Performance: loss, delay, throughput

- پارامتر delay

- وقتی ما یه بسته رو داخل شبکه ارسال می کنیم ، تا به یه نود مقصدی برسه ، این بسته از لینک ها و روتر های مختلفی میگذره و توی این مسیر طبیعتا ما با یه تاخیری روبرو میشیم و اینطور نیست که وقتی یه بسته رو به شبکه ارسال کنیم ، همون موقع به گیرنده برسه .
- برای اینکه این تاخیر end-to-end رو بفهمیم چطور ایجاد شده ، میتونیم روی هر روتر و لینکی که توی این مسیر هست تمرکز کنیم. (از کنار هم قرار گرفتن همه ی این تاخیرها ، تاخیر کل ایجاد میشه)
- وقتی یه بسته ای به یه روتری می رسه، ابتدا باید پردازش(nodal processing) بشه و یه تاخیری در اثر این پردازش ایجاد میشه که بهش میگن d_{proc} .

هدف این پردازش اینه که یه سری کار روی این بسته انجام بشه از جمله اینکه:۱- چک بشه که آیا خطایی در این بسته رخ داده یا نه ، (که خود این کار تاخیر داره) ،۲- باید مشخص بشه با توجه به آدرسی که داخل این بسته هست، کدوم لینک خروجی برای ارسال بسته ازین روتر مناسب هست تا نهایتا بسته به مقصد برسه. و کارهای دیگه ....

اردر تاخیر پردازش از مرتبه ی microsecs (میکروثانیه) هست و چشمگیر نیست، چون پردازش ها توسط ادوات الکتریکی انجام میشن که پیشرفته هستن و روز به روز این تاخیر کم و کمتر میشه.

- یه مولفه ی دیگه ای که تاخیر به ازای یک نود(روتر یا سوئیچ) داره، اینه که بعد از مشخص شدن لینک خروجی ، چون ظرفیت لینک محدوده ، اگه نرخ بسته هایی که میخوان از یه لینک ارسال بشن بیشتر از ظرفیت ارسال اون لینک باشه ، باز یه تاخیری ایجاد میشه و ما باید منتظر بشیم تا نوبت ما بشه و عملیات ارسال انجام بشه.به خاطر همین ، همیشه توی روتر ها یه بافری به ازای هر لینک خروجی وجود داره. به این تاخیر queueing delay(d_{queue}) گفته میشه.

یه نکته ای دیگه اینکه این d_{queue} هیچ فرمول مشخصی نداره و مثلا ممکنه بسته ی ما وقتی به روتر برسه که در اون لینک مشخصی که قراره بسته ی ما ارسال بشه هیچ بسته ی دیگه ای نباشه ،و $d_{queue} = 0$ باشه، یا ممکنه چندین بسته در صف ارسال باشن و d_{queue} مقدارش زیاد باشه.

- هر لینکی یه ظرفیتی داره که با bps اندازه گیری می کنیم. اگه این ظرفیت رو با R نشون بدیم و طول بسته هم L بیت باشه ،وقتی بسته نوبت ارسالش بشه، L/R ثانیه طول می کشه که بیت به بیت این بسته توسط interface روتر روی لینک قرار بگیره. به این تاخیر هم d_{trans} گفته میشه.

- یه مولفه ی دیگه هم وجود داره . هرکدوم از این بیت ها یا دسته ای از بیت ها توسط یه موج الکترومغناطیس ارسال میشن.(بحث جلسه های قبل) سیگنال های الکترومغناطیس هم برای انتشار پیدا کردن احتیاج به زمان دارن، حداکثر سرعت این انتشار توی خلأ $3*10^8$ m/s است ولی توی مدیا های دیگه مثل کابل های برق این سرعت کمتره ، مقدار تیپیکالی که برای محیط های غیر خلأ در نظر می گیرن ، حدود $2*10^8$ m/s است. اگر طول لینک و سرعت انتشار موج رو داشته باشیم میتونیم مدت زمانی که طول می کشه تا موج از فرستنده به گیرنده (در دوسر لینک) برسه رو حساب کنیم. یعنی:

d_{prop} = distance/speed

که d_{prop} همون propagation delay هست.

در نهایت داریم:

$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$

بسته ها تکه تکه میشن و هرکدوم توسط یه سیگنال الکترومغناطیس منتشر میشه ، و هرکدوم ازین سیگنال ها متحمل d_{prop} میشن ، ولی ما نمیایم d_{prop} رو در تعداد این تکه ها ضرب کنیم چون توی این فرمول داریم تاخیر کل بسته رو از زمان رسیدن به روتر تا وقتی که به فرستنده می رسه(طرف دیگه ی لینک) حساب می کنیم و بعضی از این زمان ها با هم overlap دارن و فقط یکی از d_{prop} ها برای محاسبه ی d_{nodal} کفایت می کنه.

- مثال اول ویدیوی سوم راجب اینه که چرا فقد یه بار d_{prop} رو حساب می کنیم.
- مثال کتاب راجب تفاوت d_{prop} و d_{trans} هست . اگه کاروانی از ماشین ها داشته باشیم (مثلا ۱۰ تا) و این کاروان میخواد یه مسیری رو طی کنه.این کاروان رو مثل یه بسته ی ۱۰ بیتی در نظر می گیریم.

این کاروان توی مسیرش به یه سری باجه عوارض گیری برخورد می کنه که باید اونجا متوقف بشه. اول اینکه یه مقداری طول می کشه تا کار هر ماشین با باجه تموم بشه ، دوم اینکه وقتی یه ماشین کارش تموم شد و رفت باجه بعدی، اینجوری نیست که سریعا بهش سرویس بدن و باید منتظر بمونه که سایر ماشین ها به این باجه برسن تا ماشین اول کارش با باجه شروع بشه . این باجه عوارض گیری هم شبیه link transmission هست.

در نهایت کل delay کاروان برابره با :

تعداد * bit transmission time + propagation time

- از این ۴ تا تاخیری که گفته شد از d_{proc} صرف نظر میشه. d_{trans} ممکنه مقدار زیادی داشته باشه به خاطر اینکه هم حجم لینک ممکنه محدود باشه هم تعداد بیت های یه بسته ممکنه زیاد باشه.

d_{prop} اگه طول لینک کم باشه میشه ازش صرف نظر کرد ، ولی اگه طول لینک به صورت قاره ای باشه و یا مثلا شرق و غرب یه کشورو به هم وصل کنه نمیشه ازش صرف نظر کرد.

d_{queue} جنسش اینجوره که پیچیدس و به ترافیک شبکه بستگی داره. علاوه بر اینکه بسته های اپلیکیشن خودمون برای بسته های خودمون ممکنه تاخیر ایجاد کنن ، ترافیک های دیگه و بسته های دیگه هم میتونن واسه بسته های ما تاخیر ایجادکنن . کلا برای این مولفه نمیتونیم فرمولی تعیین کنیم و یه متغیر تصادفیه. همچنین مقدارش بسته به بسته هم میتونه تغییر کنه ، مثلا اگه دوتا بسته طول یکسانی داشته باشن d_{prop} و d_{trans} شون هم یکسانه ولی تاخیر صفشون ممکنه متفاوت باشه .

- اگر دوتا عامل رو در ایجاد تاخیر موثر بدونیم :۱- نرخ بیت ورودی ۲- نرخ بیت خروجی (که به ظرفیت لینک بستگی داره)

و این دوتا عامل در اینکه طول صف چقدر بزرگ باشه موثرن . شبیه این که ما با چه سرعتی سرویس میدیم و چقدر تقاضای سرویس داریم.

- اگه a میانگین نرخ بسته ی ورودی باشه ، L طول بسته ی ورودی (تعداد بیت ها توی یه بسته) باشه (برحسب بیت) و R پهنای باند لینک باشه (bit transmission rate) :

Traffic intensity = L . a / R = arrival rate of bits/service rate of bits

هرچقدر میزان شدت ترافیک(Traffic intensity) بیشتر بشه ، بدتره .

- اگه میزان L . a / R (شدت ترافیک) نزدیک به صفر باشه ، میانگین تاخیر صف هم نزدیک به صفره .

- اگه میزان شدت ترافیک بزرگتر از ۱ باشه ، تاخیر صف خیلی زیاد میشه و به بی نهایت میل می کنه.
- حالا اگه شدت ترافیک برابر با ۱ باشه ، اینکه میانگین تاخیر صف چه مقداره، بستگی به جنس ترافیک داره.

اگه جنس ترافیک bursty باشه ، صف تشکیل میشه ولی اگه منظم و به ترتیب باشه (متناسب با همون نرخ ورودی) صف تشکیل نمیشه.(مثال نونوایی)

- اگه شدت ترافیک بین صفر و یک باشه، اگه شدت ترافیک نزدیک به صفر بشه ،میانگین تاخیر صف هم خیلی کم و نزدیک به صفره و هرقدر شدت ترافیک بیشتر بشه میانگین تاخیر صف به شکل هموگرافیک افزایش پیدا می کنه.(جنس ترافیک رو با فرایند پوآسون مدل سازی کردن). نکته ی مهم توی نمودار رابطه بین میانگین تاخیر صف و شدت ترافیک، اینه که اگه شدت ترفیک به سمت ۱ میل بکنه ، میانگین تاخیر صف هم به سمت بی نهایت میل می کنه.
- یه سری ابزار و وسایل خاصی هستن که با اون ها میشه مقدار تاخیر end-to-end از یک host به یک server رو محاسبه کرد.

مثلا یکی از این ابزار traceroute هست. وقتی آدرس یه host ی رو به این برنامه بدیم ، میاد مسیر بین مبدا و مقصد و trace می کنه و به ازای هر روتر مشخص می کنه چه مقدار تاخیر بین مبدا و اون روتر وجود داره. برای این کار میاد متناظر با هر روتر ، ۳ تا بسته از مبدا تا یه

روتر خاص ارسال می کنه که بتونه میانگین تاخیر رو حساب کنه.(چون همیشه یه مقدار مشخص نیست و تصادفیه)

و بعد این کار رو برای هر روتر در مسیر بین مبدا و مقصد تکرار می کنه.

ما توی هدر بسته ها ، یه فیلدی داریم به اسم (TTL(Time to Live که توی مبدا به این فیلد یه مقداری نسبت داده میشه و بعد هر موقع این بسته به یه روتری می رسه ، اون روتر یکی از TTL کم می کنه و اگر همچنان مقدارش بزرگتر از صفر باشه ، طبق جدول forwarding مشخص می کنه که به کجا باید ارسال بشه، ولی اگه TTL برابر صفر بشه ، دیگه اون بسته رو ارسال نمی کنه و یه پیغام خطا به فرستنده (که آدرسش توی هدر بسته وجود داره) می فرسته . حالا مکانیزم به این صورته که در ابتدا، برای اولین روتر یه بسته می فرسته و قبل از ارسال TTL اش رو برابر با ۱ میذاره، بعد روتر اول مقدار TTL رو صفر می کنه و یه پیغام خطا به مبدا می فرسته و برنامه ی traceroute زمان این رفت و برگشت رو به عنوان یه تخمینی از delay رفت و برگشت تا روتر اول درنظر می گیره و این کارو ۳ بار تکرار می کنه تا درنهایت بتونه میانگین بگیره.

بسته ی بعدی بسته ی چهارمه که برای تاخیر روتر دوم فرستاده میشه. (در واقع اولین بسته از ۳ تا بسته ی دوم ، بسته ی چهارمه). مبدا TTL این بسته رو برابر ۲ قرار میده. این بسته به روتر اول میره، TTL اش میشه ۱ ، بعد به روتر دوم میره و TTL اش میشه صفر و بنابراین یه

پیغام خطا به مبدا می فرسته و این زمان رفت و برگشت به عنوان تاخیر بین مبدا و روتر دوم ثبت میشه.

برنامه ی traceroute این روند رو هی تکرار می کنه تا بسته به مقصد برسه . توی مقصد باز هم بسته دچار یه خطایی میشه ولی از جنس این نیست که TTL برابر صفر شده باشه چون مقصد یه روتر نیست.

حالا مکانیزمی که برنامه ی traceroute استفاده می کنه تا مقصد رو وادار به ارسال پیغام خطا و محاسبه ی تاخیر بین مبدا و مقصد کنه ، چیه؟! (سوال این جلسه 😁)

- توی برنامه ی traceroute گاهی به جای بعضی رکورد ها ستاره نشون داده میشه که این میتونه به خاطر این باشه که کانفیگ روتر ها این طور تنظیم نشده باشه که با صفر شدن TTL پیغام خطا بفرستن. (که البته بیشترشون میفرستن)

  یا ممکنه به خاطر این باشه که جواب ها گم شده باشن.
- ممکنه از یه روتر به یه روتر دیگه جهشی توی تاخیر داشته باشیم ، که این میتونه نشون دهنده ی لینک بین قاره ای (trans-oceanic link) باشه.(قسمت عمده ی این تاخیر هم مربوط به propagation delay هست)
- اگه تعداد روتر ها زیاد بشه لزوما تاخیر زیاد نمیشه و حتی ممکنه کمتر هم بشه ، چون ممکنه شرایط شبکه هنگام ثبت تاخیر یه روتر با روتر دیگه فرق داشته باشه(شرایطی مثل تاخیر صف و ...)

- پارامتر Throughput

- این پارامتر وابستگی به همون delay داره :

Throughput = packet size(bit) / delay(s)

- واحدش bit/s هست و بیت ریت هم بهش میگن و گاهی گذردهی هم ترجمه میشه.

- دوحالت هم داره : ۱- آنی (instantaneous): این شکلیه که یه بسته رو میفرستیم و حجم بسته رو تقسیم بر تاخیرش می کنیم و throughput آنی به دست میاد. ۲-میانگین(average) : حالا اگه از throughput ها در بازه ی زمانی بزرگتری میانگین بگیریم ، throughput میانگین به دست میاد. :/

- برای محاسبه ی محاسبه ی throughput دو روش هست :

1 - روش تقریبی و سرانگشتی (approximate) که به روش Links as Pipes هم شناخته میشه. توی این روش فقط تاخیر transmission رو در نظر می گیریم و روتر ها رو به صورت یه سری نقاط اتصال برای یه سری لوله در نظر می گیریم که این لوله ها جایگزین لینک های شبکه شدن و قطر لوله ها متناسب با ظرفیت لینک هاست. حالا وقتی میخوایم ازین شبکه ی لوله ای سیالی رو عبور بدیم ، اینکه چه مقدار سیال رو میشه از مبدا به مقصد رسوند رو با واحد دبی می سنجن . اما برای بتونیم از شبکه ی لوله ای خود

ساخته به عنوان تقریبی از شبکه های کامپیوتری استفاده کنیم ، گذردهی هر کدوم از لوله هارو با همون واحد bit/s می سنجیم.
با این تعبیر، میتونیم مسئله ی پیدا کردن مقدار throughput رو کاهش بدیم به مسئله ی پیدا کردن اینکه چند بیت میتونیم ازین لوله ها عبور بدیم.
اگه بین host و server دوتا لوله با قطر متفاوت وجود داشته باشه، لوله با قطر کمتر throughput تقریبی میشه.(به اصطلاح بهش میگن bottleneck)
یه نتیجه ی عملی ازین قضیه که توی استفاده ی روزمره ی ما از اینترنت نمود داره ، اینه که مثلا ما با سروری رد و بدل داده داریم که در طرف دیگه ی اینترنت قرار داره و ما اطلاعاتمون از یه شبکه دسترسی که host ما رو به اینترنت وصل می کنه عبور می کنه و بعد اطلاعات ما از طریق core شبکه (ISP) منتقل میشه به اون نقطه ی دوردست و ازونجا هم از طریق یه شبکه دسترسی ما رو به سرور متصل می کنه.
حالا اگه Links as Pipes رو به بحث شبکه ربط بدیم ، اون شبکه ی دسترسی که ما رو به ISP متصل می کنه رو ظرفیتش رو R_c در نظر بگیریم(بعد از مینیمم کردن)، و همینطور برای شبکه ی دسترسی سمت سرور هم ظرفیت لوله رو R_s در نظر بگیریم، اون core اینترنت (که مجموعه ای از شبکه ها هست) رو هم یه لوله ای

با ظرفیت R در نظر بگیریم ، و همچنین طبق توضیح کتاب ۱۰ تا کلاینت به ۱۰ تا سرور هم متصل باشن ، و ظرفیت core اینترنت بین این ۱۰ تا کانکشن تقسیم شده . در این صورت اگه بخوایم end-to-end throughput هر کدوم ازین کانکشن ها رو به طور تقریبی حساب کنیم ، برابره با :

$$Min(R_c , R_s , R / 10)$$

یه نکته ای که در مورد اینترنت صادقه اینه که پهنای باند توی core شبکه خیلی زیاده . معمولا فیبر نوری استفاده می کنیم و پرووایدر ها هم provisioning انجام میدن ،یعنی بیشتر از نیازی که ما –حداقل در حال حاضر– داریم لینک ها ظرفیت دارن.بنابراین ما میتونیم فرض کنیم که R محدود کننده و bottleneck نیست.با اینکه تقسیم بر ۱۰ میشه مقدارش خیلی بیشتر از R_c و R_s هست. بنابراین از این مینیمم گیری سه گانه ما میتونیم $R / 10$ رو کنار بذاریم و بین R_c و R_s مینیمم بگیریم.

از طرف دیگه معمولا سرور ها لینک های دسترسی شون ظرفیت زیاد و خوبی داره (نسبت به کلاینت ها) و حتی بعضی هاشون به tier1 وصلن، بنابراین میتونیم نتیجه بگیریم end-to-end throughput بین سرور و کلاینت ، محدود میشه به ظرفیت شبکه ی دسترسی کلاینت (R_c) .

حالا چی توی این قضیه برای ما ملموسه ؟! اینکه وقتی لینک دسترسی مون رو عوض می کنیم حس می کنیم ارتباط اینترنتمون چقدر خوب شد و این تغییر برای ما مشهوده!

2 - یه روش دقیق هم برای محاسبه ی throughput وجود داره که فرمول اصلی محاسبه ش هست :

$$Delay = L/R_s + d1/s1 + d_{proc} + d_{queue} + L / R_c + d2 / s2$$

➔ Throughput = L / delay

طبق توضیحات قبلی اگه فقط تاخیر transmission رو در نظر بگیریم، داریم :

$$Throughput \simeq \frac{L}{\frac{L}{R_S}+\frac{L}{R_C}} = \frac{R_S R_C}{R_S + R_C} \simeq \min(R_s, R_c)$$

- پارامتر Loss

- دلیل اینکه بسته ها گم میشن دوتا ست :

۱-در انتقال بسته ها روی لینک ها خطایی رخ میده (مثلا بیت صفر به بیت یک تبدیل میشه یا برعکس)

۲-در داخل روتر در error-checking انجام میشه ،اگه خطا داشته باشه روتر اون بسته رو دور می ریزه. و خب بسته گم میشه.

یا برای بسته هایی که میخوان از یه لینک خروجی ارسال بشن ، توی بافری ذخیره میشن و اون صفی که داخل بافر وجود داره، اگر نرخ ورودیش بیشتر از ظرفیتش بشه ، بسته ی جدیدی که میخواد وارد بشه چون جایی برای ذخیره کردنش وجود نداره ،دور ریخته میشه.