

Case studies : Netflix

جلسه ۱۱ :

(بخش)

distribution ویدیو در Netflix :

Netflix از ~~استفاده~~ استفاده می کند و خودش private CDN دارد.

Netflix به وب سایت دارد هر user ی که به اون وصل می شه باید در وب سایت

register شده و پول پرداخت نه و سایت فیلم ها را ببیند و ... (این ها سرورهای

اولیه هستند که هر کس اول وارد می شود بهش اخطا می شود) و کل این جزئیات را از

front-end
back-endcloud
طریق سرورهای امازون انجام می دهد.1) content ingestion
امازون همچنین فیلم های که به Netflix می آید را درون های اصلی آنها را به درون های2) content processing
بسیار دیجیتال تبدیل می کند (رایجترین movie) و در مرحله بعد آن چیزی که

ذخیره شده را در فرمت های مختلفی (bit rate) های مختلف (قابل استفاده در

3) uploading versions to its
DASH) ذخیره می کند و در نهایت CON آن را در CDN های مختلف

Netflix می فرستد پس امازون آمار می کند 1) interface را می سازد.

2) processing movie ها را انجام می دهد.

TALASH

اما خود Netflix تمام چیزهای در CDN ها ذخیره می شوند را پیشبینی می کند

در ادای Netflix از third-party CDN استفاده می کند اما بعدها برای خودش

privat CDN ایجاد کرد. و CDN های خودش را به ISPs ^{بصورت} (brige home)

و هم بصورت ISPs (enter-deep) قرار داد. یعنی هر دوی CDN را دارد

Netflix از ^(آنلاین) ~~pull-caching~~ ^{پیشبینی} ~~pull-caching~~ می کند و بصورت pushing ویدیوهای

خود را distribute می کند و بصورت ~~pull~~ ^{pull} ~~push~~ می کند. (pull & push)

بصورت لحظای Data basis تغییر کند که برای اون time out می بردارند

به طور کلی ~~Netflix~~ از طریق اون software که در آن قرار هست Netflix

می سبیه می کند که از تمام CDN node ^{خود Netflix} استفاده بشود. سوال: CDN node چگونه انتخاب می شود؟

قبل از این که user بخواهد با CDN ارتباط برقرار کند، اون مقترقی در آن قرار

CDN های داخل شبکه ارتباط برقرار می کند. اول به اندازه گیری انجام می شود و بعد ^{node}

سرور را مشخص کند.

نکته ای وجود دارد NetFlix که مستقیماً CDN را مشخص می کند و manifest برای ارسال می کند تا بتواند از طایفه کیفیت مختلف استفاده کند.
به طور کلی NetFlix از DASH استفاده می کند.

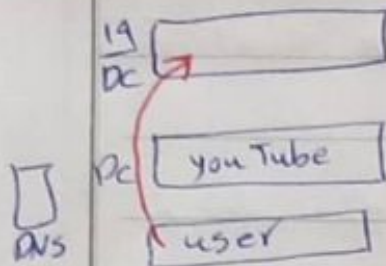
- * طول chunk دارد NetFlix می رانند است.
- * همچنین از الگوریتم rate-determination استفاده می کند.
- * redirect ← DNS را انجام می دهد.

Case studies : YouTube

- YouTube بزرگترین Database ویدیویی را دارد و از ویدیو گرفته ویدی از تبلیغات پول درمی آورد.
- اما NetFlix از ویدیوهای خود مستقیماً پول درمی آورد.
- و فضای YouTube ← content است.
- خودش CDN دارد. هر Isp مقرر می گیرند.
- * YouTube ← processing ویدیوها را در Peta center های (۱۹) بودن
- انجام می دهد. ویدی در enter-deep ها قرار می گیرند.

تفاوت NetFlix : ① pull-caching انجام می دهد و push انجام نمی دهد.

- ② DNS direct انجام می دهد. یعنی اول CDN مربوط به Isp ها را خودشون می کشند. CDN مجزا هستند و DNS مجزا دارند و وقتی که اول درخواست میاد revolt (طغیان) می کنند و روی DNS.



YouTube وقتی user درخواست میدهد سرانجام 19، وادون
19 براساس چه url ای بوده، درخواست redirect می کند
به DNS مربوط به Data Center & you Tube.

Net Flux: در Net Flux 19 خودش می تواند به هر کدام از DNS ها وصل بشه.

کار دیگری که YouTube انجام میدهد اینست: گوگل cluster-selection انجام می دهد

براساس RTT که تعیین می کند بین کلاس و کلاس دیگر و معمولاً به این ترتیب RTT

را دارد به عنوان سرور انتخاب می کند. مثلاً اینست بهینه node زبانی برای اول سرور هست...

YouTube نرمه از DASH استفاده نمی کند. و از HTTP streaming استفاده می کند.

3 نوع ارسال و دریافت داریم: ① udp streaming: یک ورژن خیلی ساده است که فقط IP می فرستد. ② HTTP streaming: خیلی مبتنی بر TCP است و به اصطلاح adaptation وجود ندارد که بخواهد نرخ های مختلف را انتخاب کند. ③ DASH.

YouTube از HTTP streaming استفاده می کند و معمولاً به quality انتخاب را برای stream کردن ویدیو ها.

Congestion control : خیلی به Transport layer نزدیک است .

و به طور ویژه Tcp پروتکل .

و می تواند به صورت Class layer در کنار لایه های مختلف انجام بشه . و صرفاً یک Panction ← Transport layer نباشه . و می تواند بخشی

از action هاش در link layer انجام بشه .

اما منشأ اصلی آن در Transport layer است و منطقی و متنا از طریق Tcp انجام می شود .

provide logical communication

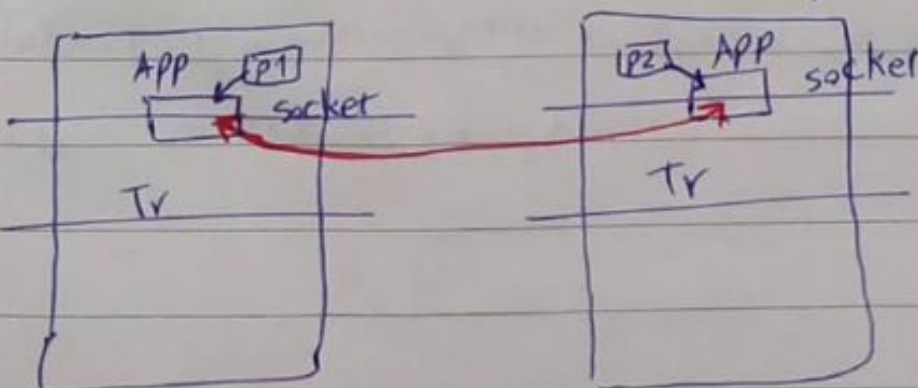
در ارتباط بین دو Host در شبکه ، باید به node access وصل بشه و بعد

node are وصل بشه و بعد node are با زیر ساخت شبکه صحبت می کنه

و بعد به همین ترتیب به یک Host دیگه برسه . و هدف اینم که ارتباط logical

صورت بگیره و یک پروتکل بین این دو run بشه .

و یک ارتباط منطقی بین proccese های در application layer برقرار می شود ، این کار را Transport layer انجام می دهد .



و اینکه با چه کیفیتی ^{انجام} ~~می‌شود~~ بر، مربوط به ^{معمولاً} ~~معمولاً~~ ساختن Transport layer می‌شود.

در لایه App، پیام‌های، ابعادی شوند Transport بین پیام‌ها را به شکل segment

تبدیل می‌کند و ارسال می‌کند به Network layer، و بعد Network layer

پیام‌ها را ارائه می‌کند و بعد ساخت پیوندها آنها را می‌گیرد و assemble (ارسال) می‌کند.

و نتایج می‌تواند و پیوند های Transport TCP > UDP

اتفاقات ~~مهم~~ Transport، فرستنده message را Transport می‌دهد.

و App با توجه به اینکه چه پروتکلی را می‌خواهد به آن که heder اضافه می‌کند و ساخت

را ابعادی می‌کند و segment را به IP layer می‌دهد و - Net آن را به شبکه ارسال می‌کند (Network)

پیونده: segment را از لایه IP می‌گیرد بر اساس پروتکل که هست heder

را remove می‌کند و بر اساس پارامترهای ^{در} socket ارسال می‌کند.

* اتفاق مهم: که demultiplexing رخ می‌دهد.

TCP: (Transmission control protocol)

- reliable, in-order delivery
- congestion control → ممکن است ارسال پیام flexible نباشد.
- flow control
- connection setup

UDP: (user Datagram protocol)

- unreliable, unordered delivery
- No-frills extension of "best-effort" IP

* برای جلوگیری از ازدحام در UDP استفاده می شود (به خاطر نداشتن congestion control)

* ۲ ضمیمه ارائه نمی شود: delay guarantees و bandwidth

کنترل ترافیک

Multiplexing and Demultiplexing

در هر لایه ها انجام می شوند، به سببی دنیا که در لایه های بالای ترانزیت می شوند و به نوبه به شبکه بین Data ها به یک خطی قراره تقسیم شود یا در لایه پایین چگونه می خواهد جمع شوند انجام می شوند.

بهترین چیزی که در شبکه انجام می شود — Demultiplex است. و به معنی این است

که یک سری دنیا را در بین socket های مختلف تقسیم می کند.

De multi : فنون سیستمی

Apache for HTTP server

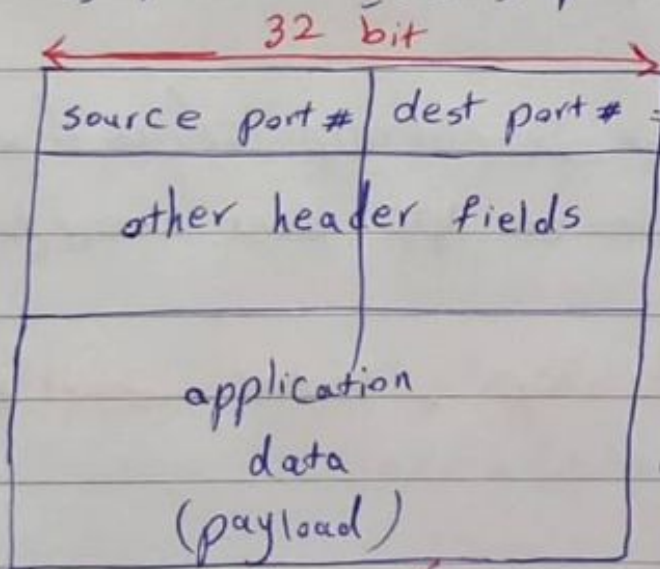
موجود دارد و message های HTTP را به چند user می فرستد.
وزمانی را این پیام ها به Transport رسیده در کنار segment های برای Netflix و S است و باید جدا شوند و به سمت کاربر فاکس برود.

De multi : توجه به مشخصه های موجود دارد باید جدا سازی را انجام بدهد.

آنها از این مشخصه ها : Ip addresses و port numbers است

که این دو روی هم socket را می سازند.

و در هر سیستم TCP که port number فرستاده و یک port گیرنده داریم.



بر اساس اینها De multi را انجام می دهد

در TCP ← IP فرستاده و گیرنده

همه اینها - اما در UDP همه نیست.

و در UDP فقط به dest port نگاه می کنند.

در TCP به هر مشخصه

socket دقیقاً مشخص می کنند.

اما در UDP دیکه وقتی بسته را فرستاده

دیکه نگاه نمی کنند برای چه کسی برده است!

فرست می

Tcp/udp

UDP: وقتی socket مشخص می شود ۲ تا چیز نیاز هست:

۱, destination IP address ۲, destination port #

وقتی فرستنده Host به UDP را دریافت کرد destination port #

را حلی می کند و به اساس آن به پیام را به socket مربوط به UDP ارسال می کند.

نکته ای که وجود دارد این است: ممکن است تعدادی Datagram وجود داشته باشد

که از سمت لایه IP وارد شده باشد و یکسری source IP address متفاوت

دارند. باید به destination port به ارسال شوند.

* کارهایی که در UDP انجام می شود بیشتر اوقات در لایه App کارهای با کیفیت ترش
رنگ می دهد. و برای App های خاص UDP استفاده می شود.

* انتخاب socket در TCP: ۴ پارامتر وجود دارد:

- source IP Address

- " port number

- dest IP address

- dest port number

Multit, De multi براس segment, datagram انجام می شود.

در UDP چیزی مهم است # destination port است (only)
در TCP هر ۴ تا مشخصه اهمیت دارد.
و در همه اینها استقاف می‌افتد.

UDP: یک "best effort" سرویس است - lost (گم شدن) در آن وجود دارد.
و delivered out-of-order to app وجود دارد.
- هیچ handshaking در آن وجود ندارد (پس فرستنده و گیرنده)
- سگمنت‌ها متفاوت از هم مدیریت می‌شوند.

* چرا UDP؟

- connection establishment نیاز ندارد و این کار باعث
می‌شود که RTT حذف می‌شود.

- ساده است. no connection state at sender, receiver.

- header size خیلی کوچک است.

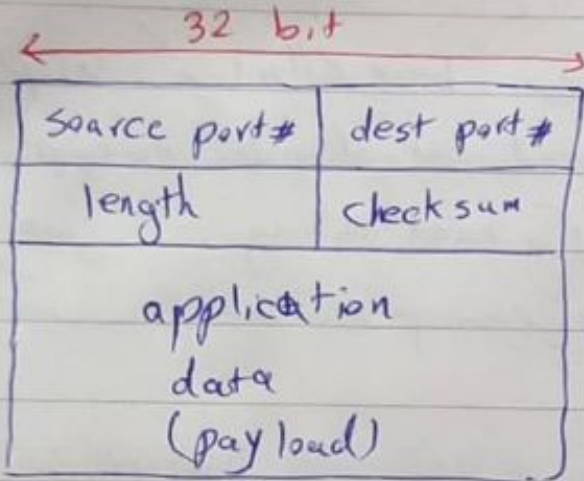
- no congestion control

* در کجاها UDP استفاده می‌شود؟

- Streaming multimedia (loss tolerant, rate sensitive)
- DNS
- SNMP
- HTTP/3

App کی reliability : HTTP/3 میں بھی ہے

- add needed reliability at App layer
- add congestion control at App layer



UDP segment format

Transmitted:	2	4	11	checksum
received:	2	4	≠ 11	

وقتی L1, L2, L3 میں سے کسی میں error ہے تو یہ ہے۔