یا اون رکورد ها را دارند. و همتا از مسئول IP مالکان را میگیرد. DHT مشخص میکند مسئول این کلیدی که دنبالش هستیم کی هست با استفاده از درهم سازی. در واقع ما از کلید به همتای مسئول میرسیم. دقت کن خروجی درهم ساز درست هست که تصادفی هست ولی هر بار خروجی یکسان دارد، ولی این رویکرد خیلی مسخره هست اینکه ما خروجی کلید را برای همه همتا ها بفرستیم خوب سربار دارد. کلید را از DHT دریافت میکنیم، این روش بهتر هست: هر ماشینی که در این سیستم هست یک فرآیندی را دارد اجرا میکند، هر کدام از اینها یک ID دارند که سعی میکنیم یکتا باشد و توسط DHT تولید میشود (البته میتواند تکراری هم داشته باشیم)، ما روی کلید و ایدی صحبت میکنیم ما به فرآیند ها ایدی و به ایتم های داده کلید اختصاص میدهیم جفت توسط DHT و هر دو درهم سازی شده، یک عدد صحیحی N بیتی است. اگر عدد خیلی بزرگ باشد احتمال تکراری بودن خیلی کم هست. خلاصه: دنبال یک موزیک هستیم موزیک را به عنوان کلید اصلی میدهیم به DHT و DHT به ما یک عدد میدهد از روی این عدد به یک ایدی میرسیم به ترانه یا مسئول کلید که میداند کیا این فایل را دارند این صورت مسئله است نگاشت بین کلید و ایدی مسئول صورت مسئله ما هست بعد از ایدی مسئول IP آنرا هم پیدا میکند. روش: مگه تمام کلید ها تو یک بازه نیست؟ همه ایدی ها هم مگه تو همین بازه نیست؟ کدام ایدی از بین ایدی های موجود مسئول آن هست؟ اونکه بهش نزدیکترین هست چون خیلی طول میکشد ما تک به تک بررسی کنیم چون اختلاف بین این 2 عدد ها زیاد هست. خوب حالا ما یک ایتم داده همون کلید را داریم کدام ایدی مسئول هست؟ نزدیکترین. نزدیکترین گره بعدی منظور ما هست دقیقا بعدی و قبلی نباید باشد. دقت کن حلقوی هم هست یعنی نزدیکترین بعدی 14 مثلا 15 نبود 0 هم نیست پس 1 هست چرخش میکند یا نزدیکترین بعدی 15 کی هست؟ خودش. حالا آرزو از کجا بفهمد کی مسئول هست؟ بین 100 میلیون بگردد؟ خوب اینکه مقیاس پذیر نیست پس باید از دیتابیس های توزیع شده بپرسد. باید جدول های درهم ساز چرخشی داشته باشیم در ساده هر کسی یک بعدی دارد ولی در میان بر یک سری میان بر بین گره ها داریم.

شبکه overlay یعنی شبکه ای از فرآیند ها و لینک های در حال اجرا هست و یک لینک واقعی نیست و گره ها فرآیند ها هستند و اتصالات منطقی مثل TCP داریم. و در این سیستم به طور میانگین N/2 پیام فرستاده میشود چرا که هر همتا از همسایه قبلی و بعدی خودش فقط آگاه هست. یک حالت دیگر این هست که تعداد پیام را خیلی کمتر کنیم به طوری که فقط یک پیام بشود ولی همسایگان زیادی برای ردیابی وجود دارد یک نوع معامله هست یا N/2 پیام و همسایه قبلی با بعدی یا یک پیام و کلی همسایه ردیابی شده داشته باشیم. البته در DHT یک موازنه ای بین این 2 هست یعنی به طوری که هم تعداد پیام ها قابل قبول باشد هم تعداد همسایگان ردیابی شده قابل قبول باشد. اگر N ما 160 بیتی باشد تعداد گره ها تا 2^160 میتواند باشد. N تعداد همتا و n بیت ها هستند. یعنی اگر n 160 باشد N میتواند تا دو به توان 160 باشد. یکی از اشکالات این سیستم ها این هست که ممکن است یک همتا خاموش کند و برود و مشخص نیست کی رفته است برای حل مشکل ورود و خروج بی خبر همتا هر همتا دو نفر بعدش خودش را نگه دار به همراه آدرس IP مثل 8 باید 10 و 12 را داشته باشد. و اگر 12 برود ما صدا میکنیم وقتی جواب داد از 10 میپرسیم دو تای بعدی شما کی بود میگه 15 میگیم که پس از این به بعد دو تای بعدی من میشود 10 و 15 تا سیستم قابل اطمینان بیشتری داشته باشد. دقت کن فایل را همه با هم موازی ارسال میکنند و وقتی میخواهیم اپلود کنیم هر کسی یک تکه هایی را شروع میکند به اپلود کردن دقت کن در تست اپلود سایر 3 برابر و دو برابر اپلود دستگاه اولی هست و بهترین زمان باید تکه ها به نوعی باشد که هیچکس بیکار نباشد واسه همین دنبال ضریب 9 هستیم.

**جلسه نوزدهم:**

شبکه های توزیع محتوا یا CDN ها، دقت کن ویدیو ها ضبط شده هستند و طبق درخواست پلی میشود. مهم ترین ویژگی یک ویدیو این هست که حجم آن زیاد هست پس مهم هست گذردهی مفید انتها به انتهایی که به کاربر تحویل بدهیم چه قدر باشد پس حداقل باید به اندازه متوسط نرخ بیت ویدیو فشرده باشد تا دچار مشکل در دیدن نشویم اگر دچار مشکل شدیم از سمت کاربر هست بخاطر گلوگاه ها و گرنه سرور ها که قوی هستند. البته یک تکنولوژی اومده خودش خودکار بر اساس فید بکی که از پهنای باند ما میگیرد خودش ترافیک را تضمین میکند. و بافر کردن مشکل جیتر را حل میکند. دریافت و ذخیره و بافر کردن بعد پلی کردن این کار مرتب صورت میگیرد از سمت کاربر. دقت کن اول چند لحظه بافر میشود به آستانه که رسید شروع به پلی کردن میشود. HTTP ساده مشکلش این هست که با نرخ بیت ثابت شروع میکند به دانلود کردن و این خوب نیست چون پهنای باند ما بالا و پایین میشود. DASH میاد یک فایل مانیفست یا بیانیه میده به کاربر و میگوید چه ویدیو هایی با چه URL هایی و با چه نرخ بیت ریتی اینجا ذخیره شده است. روش دوم در استراتژی امروزه بهتر هست زیرا که اگر یک ویدیو را فقط ایرانی ها میبینند و بنگلادشی ها نمیبینند نیازی ندارد CDN نزدیک به ما در بنگلادش این را داشته باشد و اگر یک بنگلادشی خواست برود بگیرد چه از CDN نزدیک به خودش و چه از مخزن مرجع. و CDN اینجوری عمل میکند که وسط درخواست ما در خواست را میگیرد و ما را به ویدیو مورد نظر میبرد و نمیگذارد ما به مخزن اصلی پر لایک برویم و اینجا نقش DNS هست. دو سوال مطرح هست سوال دوم مکانیزم interception یا راه بندی و تغییر جهت چه شکلی هست و تغییر جهت یا redirect یک در خواست برای یک ویدیو چگونه هست؟ از DNS استفاده میکنند و وسط راه درخواست ما را میگیرد. DNS هم که میدونی وقتی دنبال آدرس باشد اول سراغ محلی میرود بعد سراغ TLD مثل .COM بعد سراغ DNS مرجع سایت پر لایک. برای انتخاب این خوشه ها هم استراتژی های مختلفی داریم مثلا یک استراتژی ساده این هست که اون خوشه ای که از نظر جغرافیایی به ما نزدیک تر هست. چطوری یک سری LDNS یا local DNS داریم که حدودی میدانند این IP کجاست نه اینکه دقیقا IP ماشین ما را بدانند و ما کجا هستیم. این بعضی مواقع خوب نیست چون شاید از لحاظ جغرافیایی نزدیک تری باشد ولی از لحاظ شبکه ای نزدیک نباشد یعنی مثلا از کلی روتر دور بزنیم بریم به صرف اینکه از لحاظ جغرافیایی نزدیک باشد و نکته بعدی این هست که ما از LDNS های راه دور استفاده میکنیم و این LDNS درست جای ما را نشان نمیدهد یا نوسانات ترافیکی بخواهد تغییر کند همه اینها باعث میشود این استراتژی ساده خوب نباشد. پس باید مرتب اندازه گیری کنیم چطوری؟ از CDN میخواهیم که هر یک مدت یکبار بیاد یک درخواست به همه LDNS های ایران بفرستیم یا اگر در خارجیم در خارج. مثلا پینگ کن آنرا. پس دقت کن از یک سرور نمیگیریم از یک خوشه نمیگیریم از یک دیتا سنتر هم نمیگیریم. چطوری حل میکردیم با CDN ها دیگه مشکل اول چی ؟ میایم کد گذاری مختلف و کیفیت های مختلفی برای کاربران با پهنای باند های مختلف میسازیم که با بیت ریت های مختلف کاربران بتوانند ویدیو ها را ببینند. دقت کن کد گذاری یکی نیست و چند تاست 2 این فقط برای مشکل اول هست نه برای مشکل دوم مشکل دوم با توزیع محتوا یا CDN حل میشود. اینکه بتوانیم برای ترافیک جریان ویدیو اولویت بگذاریم خیلی خوب هست و در بحث تضمین خدمت خیلی خوب هست همین کار را میکنیم و فقط باعث تسریع انتقال میشود ولی ربطی به مشکلی ندارد چون مشکل کاربر با کمبود پهنای باند هست و ربطی به سرعت ندارد که. برای ویدیو استریم روش peer to peer مناسب نیست.

**جلسه بیستم:**

برنامه کاربری داخل خانه هست چون داخل فرآیند هست و مسائل لایه های پایین تر مثل دست تکانی TCP نشان داده نمیشود. دقت کن فقط در UDP میایم آدرس سرور نیم و پورت را قرار بدهیم در TCP اصلا نمیخواهد چون مشخص هست ولی در UDP باید به ازای هر پیام این اطلاعات را بفرستیم. دقت کن کلاینت باید سرور را بشناسد پس باید IP سرور را در فایل کلاینت بگذاریم ولی سرور نیازی به IP ندارد از هر جا میتواند دریافت کند. فرق TCP با UDP این هست در UDP n تا دستگاه به سرور وصل هست یا کلاینت ها هستند ولی در TCP این شکلی نیست بین یک سرور خاص و یک کلاینت خاص فقط و فقط یک اتصال برقرار هست پس دیگر نیازی به آدرس نیست چون مشخص هست. ولی دقت کن دفعه اول که ما اتصال برقرار نکردیم و سرور ما را نمیشناسد پس چطور پیام منتقل کنیم؟ در ارتباط اول میایم یک سوکت یا پورت استقبال میگذاریم دفعه اول و فقط یکبار تا از اون در میگوییم بیا با ما یک اتصال برقرار کن و بهش یک پورت جدا اختصاص میدهد. پس 3 سوکت شد 1 سوکتی که دفعه اول بود و همسایه ها و ما ارتباط برقرار کردیم 2 سوکت خود ما و 3 سوکت همسایه. دقت کن که SERVERSOCKET یک چیز همگانی و اولیه هست و سوکت استقبال نیست بلکه CONNECTIONSOCKET سوکت استقبال هست. و با متد accept منتظر میماند یکی وصل شود. و حواست هم باشد اولش آدرس سرور و پورت را میدهیم بدون آن نیست. دقت کن اول ما درخواست connect داریم بعد accept از سمت سرور میاد در کد اول نوشته نشده است. دقت کن شماره پورت سوکت مشتری را به چیزی bind نمیکنیم بلکه میگذاریم خود سیستم عامل بدهد. دقت کن دو تا سوکت داریم یکی server socket که مال استقبال هست یکی connection socket که به ازای هر کلاینت در حلقه while گوش میدهیم و میپذیریم. دقت شود حداقل باید یک کلاینت را بپذیریم. SOCK\_STREAM میگذاریم یعنی منظورمان TCP هست و وقتی AF\_INET میگذاریم منظورمان IPV4 هست. دقت کن listen اندازه بافر نیست. دقت کن به اسم قبل مساوی دقت نکنی آن هر چیزی میتواند باشد. دقت کن MYPORT درست هست که اسمش این هست ولی برای سرور پورت هست و خود کلاینت به 80 bind شده است.

**فصل سوم، لایه انتقال:**

**جلسه بیست و یکم:**

کمترین وظیفه هر پروتکول این هست که پیام را از کاربر بگیرد یعنی مالتی پلکسینگ و تحویل آن به کاربرد مورد نظر در مقصد یا دی مالتی پلکسینگ را بتواند انجام بدهد. و یک ارتباط منطقی ایجاد میکند بین دو کاربر انتهایی یا برنامه کاربردی یعنی درست هست که ممکن است یکی از این برنامه ها این کشور و یکی اون کشور باشد ولی ما اصلا به چیز ها کاری نداریم و فکر میکنیم یک ارتباط مستقیم بین این 2 هست و نگاه انتزاعی یا سطح بالا داریم. در host ها پیاده سازی میشوند و هیچ چیزی در روتر ها نیست و اصلا به سرآیند اینها دقت نمیکند روتر ها و این لایه فقط در سیستم های انتهایی هست. در لایه شبکه اون قطعه ای که از لایه انتقال را که گرفتیم و پیام لایه کاربرد در آن هست را کپسوله میکند در یک دیتاگرام یا یک IP و هدر IP بهش اضافه میشود و سوییچ ها با این IP ها کار دارد دقت کن لایه شبکه وظیفه رساندن بسته ها را از مبدا به مقصد دارد و اصلا کاری به سوییچ ندارد و فقط از مبدا به مقصد حرکت میکند بعد به لایه بالا که انتقال هست تحویل میدهد. در مورد کنترل خطا یا قابلیت اطمینان یا امنیت این امکان وجود دارد مثلا تینا نامه را میگیرد میبیند شبکه نا امن هست مثلا میاد یک سری اطلاعات اضافه در آن بسته میگذارد که بهش checksum میگوییم که میتوانیم تشخیص بدهیم که ببینیم خطا رخ داده است یا نه یا پیام ها را شماره گذاری کنیم که اگر اون وسط یکی نیومد بفهمیم یا یک کپی از آن را در مبدا نگه داریم و در مقصد از روی checksum میفهمد که خطایی رخ داده است یا نه یا از روی شماره میفهمد همه پیام ها رسیده است یا نه، دقت کن حتی اگر امنیت نباشد لایه انتقال میتواند این کار ها را بکند یا رمز نگاری بکند اصلا. تاخیر یا پهنای باند را لایه انتقال را نمیتوانند کاری کنند گذردهی هم نمیتوانند. UDP به قطعات خودش دیتاگرام میگوید مثل بسته های IP اما TCP از این استفاده نمیکند و برای اشتباه نشدن از بسته یا قطعه استفاده میکنیم برای هر دو. از کلمه دیتاگرام هم در سوکت میفهمیم که منظور UDP هست. UDP اتصال منطقی با رزرو بافر اینا ندارد. اداره پست یا IP اداره منطقی بین میزبان ها را برقرار میکند یعنی از در این خانه در کرج تا در خانه مقصد در مشهد و لایه انتقال میاد ارتباط برنامه های کاربردی بین خانه کرج و مشهد را برقرار میکند دقت کن داخل خونست مثلا ارتباط فامیل. IP میزبان به میزبان هست ولی تینا و یوسف ارتقا میدهند از اپلیکیشن به اپلیکیشن. UDP تشخیص خطا دارد یعنی میفهمد که خطا دارد ولی کاری برای رفع آن نمیکند. TCP عدالت دارد یعنی به همه عادلانه سهم برابر از ترافیک را میدهد یعنی چه به HTTP و چه به FTP و چه به SMTP سهم برابر میدهد دقت کن هر 3 اینها روی TCP هستند، پس اگر عدد دادن بین همه تقسیم کن چون ترافیک برابر هست. TELNET هم روی TCP قرار میگیرد. دقت کن خانه ها چند تا در دارند چرا چونکه ممکن هست ما 3 تا ارتباط مجزا با عمو زاده ها داشته باشیم و هر کدام پورت مخصوص دارند پس سوکت مخصوص دارند و ارتباط جدا و در مخصوص به خود را دارند. برای UDP یک سوکت کافی هست ولی برای TCP با یک سوکت نمیشود و اتصال های مختلفی دارد و هر سوکت یک اتصال دارد و بافر و متغیر های کنترلی برای هر سوکت داریم. و تینا میاد اون پیام را از یکی پسر عمو میگیرد و شماره مخصوص اون سوکت یا همان شماره پورت به سر پیام اضافه میکند، چرا در UDP آدرس خودمان و پورت خودمان و IP خودمان میگذاریم ؟ چون پسر عموی ما نیاز به آدرس برگشت دارد تا بتواند به ما جواب بدهد. چرا در TCP شماره پورت هم میخواهیم ؟ زیرا ممکن است ما دو اتصال یکسان داشته باشیم با یک IP یکسان و نتوانیم این 2 را تفکیک کنیم پس باید پورت هم باشد بتوانیم تفکیک کنیم حالا میگی فقط با پورت کار کن خوب اینجا هم از شانس ما ممکن هست دو IP متفاوت پورت یکسان داشته باشند پس اون 4 تایی نیاز هست و اگر یکی از اینها نباشد اتصال ها با هم قاطی میشود.

ولی در UDP اصلا اینها مهم نیست یک در هست همه میان تو.

**جلسه بیست و دوم:**

برنامه های چند رسانه ای ترجیح میدهند که از UDP استفاده کنند چرا که به دنبال پهنای باند بالا هستند ولی TCP میگوید کنترل ازدحام و محدود میکند ترافیک را، یا مثلا TCP عادت دارد هعی فایل تکراری بفرستد و این کند میکند. واسه همین هست که DNS از UDP استفاده میکند چون سریع هست ولی TCP 2RTT طول میکشد و UDP یک RTT طول میکشد و حاضر نیستیم DNS را کند کنیم چون شاید یک پیام گم شود. UDP میتواند تعداد اتصال بیشتری برقرار کند چون سرباری ندارد و به دنبال دست تکانی نیست. RIP هم روی UDP هست حتی اگر یک جدولی گم شود از اطلاعات روتر ها مهم نیست 30 ثانیه بعد درست میشود. دقت کن UDP بسته باشد سوییچ میکنیم به TCP. مدیران بهتر هست روی UDP بشینند تا سریع باشند و سریع عمل کنند. دقت کن که خود UDP میتواند کنترل خطا کند مثلا در لایه کاربرد همانطور که وقتی IP نا امن بود TCP درستش کرد البته باید وقت بگذاریم برای برنامه نویسی مثل TCP راحت نیست بگیم تینا این بسته را سالم برسان. هدر UDP 8 بایت هست ولی واسه TCP 20 بایت هست. چرا باید طول داشته باشیم؟ چون اندازه ها متفاوت هست و وقتی بیت ها بیان ما نمیدانیم تا کجا هستند. با افزونگی جمع میتوان خطا را پیدا کرد دقت هم کن فقط یک خطا را میتوان تشخیص داد. اگر 48 بیت دادن 3 تا 16 بیتی بکن بعد 2 تا را با هم جمع کن بعد حاصل را با اون یکی جمع کن. در جمع اگر کری نداشت که هیچی جمع معمولی اگر کری داشت: بچرخان بیار سمت راست یکبار دیگر جمع بزن بعد مکمل 1 کن. دقت کن چند تا 1 میفرستد جمع میکند و اگر AND حاصل 1 شد یعنی خطا ندارد صفر شد یعنی یکی خطا داشته است. چه لزومی دارد وقتی HUB BY HUB کنترل خطا داریم لینک به لینک دوباره در مبدا و مقصد هم کنترل خطا داریم؟ دو نکته. اولا تضمینی وجود ندارد همه لینک یا پیوند ها کنترل خطا را انجام میدهند و دو ممکن هست خطا در لینک ها نباید خطا در حافظه باشد. حتی اگر بسته درست منتقل شود ممکن هست خطا رخ بدهد و بسته از دست برود مثلا در لوپ گیر کند واسه همین برخی از وظایف خاص مثل کنترل خطا یا کنترل ازدحام و جریان باید انتها به انتها باشند. باید روی لینک ها هم کنترل خطا باشد چون زشت هست بسته خطا داشته باشد و همینطوری چندین لینک بگذرد و به مقصد برسد و در مقصد بفهمیم بسته خطا دارد.

**جلسه بیست و سوم:**

دقت کن در لایه شبکه نا مطمئن هستیم TCP میاد و آنرا مرتب و مطمئن میکند. اول فکر کن یک طرفه هست یعنی یکی میفرستد یکی میگیرد. دقت کن چی میشود که تغییر وضعیت میدهیم؟ یک event رخ بدهد، دو تا چیز را باید راجب event جواب بدهیم یک چه رویدادی باعث این انتقال میشود یا این رویداد چیست؟ دو در طی این رویداد باید چه کاری بکنیم و چه عملیاتی انجام بدهیم. Event میتواند RDT\_SEND باشد چرا چون همون چیزی هست که از بالا میاد پایین اگر چیزی از پایین به بالا بیاد رویداد هست ولی RDT-RECV هست. علامت نال بودن هم یعنی هیچ عملی یا رویدادی رخ نداده است. پروتکول هایی که برای کنترل خطا استفاده میکنیم ARQ نام دارند. به 3 قابلیت نیاز داریم یک تشخیص خطا مثل افزونگی های checksum، دو بازخورد از سمت گیرنده یعنی اینکه تینا اینور به تینا اونور میگوید NAK یا not acknowledge یعنی خطا رخ داده است و لطفا دوباره بفرست تینا جان، و سه ارسال مجدد دقت کن با RDT\_SEND حالت خود را عوض میکنیم چون رویداد بوده است و از بالا اومده و بازی شروع شد. دقت کن در RDT2.0 تا تکلیف بسته اول مشخص نشود بسته دوم را از بالا نمیگیریم و منتظر ACK یا NAK میمانیم. اگر ACK بود هیچکاری نمیکنیم. دقت کن اگر NAK اومد میریم تو مرحله ارسال مجدد UDT\_SEND دیگر جابجا نمیشویم و از بالا چیزی نمیگیریم اینقدر میفرستیم تا بلاخره ACK بیاد. گیرنده بازی شروع شد منتظر میشود تا از پایین چیزی بیاد و ما توسط اون افزونگی چک میکنیم اگر درست نبود NAK درست میکنیم میفرستیم اگر درست بود ACK درست میکنیم بفرستیم. خوابیدن یا بیدار بودن هم با یک سمافوری چیزی مثلا میسازیم. دقت کن یک مشکلی هست این هست که خود ACK یا NAK اشتباه شوند میتوانیم همین افزونگی را برای این 2 هم بگذاریم ولی این هم مشکل دارد چون تو لوپ بیفتیم و بسته های تکراری داریم چطوری بفهمیم بسته ها تکراری هستند؟ از شماره ترتیب استفاده کنیم تا اگر بسته شماره 2 دوبار دریافت شد و ما داشتیم آنرا بندازیم دور و بگیم ACK. چون بسته ها stop-wait هستند فقط 0 و1 را میفرستیم یکی رفت اوکی بود بسته بعدی 0 بعد بسته بعدی 1 همینطوری ادامه پس فقط با 1 بیت میتوانیم شماره گذاری کنیم. در RDT2.1 4 حالت داریم که یکیش منتظر بسته شماره 0 هستیم یکیش منتظر بسته شماره 1. گیرنده کلا منتظر از پایین هست در منتظر صفر بازی شروع میشود. چرا در گیرنده checksum میفرستیم؟ چون خود ACK هم ممکن هست غلط باشد. دقت کن در 2 حالت تغییر حالت نمیدهیم یک یا بسته خراب باشد یا شماره اشتباه گرفته باشیم و به فرستنده میگیم که متوجه شدیم که این را دریافت کرده ایم. در RDT2.2 میگوییم که در شبکه میتوانیم NAK نداشته باشیم و ACK را شماره دار کنیم اگر بگیم 1 یعنی درست هست و ACK تکراری معنی NAK میدهد یعنی اگر صفر دریافت کنیم یعنی NAK هیچ فرقی هم ندارند فقط فرق این هست شمارنده ACK شماره بسته سابق درست قبلی هست. ACK تکراری معنی NAK میدهد. در RDT3.0 امکان تلفات بسته را هم تشخیص میدهیم چون وقتی ما بسته را میفرستیم یا میخواهیم بدانیم درست فرستاده ایم یا اشتباه فرستاده ایم بهترین رویکرد این هست که به عهده فرستنده بگذاریم تلفات بسته را و تا یک حد زمانی مشخص نیامد بسته را دوباره ارسال میکنیم و تلفات احتمال هست. دقت کن نیاز به تایمر داریم چرا که اون حد زمانی را بدست بیاوریم. RDT3.0 یک پروتکول stop and wait یا بیت تناوبی هست. دقت کن منتظر ACK صفر هستی ولی ACK 1 میاد اینجا صبر کن فکر نکن مثلا لاست شده هست پس برای پکت صفر عجله نکن چون این ACK تکراری و حاصل تاخیر هست چون ACK سرگردان هست.

**جلسه بیست و چهارم:**

ممکن هست پروتکول های stop and wait بخاطر تاخیر زیادی که تا خبر سالم یا خراب رسیدن بسته بیاد کلی صبر کنند و بهره وری میاد پایین و با استفاده از pipeline سعی میکنم این مشکل را حل کنیم و این دسته از پروتکول ها را معرفی کنیم. RTT همان تاخیر رفت و برگشت هست و دو برابر سرعت انتشار هست و دقت کن هر چه قدر فاصله بیشتر یا سرعت بیشتر اوضاع بهره وری در آن بدتر میشود. در سوال مشخص هست که تاخیر میشود 8 میکرو ثانیه ولی 30 هزار ثانیه طول میکشد تا برود و خبرش بیاید یعنی این همه مدت بیکار ایستاده ایم. چرا سرعت بیشتر شود اوضاع بدتر میشود؟ چون مخرج کسر L/R بزرگتر میشود و زمان مفید کم میشود چرا هر چه قدر فاصله بیشتر باشد اوضاع بدتر میشود؟ چون باعث میشود RTT و زمان انتظار الکی بیشتر شود. در معماری خط لوله هم دقت کن که کنترل ازدحام و جریان به ما نمیگذارند اندازه پنجره ارسال از یه حدی بیشتر شود پنجره ارسال در شکل اون 3 تا آبی ها هستند. وقتی بگذارند 3 تا بفرستیم بهره وری 3 برابر میشود چون 3 تا از اون L/R هست دقت کن همیشه اینطوری نیست مثلا برای L/R خیلی کوچک بود یا RTT خیلی بزرگ بود در اعداد 2 3 5 میتوانی ضرب کنی و همین برابر راندمان را چند برابر میکند. دقت کن در معماری خط لوله همانطور که شماره ترتیب برای بسته ها داشتیم باید برای ACK ها هم داشته باشیم و دقت کن 0 و 1 نیست و محدوده شماره در خط لوله باید بیشتر باشد اصلا نمیتواند صفر و یک باشد چونکه معنی نمیدهد. TCP ترکیبی از GBN و SR هست. N اندازه پنجره ارسال هست. دقت کن ما دنبال پروتکول مطمئن هستیم قانون اصلی این هست که بسته ها باید سالم و مرتب به لایه کاربرد برسند خوب وقتی بسته شماره 9 آمده است بعد بسته شماره 11 آمده باشد و 10 نیامده باشد باید چیکار کنیم؟ باید بافرش کنیم تا بسته شماره 10 بیاد بعد از آن 10 را میفرستیم به بالا بعد 11 را میفرستیم TCP و SR اینکار را میکنند برای همین هست که آنرا پیچیده میکند. GBN چون ساده هست میگوید درست هست که تو سالمی ولی چون ترتیب تو درست نیست حذف میکنیم TCP هرگز بسته سالم را دور نمیندازد. Go back یعنی برو به عقب تا جایی که درست به ترتیب بود یعنی 11 اومد ACK 9 اگر 12 اومد ACK 9 تا فرستنده بفهمد. SR میگوید فقط 10 را بفرست نه اینکه از 9 به بعد را بفرست دور. ACK در GBN تجمعی است یعنی اگر ACK 2 اومد یعنی 0 و 1 و 2 درست آمده است. یعنی مثلا اگر ACK 8 در راه باشد ولی ACK 9 بیاد میفهمیم 8 هم رسیده است. چرا بهش میگیم پنجره لرزان؟ چون با آمدن هر ACK یکی از آبی ها به سبز و یکی از قرمز ها کم و یکی از زرد ها آبی میشوند انگار میلغزیم و به جلو میرویم. در RDT SEND در شرط اول چک میکنیم اگر هنوز پنجره جا داشته باشد باز میفرستیم اگر نه یعنی هیچ زردی وجود ندارد همینجوری پنجره جلو آمده ولی هیچ ACKیی نیامده هست یعنی جا ندارد، دقت کن تایمر فقط یکبار روشن میشود و برای بسته اول شروع میکند به شمردن و برای سایر بسته ها نیست. اخرش یکی به آبی ها اضافه میکنی در آخر IF اول. یک رویداد دیگر این هست که GBN اگر 10 نیاد 11 و ... را میندازد دور و میگوییم تمام آبی ها را مثلا از 9 به بعد را ارسال مجدد کن کی؟ زمانی که حس کردیم NAK شده است تایمر چرا استارت میکنیم؟ چون به بسته های قبلی امیدی نداریم که برسند و دقت کن یک انتظار مشخصی داریم که بعد یه مدتی نیومد دوباره بفرستیم بفهمیم لاست شده است. ممکن است یک ACKیی باشد که درست باشد ولی شماره آنرا ندانیم. دقت کن ما یک تایمر فیزیکی داریم و بقیه منطقی هستند و از روی اون یک جدول منطقی درست میکنیم. در گیرنده دقت کن درست باشد که هیچی میدهیم بالا و شماره مورد انتظار را یکی بیشتر میکنیم ولی اگر بسته مورد نظر ما نباشد میندازیم دور و میگوییم که همون ACK 9 که مورد انتظار ما هست میفرستیم مثلا 10 رسیده است ولی ما منتظر 9 هستیم، یک چیز پیش فرض هم دارد اینکه به صورت پیش فرض هر چی شد ACK چیزی را میفرستیم که منتظرش هستیم. اگر زرد داشتیم یا بافر کنیم از بالا وقتی میگیریم یا کاربرد را بخوابانیم تا چیزی نفرستد. دقت کن در مثال تصویری فرستنده و گیرنده اندازه پنجره 4 هست واسه همین هست که 0 تا 3 را فرستاده است ولی وقتی به 4 رسید صبر کرده هست وقتی ACK 1 آمد توانست 4 را بفرستد و دقت کن با وجود اینکه 3 و 4 و 5 رسیده است ولی فرستنده وقتی تایم اوت خورد و 2 جوابش نیامد بقیه هم که رسیده بودند دوباره میفرستد. واسه شماره ACK هم باید تعداد را 0 و 1 که هیچ بلکه به اندازه پنجره بعلاوه 1 بگیری یا N+1، چون N تا جواب نمیدهد چرا؟ 4 تا میفرستیم ولی خبری نمیگیریم، ACK 0 را میدهیم فرستنده که خیال میکند نرسیده است میاد دوباره از 0 ارسال میکند چی میشود؟ گیرنده که منتظر 0 بوده است میبیند که اینجا هم 0 وجود دارد و همین باعث میشود که فکر کند بسته تکراری به عنوان جدید فرض کند و گند رخ میدهد ولی اگر N+1 باشد ما چون منتظر 4 هستیم 0 ها را نمیپذیریم. چون گیرنده بسته خارج از ترتیب را قبول نمیکرد میومد از اون بسته که گم شده بود را از اول میگرفت میفرستاد از ادامه اون به بعد. در SR دیگر تجمعی نیست و به ازای دقیقا هر بسته باید ACK بدهی یعنی به صورت انتخابی تایید یا عدم تایید کن. و دقت کن اگر یک بسته نرسیده باشد فقط همان یکی را دوباره میفرستیم و تایمر را فقط برای همان یک بسته در نظر میگیریم. دقت کن اگر یک بسته را دریافت کردی یک دونه از اون آبی ها را سبز میکنی از اون به قبل را سبز نمیکنی چون تجمعی نیست. دقت کن فقط وقتی پنجره جلو میرود که send base آبی سبز شود اگر اون وسط یک آبی سبز شد پنجره را نمیبری جلو. در گیرنده هم دقت کن اگر بسته مورد انتظار ما هست اومد دیگر لازم نیست بافر شود مستقیم تحویل بالا بده بقیه رو هم به ترتیب بفرست برای بالا. یک بسته را هم اگر قبلا گرفته بودی و تایید کرده بودی و به بالا داده بودی با این وجود تو ACK را بده تا گیرنده هم متوجه شود. با وجود بافر ACK را میفرستد. تا وقتی هم سر پنجره جلو نرود نمیگذارد چیزی فرستاده شود. شماره پیام ها باید دو برابر پنجره ارسال باشد یعنی مثلا 4 تا بود به 8 تا شماره نیاز داریم. در TCP شماره رنج را خیلی بزرگتر از 2N میگیریم و یک سری حالات تصادفی داریم که به زودی حتی در کانکشن بعدی هم از این شماره ها استفاده نکنیم.

**جلسه بیست و پنجم:**

دقت شود که اتصال ها در اینجا فیزیکی نیست حتی مدار مجازی هم نیست چون متغیر های ما انتها به انتها هستند بر خلاف مدار مجازی که در هر روتر ذخیره میشدند. دقت شود که MSS حداکثر اندازه درون قطعه هست بدون وجود هدر TCP. Link frame که میشود پکت دیاگرام شبکه به همراه هدر و دنباله لایه پیوند بزرگترین بسته در معماری شبکه هست. segment ماکسیموم چیزی هست که میتواند درون شبکه مورد انتقال قرار بگیرد. MTU یعنی بیشترین میزانی که شبکه میتواند منتقل کند که هدر خود لایه شبکه هم درون آن هست. سرآیند TCP یا لایه انتقال را از خود بسته کم کنی MSS بدست میاد و توسط MTU تعیین شد. دقت کن ما بایت ها را شماره گذاری میکنیم نه بسته ها را. پنجره دریافت برای کنترل ازدحام هست و توسط این گیرنده به فرستنده میگوید اندازه بافر من اینقدر هست و بیشتر نفرست و عددی بین 0 تا 15 هست که بیشترین حد میشود 15 تا 4 بایت که میشود 6 بایت که 20 تا کنترلی و 40 تا داده اینا هست این مقیاس پنجره دریافت را میتوانیم توسط بیت های اختیاری تعیین کنیم. پرچم ها برای راه اندازی اتصال TCP استفاده میشود. دقت کن ACK فلگ وقتی یک باشد در اون 32 بیتی هم مشخص میشود که 1 هست. PSH کل قطعه را میگوید فوری هست و URG میگوید بخشی از این قطعه فوری هست. دقت کن گیرنده همینطور که داده میفرستد در هدر هم ACK را میفرستد چون دو طرفه هست و دو طرف میتوانند داده بفرستند. دقت کن گیرنده وقتی بخواد ACK بدهد میگوید ACK 1000 اگر ما قطعه اول که 0 تا 999 هست را فرستاده باشیم. باید یک کاری کنیم در TCP که از شماره های که در اتصال قبلی استفاده کردیم در اتصال جدید استفاده نکنیم تا ACK های سرگردان باعث خرابی نشوند. شماره ترتیب شماره اولین بایت هست. دقت کن وقتی لاست میشود چیزی نمیرسد به گیرنده که بخواهد اصلا چیزی بفرستد و دقت کن اگر یک بسته اون وسط لاست شد ولی بسته بعدی رسید نمیاد ACK 635 را بدهد چون اینجوری فکر میکند این بسته لاست شده هم رسیده است به گیرنده این پیام را میفهمد، پس باید ACK 335 بزنیم یعنی دوباره بفرستیم که بگوییم یکی اون وسط NAK شده است. اگر 3 تا ACK تکراری بیاد یعنی بسته لاست شده است. دقت کن یا تایمر زودتر تایم اوت میکند یا 3 تا ACK میرسد.

تخمین RTT و زمان انقضا که همان زمان تایم اوت هست و این زمان باید کمی بیشتر از RTTباشد. یک راه این هست که برای هر از گاهی یک sample از RTT میگیریم از ارسال های قبلی هر چند وقت یکبار؟ به اندازه یک RTT، درست هست خود این تایم متغیر هست ولی به ازای آخرین نمونه RTT این را در نظر بگیر و تخمین تا میانگین بگیری این میانگین وزن دار هست. ما بیشتر شبیه گذشته اخیر عمل میکنیم نه چند سال پیش پس وزن آخری ها بیشتر هست و مهم تر هست به این الگوریتم ها aging یا سالمندی یا EWMA گفته میشود. دقت کن وزن گذشته به صورت نمایی هعی کمتر میشود چون مرتب در یک الفا یا متغیری ضرب میشود و دیفالت ما 1.8 هست اگر در کنکور ندادند همین را دیفالت بگیر و مرتب در آن ضرب میشود. و مشخص میشود آخری ها بیشترین اهمیت را دارد مثلا در مثال خودش آخری ¼ اهمیت را دارد به تنهایی و برای هزار تا هم همین هست و ¼ اهمیت دارد نه 1.1000 مثل میانگین عادی.

نوسان نشان میدهد که باید بیشتر از میانگین معمولی بگیری چه قدر بیشتر؟ میزان نوسان اگر نوسان کم باشد به اون میانگین باید کم اضافه کنی و اگر نوسان بیشتر باید بیشتر اضافه کنی به عنوان مثال اگر میانگین 100 باشد و نوسان بین 90 تا 110 باشد ما 120 باشیم خوب هست ولی اگر میانگین 100 باشد و نوسان 40 باشد آنوقت بین 60 تا 140 داریم نوسان میکنیم و 120 کم هست. در عمل انحراف معیار یا نوسان را تخمین میزنند انحراف معیار اختلاف بین نمونه ها و میانگین هست و قدر مطلق آن میشود که میانگین بگیری که اینجا 1.4 میگیریم بر خلاف قبلی که 1.8 میگرفتیم و به آخری 25 درصد وزن میدهیم به آخرین حاصل منهای نمونه با میانگین. پس 2 تا عدد شد یکی میانگین وزن دار که مثلا 100 هست و یکی میانگین نوسان که 10 هست که بین 90 تا 110 دارد نوسان میکند پس برای اینکه ریسک نکنیم بیا اون میانگین نوسان را ضربدر 4 کن و 4 تا 10 یا 40 تا یعنی + 100 میانگین زمان بگیر اگر نیومد تایم اوت شده است. مقدار اولیه تایم اوت 1 ثانیه هست اگر تایم اوت رخ داد یعنی اوضاع خراب است پس این تایم، تایم اوت را دو برابر کن بعدش هم دوباره بر اساس همین فرمول برو. دقت کن برای اینکه بدانی تایمر برای بسته جدید چه مقدار باید باشد اختلاف زمانی ارسال دو بسته را بدست بیار بعد ببین از بسته ای که جوابش آمده است از تایمر آن تا ریست شود چه قدر مانده است بعلاوه اون اختلاف بکن مقدار تایمر منطقی واسه اون بسته را نشان میدهد.

**جلسه بیست و ششم:**

دقت شود که next sequential number عددی تصادفی است، base number هم که شماره ترتیب اولین بایت بسته بعدی هست. دقت کن وقتی تایم اوت شود فقط اون بسته که ACK آن نیامده و کمترین شماره base را دارد چون قدیمی هست که تایم اوت شده است پس دوباره تایمر را برای بسته دوم restart میکنیم. و اونکه تایم اوت شد و ارسال مجدد شد میرود ته جدول منطقی بعد چون تجمعی هست وقتی ACK 9 بیاد یعنی 8 و 7 هم رسیده است پس یهو base را تا آنجا ببر جلو. چطوری میفهمیم ACK جدید هست؟ وقتی از بیس ما بیشتر باشد. بعد تایمر را برای اون قطعاتی که هنوز ACK آنها نیومده استارت کن. تا الان لاست را تایم اوت detect میکردیم ولی در این روش با دریافت 3 تا ACK تکراری سریع دوباره بیت را ارسال میکنیم در کد این قسمت اگر بزرگتر از base بود که هیچی اگر نبود مثلا مساوی بود شمارنده میزنیم تا به 3 برسد و اون قطعه y که 3 بار تکرار شده است را دوباره میفرستیم دقت کن به بسته بعدی که اول آن y هست را دوباره میفرستیم. بایت شمار بودن و شماره تصادفی sequence number که تا ACK های اتصال قبلی با اتصال جدید قاطی نشوند از ویژگی های خود TCP هست.

**جلسه بیست و هفتم:**

TCP به فرستنده اطلاع میدهد که گیرنده چه قدر بافر آن جا دارد و چه قدر میتواند بفرستد و هر موقع خالی شد پنجره دریافت باز شود. دو شرایط یا دو عامل محدود کننده واسه فرستنده وجود دارد یکی ازدحام شبکه دو پر بودن بافر ها و عمل کنترل جریان و ازدحام برای همین دو علت هست که بسیار بهم شبیه هست امروز با کنترل ازدحام کار داریم زمانی که شبکه ازدحام دارد، اول فرض کنیم بسته که رسید خارج از نوبت بود را دور میندازیم، پنجره دریافت در هدر اون پیام هست. آخرین بیت خوانده شده و آخرین بیت خوانده شده باید کمتر از طول بافر باشند چون بیشتر باشد یعنی سرریز شده است و از منهای یکی با طول بافر اون یکی بدست میاد. دقت کن آخرین مقداری که فرستنده ارسال کرده است و آخرین بایتی که ACK آن آمده است را منها کنیم تعداد بایت های سرگردان هست که هنوز تعیین و تکلیف نشده اند و تایید نشده اند و این مقدار نباید بیشتر از سایز بافر اعلام شده باشد و گرنه سرریز میشود. با قانون اساسی کنترل جریان باعث جلوگیری از سر ریز میشویم. یک مشکل وجود دارد این هست که گیرنده گفت علی بافر من جا ندارد و شروع میکند به مصرف کردن ولی بعدا دیگر علی نمیفهمد که این بافر دارد خالی میشود و من میتوانم بفرستم چرا؟ چون هیچ داده ای گیرنده ندارد که به علی ارسال کند تا علی را متوجه کند چون پنجره دریافت بسته شده است. چون کلا یا داده میفرستیم یا پیام ACK. TCPمیگوید کلا فرستنده را مسدود نمیکنم از اون داده ها 1 بایت میفرستم تا فضای اشغال نشود و از حال گیرنده با خبر شوم که آیا جا دارد یا ندارد اوضاع چطور هست. مدیریت اتصال: اگر مدیریت اتصال TCP را خوب بلد نباشیم مورد حمله هکر ها قرار بگیریم و یا موجب افزایش شدید تاخیر ها شود. این قسمت را خوب بخوان.

دقت کن در مرحله سوم هم در ACK میاد شماره سرور را بعلاوه 1 میکند که بگوید SERVER\_ISN سرور را تایید کردم هم SEQ را CLIENT\_ISN میگذاریم چون سرور بعلاوه 1 کرده بود گذاشته بود. هر دو طرف میتوانند قطع اتصال کنند هم سرور هم کاربر. وضعیت SYN\_SENT یعنی اینکه کاربر SYN را فرستاده و منتظر است SYN ACK هست در سمت کاربر. در established تبادل داده دو طرفه داریم یعنی مرتب داده میفرستیم تا وقتی که بخواهیم اتصال را قطع کنیم قبل از کامل بسته بودن هم چند ثانیه صبر میکنیم تا بسته های سرگردان و هنوز نرسیده برسند و بعد قطع میکنیم و منابع آزاد میشوند. در قسمت سرور این برعکس هست سرور در حالت listen هست و از طریق ولکام سوکت منتظر درخواست ها هست. RST فلگ ریستارت هست مال وقتی هست مثلا یک بسته ای آمده است با پورت 80 ولی ما هیچ ولکام سوکتی با این پورت نداریم اصلا HTTP سرور نداریم و این فلگ را 1 میکنیم یعنی جواب منفی میدهیم به کاربر که اصلا همچین چیزی اینجا نیست. UDP میاد ICMP میفرستد این پروتکول وظیفه گزارش خطا دارد در کنار IP. NMAP ابزاری هست که نگاه میکند چه پورت هایی باز هست در هدف یا مقصد یعنی یک بسته الکی میفرستد مثلا میگوید SYN 6789 میفرستد که این عدد الکی هست و میزبان SYNACK میفرستد و NMAP میگوید این پورت باز است و هکر متوجه میشود. حالت سوم فایروالی اون وسط هست که اصلا پاسخی نمیاد. اگر همزمان فرستنده و گیرنده FIN و ACK آنرا بفرستیم دیگر منتظر FIN\_WAIT2 بمانیم. دقت کن هر دو طرف میتوانند اتمام اتصال را آغاز کنند.

**جلسه بیست و هشتم:**

اصول کنترل ازدحام: مهم ترین نشانه ازدحام، تایم اوت رخ داده است به خاطر lost. ارسال مجدد بسته نشانه ازدحام هست. یک نشانه دیگر 3 تا ACK تکراری شدن هست. برای کنترل ازدحام 2 رویکرد هست یا از روتر ها کمک بگیریم و بگیم کمتر بفرستند یا بدون کمک روتر که به ما اعلام ازدحام کنند، میفهمیم که مسیر شلوغ هست به صورت انتها، انتها. دقت کن چون روتر ها اصلا لایه انتقال را نمیفهمند و به هدر آن توجهی ندارند پس روش دوم کنترل ازدحام به صورت انتها به انتها صورت میگیرد. گاهی اوقات روش اول هست و گاهی اوقات روش دوم و مثلا TCP روش دوم هست و هیچ کمکی از روتر های بین مسیر نمیگیرد. در روش دوم از نشانه ازدحام میفهمیم که ازدحام وجود دارد مثل بالا رفتن تاخیر. برای لاست 2 نشانه داریم اولی خیلی قوی هست که تایم اوت هست و دومی ضعیف تر هست که 3 ACK تکراری بیاد. در روش اول شبکه با یک سری پرچم اعلام میکند ازدحام شده است. مثلا در همین روش یک روش کنترل ازدحام هست به نام ATM ABR که میگوید روتر ها خودشان اعلام کنند چه قدر بیت ریت خالی دارند اعلام کنند تا مبدا تصمیم بگیرد سرعت انتقال را کم یا زیاد کند. پروتکول XCP دقیق هست به طور مشخص به فرستند میگوید چه قدر کم و زیاد به سرعت خودت اضافه کنی. در رویکرد های کنترل ازدحام توسط شبکه دو رویکرد داریم 1. یا مستقیم به فرستنده میگوییم که کمتر بفرستد یا غیر مستقیم بگیم روتر پرچم را ببرد بالا بسته ما به عمه ما برسد و عمه ما به ما که فرستنده هست بگوید ازدحام شده است و سرعت را بیار پایین. ATM چندین رویکرد را با هم استفاده میکند هم مستقیم هم غیر مستقیم، ABR وقتی بیت ریت زیاد باشد تشویق میکند زیاد بفرستید و وقتی بافر ها پر شوند و ازدحام باشد میگوید کمتر بفرستید و نرخ را کم میکند. به جای روتر از سوییچ استفاده میکنند، بعد سلول RM از طریق مقصد به مبدا ارسال میشود تا یک سری اخبار بین مسیر را مبدا هم متوجه شود. همچنین یک سوییچ میتواند سلول RM خود را بسازد و مستقیم به مبدا ارسال بکند. بیت EFCI همون کار آقا پلیس و عمه را میکند یعنی پرچم را میدهد بالا عمه میفهمد بعد عمه که مقصد هست به ما مبدا میگوید. به 2 صورت خبر میدهد 1. با تاکید زیاد که میگوید کمتر کن ارسال بیت را و 2. میگوید بیشتر از این نکن و تاکید کمتری دارد. CI میگوید نرخ را کم کن و نشانه ازدحام شدید هست و تاکید زیاد و NI میگوید فقط اضافه نکن ریت را. تنظیم ER: صراحتا میگوید چه مقدار ما بیت ریت آزاد داریم مثل قبلی ها نیست بگوید ما ازدحام داریم یا نه. اگر ER زیاد باشد زیاد اضافه میکنیم و برعکس.

**جلسه بیست و نهم:**

کنترل ازدحام TCP: از کنترل ازدحام انتها به انتها استفاده میکند و فرستنده ها را وادار به کنترل ارسال و تزریق بیت میکنند. RWND هم که یادت هست پنجره دریافت هست و به فرستنده میگوید به اندازه این پنجره میتوانی بیت های سرگردان تعیین و تکلیف نشده داشته باشی و برای کنترل جریان بود حالا یک متغیر دیگر داریم به نام CWND که برای کنترل ازدحام هست، و فرمول قبلی که گفته بود آخرین بیتی که ارسال شده با آخرین بیتی که ACK آن آمده است تفاضل اینها باید کمتر مساوی پنجره دریافت باشد تا بافر فرستنده سرریز نشود همچنین الان میگیم که باید از CWND هم این مقدار کمتر باشد تا بار شدیدی تحمیل نکنیم به شبکه پس باید از مینیموم این 2 کمتر باشد. دقت کن نرخ انتقال باید طول تقسیم بر زمان باشد زمان همان RTT هست همان زمانی هست که جواب بسته های ACK نشده میاد و بعد که اومد به اندازه CWND میتوانیم بفرستیم و صورت هم بدست آمد. پس در هر RTT ما به مقدار CWND میفرستیم و اگر ما نرخ فرستنده را محدود کنیم یعنی CWND را محدود کردیم و گرنه که RTT یک چیز ثابت هست. اگر نرخ ACK ها کند هست یعنی دیر به دیر میرسند پس افزایش ارسال بیت را شیب کمی افزایش میدهیم ولی اگر ACK ها تند به تند دارند میرسند ما هم ارسال بیت را با شیب زیادی بیت ارسال میکنیم. مثلا اگر 10 تا ACK بیاد به CWND 10 تا اضافه میکنیم اگر 2 تا بود 2 تا اضافه میکنیم این کار را با یک خود شمار انجام میدهد. دقت کن وقتی ازدحام شد نباید یهو خیلی کم کنی و از پهنای باند شبکه استفاده نکنند پس TCP میگه فرستنده ها باید با بالاترین نرخی که ازدحام پیش نیاد بفرستند. TCP در همه سیستم ها هست و متمرکز نیست و ماشین ها بر اساس اطلاعات محلی خودشان تصمیم میگیرند که همان نشانه های مثبت و منفی بود. دقت کن تا زمانی که لاست نشود کم به کم افزایش میدهد بعد که لاست رخ داد یهو به شدت کاهش میدهد دوباره همین کمی افزایش بعد یهو کاهش. دقت کن اگر MSS روی 100 بود و بوی ازدحام آمد تو دفعه بعدی از 50 بگیر بفرست. دقت کن اول یهو زیاد میبرد بالا بعد کم به کم افزایش میدهد تا بتواند از ازدحام اجتناب کند. دقت کن وقتی نصف کردیم و بوی ازدحام آمد دوباره از اول شروع میکنیم و نمایی اضافه میکنیم تا به آستانه ازدحام برسیم اونجا که رسیدیم دیگه از 32 نمیپریم 64 بلکه به صورت خطی یک به یک اضافه میکنیم. از نصف جایی که ازدحام رخ داد میایم اجتناب میکنیم. تایم اوت بوی ازدحام شدید و 3 ACK بوی ازدحام خفیف را میدهد و برای ازدحام خفیف slow start نمیکند. دقت کن تایم اوت شد slow start میکنیم چون شدید بوده. SSTHRESH همان نصف مقداری هست که ازدحام شده بود مثلا اگر 32 ازدحام شد میشود 16. دقت کن CWND میاد MSS میگیرد قطعه میگیرد که مثلا داخلش 500 بایت هست. دقت کن اگر به آستانه رسیدی اجتناب میکنی که بعد از تایم اوت و ازدحام رخ میدهد. دقت کن اگر 3 تا ACK تکراری اومد اون کار قبلی که میشود ارسال مجدد سریع تا قبل از تایم اوت را باید انجام بدهیم. فست ریکاوری: برای TCP های جدید هست بعد از اینکه 3 تا ACK تکراری بیاد صورت میگیرد. دقت کن یا دفعه اول از یک جایی تصادفی شروع میکنیم یا slow start میکنیم اگر تو سوال دیفالت نداد slow start بکن. TCP RENO که مدل جدید هست میاد فست ریکاوری میکند دیگر از اول نمیاد شروع کند از یک جایی اون وسط ها شروع میکند بر خلاف TCP TAHOE که مدل قدیمی تر هست و بعد از 3 ACK از اول شروع میکند به SLOW START. شروع آهسته هم که میدانی نمایی میرود جلو. دقت کن وقتی 3 تا اومد SSTHRESH هم چون روی 12 رخ داد گذاشتیم روی 6. اینقدر اضافه میکنیم تا 3 تا ACK بیاد یا تایم اوت شود. به حد اجتناب رسیدی رشد از نمایی به خطی میرسد. دقت کن اون تقسیم MSS/CWND نشان میدهد که در هر CWND چه قدر MSS جا میشود و چه کسری از آن را باید بگیریم و ضربدر MSS کنیم تا به ACK ما اضافه شود. توضیح همین هست که چطوری رشد خطی دارد. دقت کن عقب نشینی تایم اوت همیشه slow start هست. دقت کن وقتی 3 تا ACK میاد نترس ملایم تر رفتار کن یعنی شبکه هنوز دارد کار میکند که 3 تا بسته رسیده است که گیرنده به تو 3 تا ACK داده است پس تو بیا دیگه از اول شروع نکن از نصف اون مقداری که 3 تا ACK اومد شروع کن بعلاوه 3 تا کن بخاطر اون 3 تا ACK که رسیده است یعنی اگر در 12 رخ داده است از 9 شروع کن خطی اضافه کردن به این میگویند فست ریکاوری و فقط این ورژن TCP این کار را میکند. دقت کن به ازای هر ACK باید dupACKcount را صفر بکنیم که مشکل پیش نیاد برای سایر ACK ها. دقت کن وقتی 4 تا بفرستد منتظر 4 تا نمیماند یک ACK بیاد چون پنجره باز میشود دوباره شروع میکند به فرستادن. دقت کن قطعه ای که تایم اوت رخ داد را باید دوباره بفرستی.

**جلسه سی‌ام:**

دقت کن TCP به صورت AIMD هست یا additive increase multiple decrease یا افزایش جمعی و کاهش ضربی هست یعنی وقتی افزایش پیدا میکند یک به یک اضافه میشود ولی وقتی کاهش پیدا میکند ضربی هست که ضربدر یک دوم میشود و نصف میشود. و TCP به صورت دندان اره ای است. الگوریتم وگاس دو ایده اساسی دارد 1. اینکه همیشه ما ازدحام را وقتی گند میزدیم متوجه میشدیم مثل لاست ولی وگاس میخواد ازدحام را در مسیریاب ها و قبل از لاست متوجه شود و بوی ازدحام میاد و هنوز ازدحامی نشده است و تشخیص بدهد و بعد میاد بر خلاف کاهش ضربی میاد کاهش خطی میکند یعنی یک به یک میاد پایین. دقت کن لاست سر W رخ میدهد(loss). و میانگین همیشه 0.75 W/RTT چون همیشه بین ماکس آن و نصف آن بالا و پایین میشود دقت کن از slow start صرف نظر کردیم اگر بود هم چون خیلی کوچک هست اهمیتی ندارد. دقت کن 2 تا فرمول داریم اگر فقط W را دادند یعنی نقطه ای که مثلا تایم اوت یا 3 تا اک آمد را دادند یعنی اندازه اون پنجره مثلا گفتند 100 MSS و RTT را دادند فرمول اول که 0.75 دارد اگر نه نرخ لاس هم دادند از فرمول دوم استفاده میکنیم. دقت کن اگر بین هر لاست نیم ثانیه فاصله باشد نرخ آن 2 لاست در واحد زمان هست. دقت شود نرخ انتقال را مساوی بین لینک ها تقسیم میکنیم. K تعداد اتصال هست و r نرخ انتقال کلی هست. رفتار TCP رفتار عادلانه ای است یعنی اگر 3 تا HTTP بود یک FTP و پهنای باند 4 مگابایت بود به هر کدام 1 مگابایت میرسد البته این نسبتی هست ولی ما در کل در نظر میگیریم. دقت کن نقطه وسط اون ضربدر برای ما مناسب هست چون هم هر دو تا اتصال سهم پهنای باند مشترک و برابر دارند هم خط بهره وری روی پهنای باند کامل هست و بهره وری ما خوب هست و اگر ما از هر نقطه ای مانند A شروع کنیم باز به اون ضربدر وسط میرسیم. دقت کن در نقطه B ازدحام شده است چون از بهره وری زدی بالاتر وقتی این خبر میرسد جفت اتصال ها شروع میکنند به کاهش چون کمتر از R/2 میشود به C میرسیم و همین برای ادامه. عدالت خیلی هم واقعیت ندارد. 1. چون ما فرض کردیم فقط اتصالات TCP وجود دارند و RTT هم یکسان دارد. دقت کن ما علاقه داریم به RTT های کوچکتر چون مخرج کسر کوچکتر هست و کسر بزرگتر هست و پهنای باند بیشتری استفاده میشود و زودتر رشد میکنند و سهم بقیه را میگیرند و زود تر پیام های ACK آنها برمیگردد. RTT کمتر یعنی استفاده بیشتر. 2. UDP ها هم در شبکه هستند و میان پهنای باند را میگیرند. 3. اتصالات موازی هست یعنی اینکه مثلا یادمان بود که اگر یک فایل HTTP داخل آن 10 تا عکس بود میتوانی برای دانلود هر کدام 10 اتصال موازی ایجاد کنی و هر کدام سهم میگیرند. و در نهایت نقاط ضعف TCP در عدالت هستند و TCP هم عدالت ندارد و هیچکدام ایده آل نیستند. QUIC الگوریتم لایه کاربرد هست که بعضی از وظایف را انجام میدهد. QUIC روی UDP میشیند و نه به سفت و سختی TCP هست و نه به آسانی و سبکی UDP بلکه خودمان نیاز های خودمان را میگذاریم. TLS امنیت HTTP را به شما میدهد. UDP هیچی نمیدهد ولی QUIC هم امنیت هم کنترل ازدحام هم تشخیص خطا و کنترل جریان را میدهد و بعدا HTTP3 در داخل خودش HTTP2 و QUIC را دارد. چرا QUIC سریعتر از TCP هست؟ مگه همون امکانات را درون QUIC نگذاشتیم؟ مگه همان سربار را ندارد؟ خوب چرا سریعتر هست؟ چون در TCP چند تا RTT داشتیم واسه اینکه یک اتصال ایجاد شود که 2 تا بود و دوباره باید یک سری اطلاعات دیگه پاس کنیم تا امنیت برقرار شود برای TLS یعنی دو تا جدا ولی QUIC یک ضرب در یک بار هم پارامتر های امنیت و کنترل جریان و ازدحام و تشخیص خطا را میفرستد و جدا به جدا نیست یعنی تکرار دست تکانی داریم. و QUIC باعث سرعت بیشتر در استریم ها میشود و در یک اتصال چندین استریم را مالتی پلکس میکند و مشکل head of line را هم حل میکند. حالا HTTP3 اصلا میتواند برای هر شی جریان مستقل ایجاد کند و عدالت را زیر سوال نمیبرد. چطوری بسته ها در استریم قاطی نمیشوند؟ هم شناسه اتصال داریم هم شناسه استریم داریم یعنی استریم 1 از اتصال شماره 10 یا استریم 2 از اتصال شماره 10. دقت کن مشکل head of line و مالتی پلکس کردن را قبلا هم وجود داشت و پروتکول هایی بودند که سعی کردند این مشکل را حل کنند و پیشگام تر بودند. Head of line چی بود ؟ من گیرنده 10 تا بسته میخواهم که به لایه بالاتر تحویل بدهم ولی از 10 تا شماره 2 آن نیامده است من 3 و 4 و5 و6 را هم درون بافر میگذاریم چون صرفا باید به ترتیب تحویل لایه بالاتر بدم و من اصلا نمیدانم که این شاید مال یک چیز دیگر باشد و ربطی نداشته باشد و الکی در بافر بسته ها را نگه میداریم. QUIC جلوی این اتفاق را میگیرد یعنی میگوید اگر استریم شماره 2 مشکل دارد به استریم 3 چه کاری دارید اونها را به ترتیب تحویل بدهید و ترتیب را فقط درون هر استریم کنترل میکند. دقت کن در HTTP3 و QUIC برای هر درخواست و استریم قابلیت اطمینان را جدا رعایت میکنند و مشکل HOL را حل میکند.

فصل چهارم:

SDN ها میان مسیریابی میکنند امروزه نه روتر ها. هدایت یعنی این بسته را حالا به کدام پورت خروجی بفرستیم. روتر ها لایه سوم هم دارند ولی سوییچ ها فقط لایه 1 و 2 را دارند. دقت کن بهترین مسیر را الگوریتم مسیریابی و تعیین تکلیف اینکه بسته فعلی در کدام پورت خروجی قرار بگیرد را هدایت گویند. جدول های مسیریابی توسط الگوریتم های مسیریابی پر میشوند. هدایت محلی و مسیریابی سراسری هست. مدار مجازی ها کانکشن فیزیکی برقرار میکنند بر خلاف ip که اصلا کانکشن برقرار نمیکند. روتر ها اطلاعات وضعیتی را نگه میدارند مثل TCP.

**جلسه سی‌دوم:**

اتصال منطقی یعنی اینکه یک سری بافر در دو لبه اینترنت یا همان کاربران نهایی وجود دارد و یک سری بافر و منابع را ذخیره میکنند در دو طرف سیستم انتهایی و متغیر های کنترل جریان و ازدحام بین دو طرف هست. دقت کن اینکه در هر لینک و مسیر شماره مجزا به بسته میدهیم باعث سادگی و راحتی ما هست. چون بسته ها وقتی از یک روتر میان که دو طرف آن مدار مجازی برقرار هست نمیتوانند شماره یکسان داشته باشند چون نمیتوانیم بفهمیم این بسته مال اتصال 1 هست یا اتصال 2، چون 2 تا اتصال مدار مجازی داریم. این کار باعث میشود که ما فقط در همین لینک شماره مجزا داشته باشیم و دوما اعداد ما خیلی بزرگ نباشد چون شماره یکسان اولا باعث میشود همه بدانند که این شماره واسه ما هست و دوما اعداد بزرگ میشود. دقت کن در مدار مجازی مقصد و مبدا را اول کار پرسیده اند بین روتر ها فقط ما میگیم شماره مدار مجازی شما چند هست با این شماره مجازی برو بیرون. دقت کن این شاخص گذاری ها چون نیاز به سرچ داریم و بر اساس یک تگ یا شماره هست سرعت بسیار بالایی دارد برای همین ATM ها. پیام های سیگنالینگ چی هست؟ برای دست تکانی اولیه و ایجاد ارتباط و رزرو منابع در دو طرف هست و کیفیت سرویس هست که پروتکول سیگنالینگ داریم. شبکه های دیتاگرام یا همون ip: فقط میپرسیم مقصدت کجاست و میفرستیم تمام نه رزرو منبعی هست نه دست تکانی نه مدار مجازی برقرار هست. در دیتاگرام هدایت کردن فقط با آدرس مقصد کار دارد و اصلا نیازی به مدار مجازی و شماره مدار مجازی نیست. حالا شاید بگی که جدول خیلی بزرگ میشه که ولی اینطوری نیست ما کلک میزنیم و محدوده بندی میکنیم میگیم از range فلان تا فلان از فلان لینک خارج بشود. دسته بندی میکنیم مثلا میگیم آمریکا از این ور برن اگر بسته ها برای تهران هستند از اینور بروند تا جدول محدود شود. 21 بیت مشترک و 11 بیت آن اختصاصی هست یعنی یک رنج دو به توان 11 تایی را دارد همه این آدرس ها پیشوند یکی دارد یعنی مثلا 3 تای اول کشور 3 تای بعدی شهر را نشان میدهد پس سریع میفهمیم این بسته مال تهران هست و شهرستان ری و دیگر بقیه مهم نیست دقیقا کجا هست. با پیشوند مشخص میکنیم کجا بروند. ممکن هست یک بسته بیاد 2 تا آدرس مطابقت داشته باشد ولی اونی را میپذیریم که پیشوند بیشتری جور هست باهاش و دقیق تر هست. دقت کن برخلاف مدار مجازی ما دیگر index نداریم و واقعا باید سرچ کنیم و سرچ باعث کندی سرعت میشود راه حل چیست؟ از جستجو موازی استفاده کنیم. جداول هدایت در دیتاگرام هر یک تا 5 دقیقه اپدیت میشود ولی مدار مجازی ها چون به ازای هر اتصال باید اپدیت کنی در مقیاس میکرو ثانیه هست در ستون فقرات و بدتر از دیتاگرام هست.

تا پایان این جلسه دیده شد.

**جلسه سی‌سوم:**

دقت کن پورت ورودی و خروجی به یک لینک وصل هست یعنی مثلا فقط ورودی و فقط خروجی نداریم. هر روتر پورت های مربوط به 3 لایه اول را دارد از فیزیکی تا ip. و دقت کن هدایت میکنیم به بافر پورت خروجی از طریق همان زیر بنای سوییچینگ. اون پورت لایه های قبل مربوط به سوکت نامبر هست ولی در این فصل همان منظور interface دارد که نقش ورودی و خروجی مسیریاب ها را دارد. و وظیفه پردازنده این هست که این بسته که در بافر ورودی قرار دارد از طریق کدام پورت خارج شود و جدول هدایت را پر میکند. دقت کن هر 51.9 ثانیه یک بسته میاد و تو این زمان وقت داریم تا تعیین تکلیف کنیم این بسته را و گرنه بافر ما پر میشود. در رویکرد مدرن امروز از SDN استفاده میکنیم یعنی تمام این پردازنده های روتر ها جمع شده و داخل یک مرکزی هست و الگوریتم های مسیریابی آنجا اجرا میشود. و بعد نتایج را درون دیتا پلین ها اعمال میشود. SDN یک دیتاسنتر هست. پورت ورودی: انتهای خط امواج را به داده 0و1 تبدیل میکند. در مرحله بعد کنترل خطا و جریان و دسترسی به رسانه انجام میشود دقت کن کنترل جریان فقط در لایه انتقال نیست بلکه به صورت hub by hub هم هست. کنترل دسترسی به رسانه مشترک هم بخاطر این هست که بسته ها قاطی نشوند مثلا 2 نفر همزمان دیتا روی وای فای قرار ندهند. خاصیت این روش این هست که به جای اینکه ما در پردازنده یک نرم افزار داشته باشیم و واسه n تا پورت ورودی و به صورت time sharing بیایم و تعیین تکلیف کنیم این 3 مرحله را و کلی وقت بگیرد در حالی که وقت ما محدود هست روی همان پورت ورودی به صورت جدا به جدا این کار را انجام میدهیم. CAM میگه جای اینکه خطی کل خانه های حافظه را مقایسه کنی اگر 10 تا خانه داری بیا 10 تا را به صورت موازی جستجو کن OR کن بعدش اگر کلا شد 0 یعنی پیدا نشد و اگر یکی 1 شد کلش 1 میشود پس hit شده است. بخاطر همین میگویند حافظه انجمنی.

تطابق را قبلا هم داشتیم برای تطابق دادن با طولانی ترین پیشوند. هدایت کردن میشود عمل و یافتن یک آدرس ip میشود تطابق. پس کلا تطابق بعلاوه عمل داریم. در NAT پورت نامبر را عوض میکنیم. تطابق عمل باید هر اکشنی در هر لایه ای را ساپورت کند. پس نگاه کن یک بسته میرسد وقفه میاد بعد پردازنده بسته را برمیدارد میگذارد در حافظه خودش در روش مبتنی بر حافظه، بعد به آدرس مقصد نگاه میکند بعد مثلا 12 بعد به جدول هدایت نگاه میکند که به کدام پورت خروجی هدایت شود. برای تمام پورت های ورودی و خروجی یک باس و یک حافظه داریم در هر لحظه یک چیز میتوانیم بنویسیم بخوانیم، یعنی اگر بنویسیم نمیتوانیم بخوانیم و برعکس. یعنی اگر 10 تا بسته بیاد در هر لحظه فقط یکی را میتوانیم هدایت کنیم. واسه همین گذردهی یا انتقال بسته کمتر از B/2 هست چون 2 عمل انجام میشود به ازای هر بسته یکبار در حافظه مینویسیم یکبار میخوانیم برای هدایت کردن. روی هر پورت یک کنترلر داریم که انگار یک سری چند پردازنده هستند که از حافظه مشترک استفاده میکنند. در باس پورت ها شماره دارند و همه پورت ها میتوانند ببینند ولی فقط اونکه شماره متناظر دارد میتواند بسته را بردارد و فقط در هر لحظه یک نفر میتواند از باس استفاده کند. اگر n تا ورودی و n تا خروجی داشتیم در روش سوم میتوانیم n تا انتقال انجام بدهیم به شرط اینکه مقصد یکسان نباشد. برای پورت خروجی هم یک سری چالش داریم اولی الگوریتم زمان بندی بافر هست که باید به چه صورت باشد، دوما مثلا یهو 10 تا پورت ورودی میگن همه از پورت 3 خارج بشوید و اون بافر پر و سرریز میشود و لاست رخ میده چه کسی را باید بیندازیم دور کی باید بیندازیم دور؟ RTT دو برابر زمان انتشار هست. صف بندی عادلانه وزن دار مثل دو تای قبلی کلاس بندی دارد ولی نه مثل RR نگاه میکند که به همه کلاس ها دید یکسان داشته باشد و نه مثل اون یکی فقط به کلاس با اولویت بالا جواب میدهد بلکه میاد میچرخد منتها به کلاس ها با اولویت های بالاتر بسته های بیشتر عبور میدهد و تایم بیشتری میگذارد چطوری وزن را محاسبه میکند؟ مثلا اولویت من خیلی بالاتر هست و 3 بسته باید برسانی و وزن من 3 هست اون یکی 2 و اون یکی 1 میایم جمع وزن ها را حساب میکنیم در مخرج کسر و وزن خودمان را تقسیم میکنیم اون مقدار سهم به ما میرسد و وزن ما اونقدر هست. دقت کن این وزن حداقل هست چون ممکن است سایر دوستان شما کاری نداشته باشند و باز به شما برسد چون مبتنی بر کار هستیم و بیکار نمی ماندیم. DRR نوبت گردشی کسری میاد میگوید که اگه این سری کم فرستادی ایرادی ندارد من سهم تو را برای دفعات بعدی نگه میدارم مگر اینکه خاموش و غیر فعال باشی. بسته ها طول های مختلفی دارند پس باید بگوییم چند بایت حق هر کسی هست بر اساس بسته که متغیر هست نمیتوانی بگویی. در وزن دار قبلی میگفتیم مثلا 1000 بایت علی میتواند بفرستد علی 2 تا بسته 700 تایی داشت خوب علی فقط میتوانست 700 تا استفاده کند و بقیه 300 تا سوخت میشد در صورتی که در این الگوریتم برای دفعات بعدی نگهداری میکنیم. دقت کن اگر علی کلا بسته ای نداشته باشد کلا هم چیزی برایش نگه نمی داریم تا بقیه دچار گرسنگی نشوند. الگوریتم RED : با بقیه کلا فرق دارد چون مال وقتی هست که صف ها پر شده اند و دارند سر ریز میکنند و باید بدانیم کدام پکت را حذف کنیم جدید تر ها یا قدیمی تر ها این الگوریتم پیشبینی میکند که بافر دارد پر میشود. 2 تا آستانه میگیرد برای حالت پر شدن یا نیمه پر یا خالی شدن. وقتی از max TH عبور کنیم اوضاع خراب است، دقت کن اوضاع میانگین را میگیرد نه لحظه ای. اگر میانگین لحظه ای قبل از آستانه اول بود که هیچی حذف نمیکنیم اگر بعد از آستانه دوم بود به احتمال 1 یعنی قطعا حذف میکنیم و اگر بین 2 آستانه بود به احتمالی بین 0 و 1 حذف میکنیم که این عدد اگر به آستانه دومی نزدیک تر باشد پس به 1 هم نزدیک تر میشود. دقت کن در انتقال در زیر بنای سوییچ یا همان باس در هر لحظه برای هر پورت خروجی فقط و فقط یک پورت ورودی میتواند بهش متصل شود و خارج شود بنابراین در شکل بعدی بخاطر همین هست که در ابتدا بسته آبی رنگ نمیتواند منتقل شود با اینکه پورت خروجی آبی مشغول نیست زیرا جلوی یک بسته قرمز رنگ هست و بسته قرمز رنگ مکس کرده است چون قبلا پورت ورودی دیگری آن باس را تا پورت خروجی اشغال کرده است.

**جلسه سی‌چهارم:**

8 بایت از 20 بایت هدر IP برای آدرس مبدا و مقصد هست. طول سرآیند بین 5 تا 15 هست حداقل 5 و حداکثر 15 هست. IP میاد checksum را میگیرد برای مراقبت خودش نه برای مراقبت بچه ما یعنی فقط مراقب هست هدر خودش خراب نشود بسته ما خراب شد هم مشکلی نیست. دقت کن در IP ممکن هست بسته های ما شکسته شود مثلا یک روتر MTU آن بسیار از بسته ما کوچکتر باشد برای همین تیکه میشود ولی در ورژن 6 برخلاف ورژن4 بسته را بیرون می اندازد و میگوید کوچکتر کن و دوباره بفرست به روتر فرستنده، دقت کن وقتی یک بسته میرسد ما 3 تا چیز را باید بدانیم، آیا بسته فعلی تکه ای از دیتاگرام بزرگتر هست یا خیر که از روی شناسه میفهمیم، سوال بعدی این هست که مقصد باید بفهمد این بسته چند تا تیکه دارد و تا کی ارسال میشود و آخرین کدام هست از روی فلگ ها و سوال سوم ترتیب این بسته ها هست که به چه شکلی باید باشد که از روی آفست متوجه میشود و همه اینها در هد دیتاگرام وجود دارد. وقتی MF صفر میشود متوجه میشویم این دیتاگرام آخرین تکه از آن دیتاگرام بزرگی بوده است که تکه به تکه شده است و مخفف more fragment هست. ترتیب دیتاگرام برای وقتی که میخواهیم تکه ها را سر هم کنیم چی ؟ با آفست نشان میدهد این تکه که داریم دقیقا باید کجا باشد، اگر اونجا نبود به اندازه آفست برو جلو تا به جای اصلی برسد اگر هم اونجا نبود دوباره به اندازه آفست برو جلو. دقت کن هدر های TCP و UDP با اینکه هدر هستند ولی جز داده اصلی در دیتاگرام حساب میشوند و هدر IP فقط 20 بایت هست. و دقت کن وقتی میخواهی تکه به تکه کنی باید 20 بایت هدر دیتاگرام را هم در نظر بگیریم نمیتوانی به تکه 1500 تایی بشکنی چون با 20 تا هدر میشود 1520 تا و خطا میخورد باز. شناسه تکه ها یکی هست. دقت کن برای آفست فاصله تکه ها را تقسیم بر 8 کن بعد بنویس. چرا؟ چون آفست را در هدر 13 بیتی گرفته هست ولی طول دیتاگرام 16 بیتی هست پس اگر بزرگترین حالت ممکن رخ بدهد ما تقسیم بر 8 میکنیم که در 13 بیت جا شود. پس تکه ها را تقسیم بر 8 کن. اگر عدد تقسیم بر 8 نمیشد و باقیمانده داشت بزرگترین عدد بخش بر پذیر 8 را که کوچکتر از عدد فعلی هست بگیر. هر واسط باید یک آدرس IP داشته باشد تا بتواند مسیریابی شود و بسته ها بهش برسد. یک میزبان میتواند چند واسط داشته باشد که هر واسط باید IP مخصوص به خودش را داشته باشد. دقت کن به روتر IP نمیدهیم بلکه به واسط ها میدهیم. اگر از NAT استفاده نکنیم باید دقیقا IP جهانی داشته باشیم ولی اگر استفاده کنیم میتوانی با IP لوکال هم کار کنی. subnetwork: یک مفهوم فیزیکی دارد که میشود یک زیر شبکه یعنی به جایی اینکه به یک بسته بگوییم محله مصباح در کرج و خیابان رنجی، میگوییم مثلا شبکه مصباح دیگه دقیق نمیگیم همون اول مفهوم چیست؟ یک سری LAN و شبکه محلی و subnetwork داریم و یک روتر وسط هست و با آدرس IP مشخص میکنیم که کدام ها مال کدام زیر شبکه هستند مثلا 223.1.3 برای یک LAN پایینی کلا مفهومش این هست که کارت های شبکه ای که بهم وصل هستند بدون اینکه روتری بین آنها باشد. هر LAN یک زیر شبکه هست ولی هر زیر شبکه، LAN نیست. دقت کن بین 2 تا روتر هم یک زیر شبکه میتواند باشد بدون وجود LAN چون کارت شبکه دارند دیگه مثلا در شکل بعدی بین R2 و R1 2 وجه هم دارد از نظر فیزیکی همین چیزی که گفتیم و از نظر آدرس دهی که آدرس مشترک دارند. IP آدرس دهی مبتنی بر کلاس دارد البته CIDR هم وجود دارد که آدرس دهی بدون کلاس هست. آدرس را طبقه بندی میکنند تا بتوانند پیدا کنند اگر یک مشت اعداد الکی بودند که نمیتوانستیم کسی را پیدا کنیم. آدرس 4 تا صفر یعنی اینکه یک نفر از شما بپرسد کی هستید شما بگویید نمیدانم من 4 تا صفر هستم. اگر خودم را بشناسم ولی شبکه را نشناسم دقت هم کن که به کلاس a,b,c بستگی دارد اگر a بود 1 بایت اگر b بود 2 بایت الی آخر. دقت کن ip میدن میگن طبقه بندی کنید و تعداد زیر شبکه و کلاس هم میدن میگن نحوه طبقه بندی زیر شبکه هاش چه شکلی هست. از بیرون ما همون 3 بایت اولی هستیم و اون بایت 4 امی آدرس های درون اون شبکه هستند دقت کن بقیه شبکه ها ما و زیر شبکه های ما را با همین 3 بایت میشناسند این 3 بایت برای بیرون دانشگاه هست. از کجا میفهمیم دو تا ip به یک زیر شبکه متعلق هستند یا به 2 زیر شبکه جدا متعلق هستند؟ از طریق subnet mask که الگو زیر شبکه هست و مهم هست که هر ماشین مبدا باید تشخیص دهد که ماشین مقصد در زیر شبکه خودش هست یا نه زیرا برای قرار دادن یک فریم در شبکه LAN اگر داخل زیر شبکه خودمان بود باید MAC آدرس روی LAN ما بدهیم ولی اگر مقصد از یک شبکه دیگری بود باید جوری آدرس دهی کنیم که روتر بتواند رد کند و آدرس فیزیکی روتر را باید قرار دهیم جا MAC آدرس داخلی چون اگر داخلی بگذاریم روتر دیگه رد نمیکند. 24 بیت اول شبکه و زیر شبکه را مشخص میکند. چطوری میفهمیم تو یک زیر شبکه هستند یا نه؟ با 24 تا بیت 1 بیا AND کن بایت 4 ام که کلا 0 هست حاصل میشود 0 که هیچی. این میشود subnet mask، زیرا با همدیگر AND کردیم و همچنین host هم گذاشتیم کنار چون در مرحله فعلی مهم نیست. اگر AND یکی شد که هیچی اگر نشد پس تو زیر شبکه ما نیست پس باید طوری آدرس دهی کنیم که بتواند از روتر رد شود. default gateway همان آدرس فیزیکی روتر پیشفرض هست. این 2 تا 8 کردن استاندارد هست در بقیه موارد خاص میتواند 6 10 باشد چونکه مثلا ما اینقدر زیر شبکه نداریم ولی دستگاه داریم پس بیت بیشتری به host ها اتلاق میکنیم. دقت کن کلا اگر گفتند چند تا زیر شبکه داریم ببین چند بیت اتلاق کردن بهش در مبنای 2 به توان برسان و منهای 2 کن همیشه مثلا با 4 بیت 14 تا زیر شبکه داریم. دقت هم کن اگر گفتند ما 31 تا زیر شبکه داریم 25 نگیری حواست باشد بلکه باید 2 تا بهش اضافه کنی و میشود 33 تا پس باید 26 بگیری.

پایان این جلسه.

جلسه سی‌پنجم:

مسئله کمبود ip های 32 بیتی هست یک مقداری هم آدرس هدر رفته بود واسه همین الان دیگه سراغ آدرس های بدون کلاس رفتیم. آدرس های بدون کلاس ربطی به ip ورژن 6 ندارد و همان 4 ها را میخواهد به صورت موثر استفاده کند. اسلش هشت همان کلاس A هست. ایندکس کردن هم که میدانی یعنی شماره گذاری کردن و مثلا خانه 1000 در جا پیدا میشود. دقت کن بعد از AND کردن میرفتیم داخل آن کلاس دنبال اون میزبان میگشتیم اما در CIDR داستان فرق میکند و کلا فیلد های 32 بیتی داریم و کلا یک جدول هست و طولانی ترین پیشوند را باید مبنا قرار بدهیم. جستجو سخت افزاری موازی انجام میشود در این بخش برای پیدا کردن طولانی ترین پیشوند که TCAM نام دارد. دقت کن اگر اسلش 21 مال زیر شبکه ما باشد پس اسلش 24 هم مال همین زیر شبکه ما میتواند باشد. دقت کن شریف چون 2 کیلو آدرس گرفته پس باید اسلش 21 را به او بدهیم چون خودش 211 هست. چطوری رنج IP را حساب کردیم؟ نگاه کن شریف 2 کیلو دارد داخل 2 کیلو چند تا 256 تا جا میشود؟ 8 تا پس میشود از 0 تا 7 و خود 7 هم تا آخرش یعنی 255 و چون اولش هشت تا 0 هست پس انتهای آخری هم هشت تا 1 هست. الان مثلا 256 تای دوم میشود 194.24.1.0 تا 194.24.1.255. یک راه دیگر این هست که بنویسی بعد شریف که 2 کیلو میخواست و میشود 211 پس 2 کیلو برو جلو و اونجا 11 بیت سمت راست را 1 کن همین عدد بالا بدست میاد. دقت کن طولانی ترین تطبیق برای وقتی هست که با چند تا تطبیق دارد بین اونها طولانی ترین. حالا یک سطح بالاتر بیایم کلا دانشگاه های تهران را در قالب ایران تجمیع کنیم برای کشورهای خارجی و دیگر نیازی نیست ما بدانیم این بسته دانشگاه تهران هست یا شریف و لازم نیست مجزا کنیم میگیم اینها تجمیع شده در تهران هستند و کلا بفرستید تهران و جا 3 تا entry به 1 entry میرسیم و 2 تا سود کردیم برعکس زیر شبکه که بیشتر میکردیم. ادغام کردن باعث جمع کردن و کوتاه تر شدن جدول یا خلاصه کردن میشود. دقت کن در این تجمیع کردن لازم نیست حتما داخل تهران باشد اگر یک دانشگاهی در ارومیه هم بردارد که مسیرش از تهران کاملا متفاوت هست هم میتواند از آن 1 کیلو فضای خالی استفاده کند به این شرح که بر اساس طولانی ترین پیشوند عمل میکند یعنی درست هست که با تهران یا شریف /19 تطابق دارد ولی با /22 ارومیه هم تطابق دارد و این بیشتر هست و بر اساس طولانی ترین پیشوند هست. و داخل جدول مسیریابی دو تا ردیف میگذاریم یکی برای تجمیع شده تهران و یکی برای ارومیه. دقت کن برای تجمیع کردن از چپ تا جایی که یکی هست یکی بگیر به همون مقدار هم بیت تخصیص بده برای قسمت مشترک مثلا وقتی بایت سوم آنها همه تو رنج 80 هستند خوب معلوم هست که تو 4 بیت اول سومین بایت با هم اشتراک دارند. دقت کن اون 4 کیلو که نوشته چون از سمت راست 12 بیت 0 دارد یعنی 212 که میشود 4 کیلو. حالا چرا 4 تا 256 نشده 1 کیلو شده است 4 کیلو؟ چون از اون 4 کیلو جدولی که ما گرفتیم اینها هر 4 تا که داخل یک قطعه 1 کیلویی نیستند که کنار هم باشند داخل 2 کیلویی هم نیستند بلکه داخل همان 4 کیلویی هستند چون پخش شده اند بین 2 کیلو بالا و 2 کیلو پایین پس باید 4 کیلو در نظر بگیریم. پروتکول بعدی: DHCP: یک آدرس IP، dynamic از رنج آدرس های سلف یا کتابخانه به صورت موقت به ما میدهد DHCP علاوه بر IP یک الگو زیر شبکه و آدرس روتر اولین گام و DNS را هم میدهد دقت کن آدرس روتر اولین گام همان default gateway هست که قبلا هم گفتیم که اگر در زیر شبکه ما نباشد مقصد باید آن را بشناسیم و برای آن بفرستیم. با دادن آدرس داینامیک میتوانیم در دادن ip ها صرفه جویی کنیم. دقت کن حتی اگر زیر شبکه ما هم DHCP سرور نداشته باشد روتر ها این امکان را دارند که درخواست ما را به DHCP سرور سایر زیر شبکه ها بفرستند و آنها برای ما این کار را انجام بدهند. پس هر LANیی الزاما DHCP سرور نداشته باشد همچنین یک زیر شبکه هم میتواند چندین DHCP داشته باشد و مثلا 3 تا سرور داشتیم یکی اومد داخل 3 تا پیشنهاد IP بهش میاد و این پروتکول همچنین تنظیم میکند کدام یکی از این پیشنهاد ها را بپذیریم. دقت کن 2 تا RTT طول میکشد برای ایجاد ارتباط با DHCP. روی UDP میشیند DHCP. Transaction id هم مشخص میکند دقیقا ما هستیم این برای وقتی هست که همزمان چند تا درخواست اومدند بدانیم برای کی هست. چرا یکی بهش اضافه کردیم چون وقتی یکی را پذیرفتیم لازم نیست به سایرین بگیم نه یکی به ایدی اضافه میکنیم بعد ip اون DHCP سرور هم میدهیم بگیم من با ایشون دیگه کار دارم. دقت کن اون yiaddrr ای پی هست که به ما داده میشود. NAT: پروتکول بعدی هست و نگاشت آدرس شبکه هست. که ترجمه آدرس شبکه هست. موضوع این جلسه کمبود آدرس IP هست. 10.0.0 فقط در LAN ما معنی دارد با اینکه آزاد برای کل جهان هست و ما پشت روتری که NAT را پشتیبانی میکند و جدول مخصوص به آن را دارد قایم شویم کافی هست روتر ما یک آدرس IP مشخص داشته باشد بعد اون میتواند این درخواست ها را برای هر سیستمی که داخل این IP خصوصی گرفته ترجمه کند چطوری میفهمد مال کدام ماشین هست از طریق پورت. چرا روتر پورت را عوض میکند؟ چون ممکن هست سیستم عامل ما شماره پورت دقیقا همزمان با همان شماره پورت از طریق یکی دیگه تو همین شبکه خصوصی درخواست کند واسه همین عوض میکند. و یکی میشود 5001 و اون یکی میشود 5002 و اگر برگشت با این پورت ترجمه میکنیم و متوجه میشویم برای کی هست. سرور ها پشت NAT نیستند.

دقت کن پروتکول بعدی UPnP برای همین مشکل که در یک شبکه p2p دو تا همتا در پشت NAT قرار گرفته باشند استفاده میشود. و مشکل همتای های پشت NAT هستند را حل میکند دقت کن وقتی یکی پشت NAT باشد یکی نباشد مشکلی ندارد از واسط سوم استفاده میکند این پروتکول برای هر وقتی هست که هر دو همتا پشت NAT باشند و همچنین هر دو NAT را میشناسند، تا قبل از این فرض میشد میزبان ها با NAT ارتباطی ندارند ولی الان میزبان به نزدیک ترین NAT ارتباط برقرار میکند و آدرس و پورت را درخواست میکنند. و ما میزبان خودمان را به ردیاب درون روتر اطلاع میدهیم و همتا دیگر ما از P2P، میتواند با ارتباط با این ردیاب ما را پیدا کند و ارتباط برقرار کند بدون مشکل.

جلسه 36:

ICMP یک پروتکول در بالای IP قرار میگیرد پروتکول لایه انتقال نیست بلکه پروتکول لایه شبکه هست و ربطی به IP ندارد. دقت کن شبکه غیر قابل دسترس هست یعنی پیدا کرده ولی مثل اینکه اینترنت اونجا قطع هست میزبان غیر قابل دسترس هست یعنی اینکه شبکه را پیدا کنیم ولی میزبان مثلا کامپیوترش خاموش هست. پورت غیر قابل دسترس باشد یعنی پروتکول قابل دسترس بوده است. ناشناس بودن یعنی پیدا نکردیم طرف را. دقت کن از اونجا که TCP انتها به انتها کنترل ازدحام میکند پس توجهی به پیام ICMP که میگوید ازدحام شده است توجهی نمیکند. چطوری traceroute کار میکند؟ یک سری دیتاگرام الکی تولید میکند و میفرستد و TTL آنها 1 را میگذارد اولین دیتاگرام به اولین روتری که رسید میگوید time exceeded و TTL ما از 1 تبدیل به 0 میشود و منقضی میشود و خطای تایم اوت یا منقضی شدن تایم را میدهد. پروتکول traceroute از ICMP استفاده میکند. بعد دقت کن برای پیام دوم یا دیتاگرام دوم TTL میشود 2 همینطوری به ترتیب یکی اضافه میشود و دقت کن که به ازای هر دیتاگرام میفرستیم یک تایمر روشن میکنیم و تایم رفت را بدست میاوریم و پیام خطا هم شامل نام روتر هست. دقت کن تهش هم خطا پیدا نشدن پورت میاد. Firewall فقط هدر بسته ip را نگاه میکند و مانع از ورود هدر های مشکوک میشود در طرف مقابل IDS میاد محتوای داخل پیام ها را هم نگاه میکند علاوه بر هدر پروتکول IPS علاوه بر گزارش دادن مثل IDS برخلاف اون میاد مسدود هم میکند. پروتکول IPv6: ICMP هم ورژن جدید شده است. دیگر فلگ نداریم چون اصلا وقت نداریم. IPsec: این پروتکول با روتر ها کاری ندارد فقط از طریق انتها به انتها و در میزبان ها ارتباط موجود باشد. اتصال گرا هست و یک نشست برقرار میکنند. دقت کن IPsec کاری به ip ندارد بلکه بسته را که از TCP گرفت رمز نگاری میکند و خودش را داخل یک دیتاگرام IP قرار میدهد و کسی هم متوجه نمیشود.

جلسه 37:

دنبال مسیریابی هستیم نه هدایت. دقت کن first hop router همان روتر پیشفرض ما هست که اگر مقصد در زیر شبکه ما نبود باید به آن بفرستیم. و همین میشود source router ما. اگر ما فرستنده باشیم و اگر گیرنده باشیم همین روتر پیشفرض برای ما میشود destination router. روتر ها گراف ها هستند و یال ها لینک های متصل بین روتر ها هستند. هزینه یک مسیر برابر است با مجموعه هزینه یال های مسیر. دقت کن یک سری هم میگویند کمترین هزینه برابر است با کمترین گام مثلا 3 تا روتر با هزینه 30 نسبت به 4 تا روتر با هزینه 20 بهتر هست ولی مد نظر ما نیست مگر اینکه هزینه یال های را 1 بگذاریم که کمترین مسیر و کمترین گام یکی باشد.

جلسه سی‌هشتم:

LS مشکل متمرکز بودن را دارد یعنی اینکه در یک شبکه بزرگ ما یک لینک کوچک عوض کنیم به کل روتر ها باید اعلام کنیم که این سربار زیادی دارد. DV هم همینطوری هست. یک مسئله دیگه این بود که سازمان ها میخواهند از الگوریتم های مسیریابی مخصوص خودشان استفاده کنند یعنی به سیستم های خود مختار یا به اصطلاح AS نیاز دارند. پس ما اینترنت را به AS های کوچک میشکنیم و هر کدام برای خودشان خودمختاری دارند. و الگوریتم را برای هر AS در نظر میگیریم. لینک های بین روتر ها به صورت P2P هستند. اون روتری که به AS های روتر های بیرونی راه دارد بهش دروازه گفته میشود. جدول هدایت را 2 گروه میسازد یک گروه داخل AS ها هستند که مستقل و متفاوت از هم هستند مثلا RIP یا OSMF و یک سری گروه دیگر داریم که میشود الگوریتم بین AS ها مثل BGP، و دقت کن روتر ها باید بسته را از طریق دروازه ها به سایر AS ها برسانند. دروازه ها میفهمند که زیر شاخه های دیگر کجا هستند و به سایر روتر ها اطلاع میدهند و سایر دروازه ها هم میفهمند و دروازه ها به AS های دیگر هم میرسانند. دقت کنید دروازه ها باید از الگوریتم inter-AS یکسان استفاده کنند همه 20 هزار تا ولی AS ها میتوانند الگوریتم interaAS خودشان را داشته باشند. دقت کن چندین مسیر بین AS ها وجود دارد و الگوریتم های بین مسیر قرار هست بهترین آن ها را برای ما انتخاب کند. الگوریتم مسیریابی درون AS میگوید این بسته که قرار هست به دروازه 1C برسد باید از کدام طرف برود. فرض کن هزینه هر دو تا دروازه به ما در روتر فعلی یکسان باشد حالا از کدوم بریم؟ به اونکه نزدیک تر هستیم برو. به این الگوریتم میگویند hot potato، یا دردسر هست. دقت کن شرایط مقصد مساوی هست نه اینکه ما چون شرق تهرانیم و بخواهیم به تبریز برویم از دروازه شرق خارج شویم. تا 45 اول دیده شد. در BGP اتصالات مش دارند یعنی هر روتر در داخل یک ایالت به صورت منطقی با یک روتر دیگر ارتباط دارد ممکن هست در واقعیت لینک مستقیم بین این 2 نباشد ولی یک اتصال IBGP نیمه پایدار TCP بین آنها باشد. هر AS در دنیا یک شماره سیستم منحصر به فرد دارند. AS-PATH چطوری از حلقه جلوگیری میکند؟ با ASN یعنی نمیگذارد ASN تکراری بیاد داخل و بسته دور ریخته میشود.