



IS
2nd
material

Network





بسم الله والحمد لله والصلاة والسلام على رسول الله

عن هذا الملخص:

هذه تكملة المحاضرة الفائتة بعنوان "chapter 2: Application layer" وفي المرة السابقة ذكرنا:
Principles of network applications
Web and HTTP بعض منه
وهذه المرة نكمل مع المتبقي من Web and HTTP و

- E-mail, SMTP, IMAP
- DNS
- P2P applications
- video streaming and content distribution networks
- socket programming with UDP and TCP

Web and HTTP

web caching

(web cache) هو جهاز يكون بين جهازك (client) والجهاز الرئيسي (server) في مكان قريب من جهازك (local) يرسل له جهازك الطلبات (request messages) بدلا من الجهاز الرئيسي. وإذا كان عنده ما تطلبه أرجعه إليك مباشرة أما إذا لم يكن لديه: يرسل إلى الجهاز الرئيسي يطلب منه البيانات المطلوبة ثم بعد أن تصله يرسلها إلى جهازك (response message)... لاحظ التالي:

1. Web cache acts as both **client and server**:
 - server for original requesting client
 - client to origin server
2. server tells cache about object's allowable caching in response header:

Cache-Control: max-age=<seconds>

Cache-Control: no-cache

3. Why Web caching?
 - reduce response time for client request: cache is closer to client
 - reduce traffic on an institution's access link
 - enables "poor" content providers to more effectively deliver content

النقطة الثانية معناها أن الجهاز الرئيسي هو من يحدد لجهاز ال (web caching) ما هي مدة صلاحية البيانات التي سيخزنها. متى سيحتاج إلى طلبها من الجهاز الرئيسي مرة أخرى. أو حتى يمكن أن لا يسمح له أصلا بتخزين أي بيانات في بعض المواقع.

في شرائح (slides) المحاضرة من الصفحة 44 حتى 47 مثال يوضح كيف أن شراء جهاز web caching في مؤسسة سيوفر كثيرا وسيكون سريعا أكثر حتى من شراء توصيلة (cable) أسرع وأعلى ...

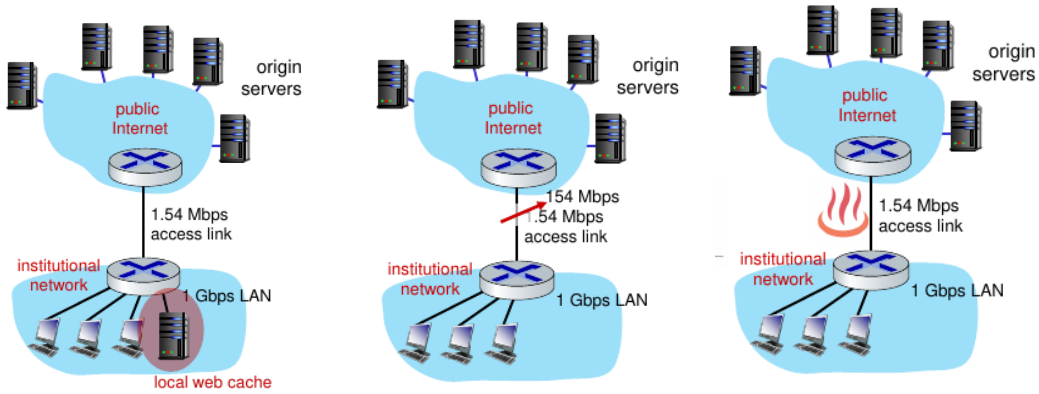
المثال يقارن بين ثلاث حالات: الحالة العادية، وحالة شراء توصيلة أسرع للانترنت (access link)، وأخيرا حالة شراء (web cache) ويقول بأن الزمن الكلي لنقل البيانات في الثلاث حالات =

internet delay (RTT) + access link delay + LAN delay

الذي يختلف في الثلاث حالات هو ال (access link delay) فيكون أقل ما يكون عندما نستخدم (web caching) ثم شراء توصيلة انترنت أسرع ثم الحالة العادية بدون شراء شيء ...

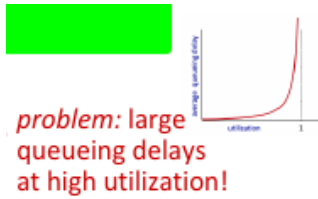
افترض أن المعطيات هي كالتالي:

- access link rate: 1.54 Mbps (أعلى سرعة لتوصيلتك بالانترنت)
- RTT from institutional router to server: 2 sec (HI !)
- web object size: 100K bits
- average request rate from browsers to origin servers: 15/sec



بما أن حجم (web object) واحد = 100K ويوجد متوسط حوالي 15 في الثانية الواحدة إذا الحجم الكلي في الثانية الواحدة = $100K * 15/sec$

1.5 Mps (1500K/sec)



بما أن هذا الحجم قريب جدا من أعلى سرعة توصيل انترنت لديك إذا ستحس ببعض البطء ولكن يمكنك شراء توصيلة انترنت أسرع وسيحل هذا الأزمة لكن ستدفع كثيرا ... أخيرا يمكنك باستخدام ال (web cache) أن تدفع قليلا وفي نفس الوقت هي أسرع من شراء توصيلة انترنت باهظة !

أنصحك أيها القارئ بالتأكد من فهمك للمسألة من شرائح المحاضرة

المتصفح الخاص بك (browser) يستخدم أيضا تقنية ال (caching) ولكنه **دائما** يرسل للجهاز الرئيسي (web server) يسأله هل البيانات التي لديه مطابقة لما عند المتصفح (<If-modified-since: <date>) ؟ إذا كان الأمر كذلك يستخدم البيانات المخزنة في ال (cache) عندك على الجهاز (HTTP/1.0 304 Not Modified) والعكس صحيح

HTTP/2 and 3

هناك نوعان آخران من ال (HTTP) غير 1 و 1.1 وهما النوع 2 [RFC 7540, 2015] والنوع 3 ... النوع 2 فيه زيادة عن النوع 1.1 أنه لا يرسل كل (object) بعد الآخر إنما يقسمهم ويرسل جزءا من كل (object) ثم يرسل جزءا آخر من كل (object) وهكذا حتى تتم العملية بنجاح (مثل: round robin)

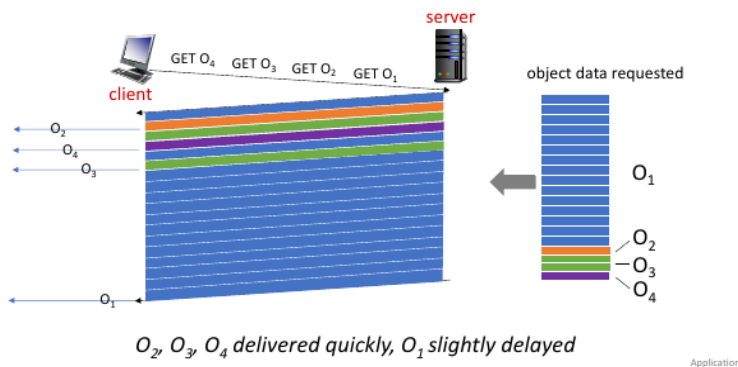
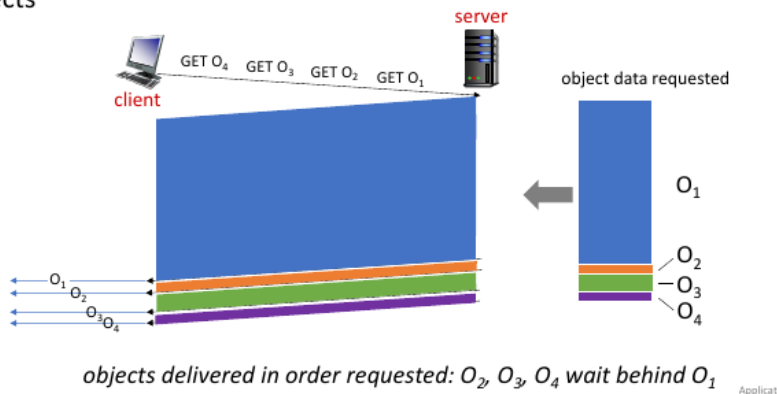
أما النوع 3 فهو يزيد على النوع 2 في ثلاث نقاط:

1. security
2. per object error control



3. congestion control (more pipelining) over UDP

ects



الرسم الأولى غير المقسمة هي ل (HTTP 1.1) والثانية ل (HTTP 2) ... لاحظ الفرق ... هنا بعض الملاحظات على النوع 2:

- transmission order of requested objects based on client-specified object priority (not necessarily FCFS)
- push unrequested objects to client
- divide objects into frames, schedule frames to mitigate head of line (HOL) blocking

EMAIL, SMTP, IMAP

تتكون خدمة الإيميل مثل باقي الخدمات من جهازك (UA - user agent) والجهاز الرئيسي (server) و (protocol) يسمى (SMTP)

- UA are mail readers
- mail servers work as mailboxes (for incoming messages) and as message queue (for outgoing messages)
- the SMTP protocol do two things: sends UA message to the server and send messages between the servers

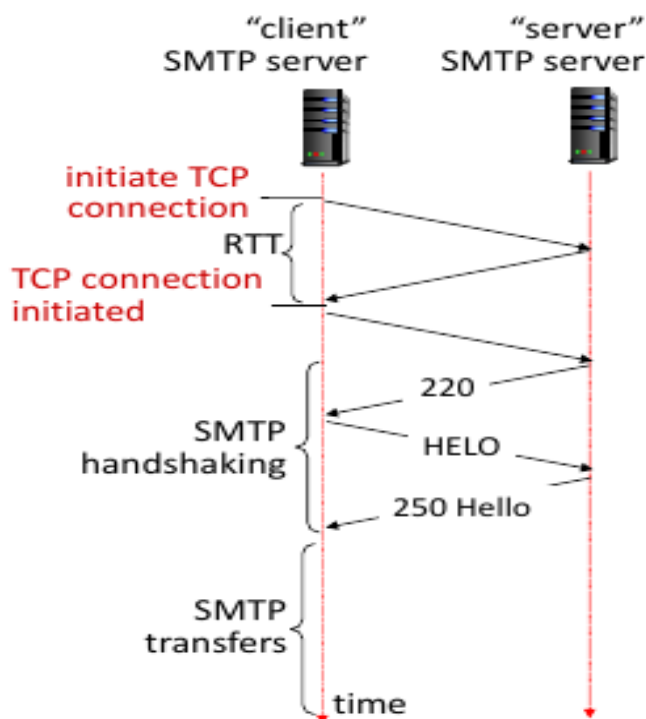
هذه الجزئية ذكرت في المحاضرة وليست في الشرائح:



after the message arrives at the closest server to the (destination UA) the used protocol is called (IMAP) not (SMTP)

يستخدم SMTP (أو RFC 5321) أيضا (TCP) تحديدا على (port 25) ويمكن أن يرسل من server إلى آخر كما ذكرنا سابقا ... ويمر بثلاث مراحل:

- SMTP handshaking (greeting)
- SMTP transfer of messages
- SMTP closure



موضحة في الرسمة المقابلة:

لاحظ أن الكود 220 يستخدم كمؤشر أن

ال (SMTP handshaking) تم بنجاح

وبعده (HELO)

ثم (Hello 250)

SMTP vs. HTTP

HTTP: client **pull**

- SMTP: client **push**
- both have **ASCII** command/response interaction, status codes
- HTTP: **each object** encapsulated in its own response message
- SMTP: **multiple objects** sent in multipart message

- SMTP uses **persistent** connections

- SMTP requires message (header & body) to be in

7-bit ASCII

- SMTP server uses **CRLF**.CRLF to determine end of message (control line feed)



لاحظ أن استخدام ال(SMTP) لل(ASCII) معناه أنه فقط يكتب بالانجليزية. إذا أردت استعمال لغة أخرى عليك استعمال (protocol) آخر مع ال(SMTP).

احفظ هذا قد تحتاجه في الامتحان

1. RFC 5321 protocol -> SMTP
2. RFC 7231 protocol -> HTTP
3. RFC 2822 protocol -> the syntax of the email message itself (header, blank line, body)
4. RFC 3501 protocol -> IMAP (internet mail access protocol)

تطبيقات الإيميل التي توجد على الويب مثل gmail تستخدم HTTP بشكل عادي لأنها صفحات ويب لكن تستخدم لإرسال الإيميلات SMTP ولاستقبال البيانات IMAP

DNS (domain name server)

مجموعة أجهزة تكون (distributed database) تحمل اسم (name) وال(IP) الخاص بكل موقع وظائفه كالتالي:

1. hostname-to-IP-address translation (كما ذكرت)
2. host aliasing (يمكن أن يعطي اسما مستعارا)
 - canonical, alias names
3. ▪ mail server aliasing
4. ▪ load distribution
 - replicated Web servers: many IP addresses correspond to one name

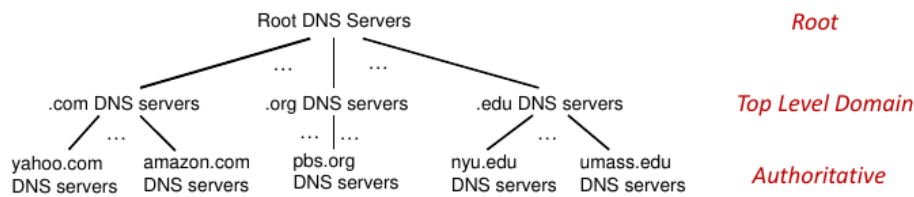
لماذا لا تتركز (DNS database) في جهاز واحد ... لماذا هي (distributed)؟

1. single point of failure (إذا حدث خلل سيقع كل النظام)
2. traffic volume will be huge
3. distant centralized database (بعيد مكانيا)
4. maintenance (صيانة أصعب)

DNS servers are complex ... just think about the **billions** of records it contains, the **trillions** of queries it receives and each must be very fast, and the **millions** of organizations that are responsible for their records.



DNS: a distributed, hierarchical database



Client wants IP address for **www.amazon.com**; 1st approximation:

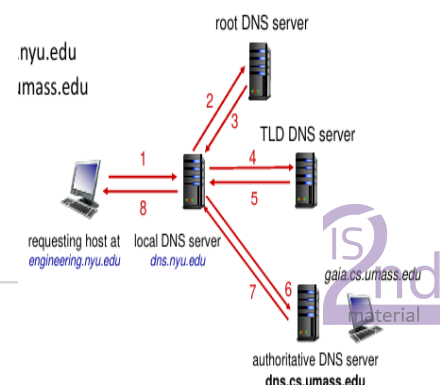
- client queries root server to find .com DNS server
- client queries .com DNS server to get amazon.com DNS server
- client queries amazon.com DNS server to get IP address for www.amazon.com

From the illustration above:

1. **Root** name servers (DNS) are very important and they are managed by **ICANN** (internet corporation for assigned names and numbers)
 - the **DNSSEC** (domain name system security extensions) provides security for DNS
 - there are 13 logical root name servers worldwide
2. **Top-level domain** servers (TLDs) is responsible for:
 - a. .com, .org, .net, .edu, .aero, .jobs, .museums,
 - b. all top-level country domains: .cn, .uk, .fr, .ca, .jp
 - c. Network Solutions: authoritative registry for .com, .net TLD
 - d. Educause: .edu TLD
3. **Authoritative** DNS servers:
 - a. organization's own DNS server(s), providing authoritative hostname to IP mappings for organization's named hosts
 - b. can be maintained by organization or service provider

Local DNS name servers

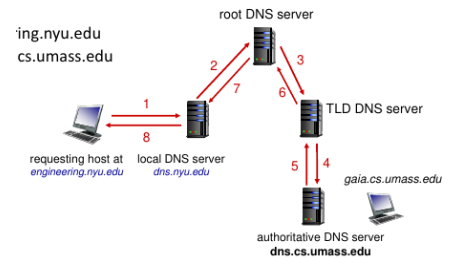
1. when you try to access a website, the request is sent in the following order to be resolved :
 - your host -> Local DNS -> DNS hierarchy
2. each ISP has it's local DNS to find yours:
 - a. • MacOS: % scutil --dns
 - b. • Windows: >ipconfig /all
3. local DNS server **doesn't strictly** belong to hierarchy
4. there are two ways to do **DNS name resolution**:
 - a. iterated query: your message goes from your host to the local DNS server then
 - i. from local DNS to root DNS then returns to local DNS again





- ii. from local DNS to TLD DNS then returns to local DNS again
- iii. from local DNS to authoritative DNS then returns to local DNS again
- iv. from local DNS to your host

- b. recursive query: your message goes from your host -> Local DNS -> Root DNS -> TLD DNS -> authoritative DNS ... then the message is returned again in the same path from authoritative DNS to host.



5. ال DNS يستخدم ال (caching) وبعد فترة معينة تسمى (TTL)

تنتهي صلاحيته ... عادة ما تخزن أسماء (TLD) في ال (local server)

DNS: distributed database storing resource records (RR)

RR format: (name, value, type, ttl)

name	value	type
hostname	IP address	A
domain	hostname of authoritative name server for this domain	NS
canonical (alias)	real name	CNAME
name of SMTP mail	name	MX

ال (DNS) رسائل متطابقة سواء query أو reply ... ليس مثل HTTP الذي تختلف شكل الرسائل فيه إذا كنت sender عن إذا كنت receiver وهذا شكلها

← 2 bytes →	← 2 bytes →
identification	flags
# questions	# answer RRs
# authority RRs	# additional RRs
questions (variable # of questions)	
answers (variable # of RRs)	
authority (variable # of RRs)	
additional info (variable # of RRs)	



ال (DNS) قد يحصل عليه هجمات منها:

1. DDoS (denial of service): ارسال الكثير جدا للجهاز الرئيسي حتى يتوقف تماما
2. Spoofing attacks: اختراق الاتصال بالجهاز والرد بموقع آخر

P2P applications

أي جهاز يمكن أن يرسل أو يستقبل رسائل من أي جهاز آخر على عكس ال (client/server model) وأمثلته:

- file distribution (BitTorrent)
- streaming (Kankan)
- VoIP (Skype)

الوقت الذي يأخذه نشر ملف معين في شبكة تستخدم ال (client/server model):

$$\text{time to distribute } F \text{ to } N \text{ clients using client-server approach} \quad D_{c-s} \geq \max\{NF/u_s, F/d_{min}\}$$

سرعة أبطأ جهاز حجم الملف سرعة رفع الملف

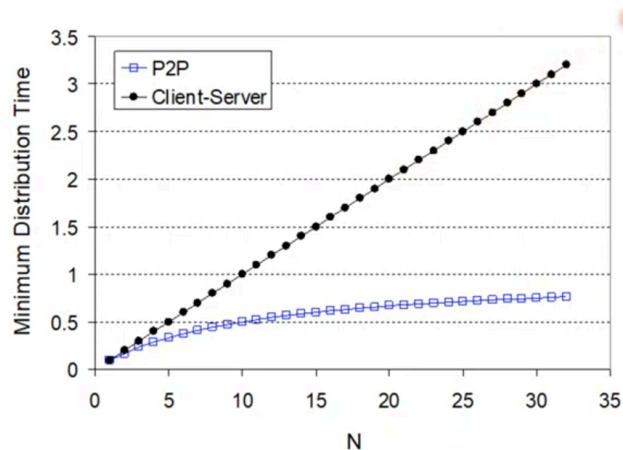
- طبعا كلما زادت (N) كلما زاد (NF/u) في علاقة (linear)

الوقت الذي تأخذه نفس العملية في شبكة من نوع (peer to peer)

$$\text{time to distribute } F \text{ to } N \text{ clients using P2P approach} \quad D_{P2P} \geq \max\{F/u_s, F/d_{min}, NF/(u_s + \sum u_i)\}$$

سرعة رفع الملف من الجهاز الرئيسي سرعة تنزيل أبطأ جهاز في الأجهزة العادية سرعة رفع الملف من الأجهزة العادية

مقارنة بين السرعتين النوعين P2P vs. client/server:





كلما زاد عدد الأجهزة في الشبكة يزيد الوقت المطلوب لنشر الملف بينهم بشكل (linear) في حالة (client/server) أما في حالة (p2p) كلما زاد العدد يزيد الزمن المطلوب إلى نقطة معينة بعدها لا يزيد الزمن كثيرا (لأنه بزيادة الأجهزة تزداد سرعة رفع الملفات + يتطلب الأمر سرعة تنزيل أسرع أيضا في نفس الوقت)

من أمثلة استخدام (P2P) برامج BitTorrent:

برامج هدفها نقل البيانات بين المستخدمين تقوم بتقطيع هذه الملفات لأجزاء (265 chunks Kbytes) وتقوم بتبادلها باستخدام طريقة (tit-for-tat). هذه الطريقة تجعل المستخدم يفتح سرعة التنزيل القصوى لأربعة أجهزة هو يأخذ من عندهم ملفات وهم يأخذون من عنده ملفات. وكل 10 ثواني نقيم الأربعة أجهزة الأفضل (الأسرع) مرة أخرى وكل 30 ثانية نختار بشكل عشوائي جهازا غير الأربعة من الشبكة إذا كان أسرع من أحد الأجهزة الأربعة نبدله بأبطأ الأجهزة الأربعة. الأجهزة في هذه العملية يعبر عنها بكلمة (churn) لأنها تدخل وتخرج من الشبكة سريعا

Video streaming and CDNs

بث الفيديو (video streaming) من أكثر ما يستهلك الانترنت ويواجه تحديات كثيرة منها:

1. كيف نستطيع بث الفيديو للأعداد الهائلة على الانترنت (scale)
2. كيف نتعامل مع اختلاف ظروف المستخدمين (heterogeneity)
3. في حالة (streaming stored videos) نواجه (client interactivity)
4. يمكن أن نفقد packets

الحل لهذا هو استخدام (distributed application-level infrastructure)

لتقليل حجم الفيديو المرسل نستخدم طريقتين

1. استغلال تكرار الألوان في نفس الصورة (spatial encoding)
2. استغلال تكرار الألوان من صورة لأخرى (temporal encoding)

إذا كان ال (encoding rate) ثابتا يسمى (CBR = constant bit rate) وغير ذلك إذا كان مقدار ال (spatial and temporal encoding) يتغير يسمى (VBR = variable bit rate)

عند ال (client) يكون هناك مساحة تخزينية (buffer) للفيديوهات المرسله. عندما يستقبل جهازك جزءا منها

من أهم وسائل عمل بث للفيديوهات streaming: استخدام (DASH = dynamic adaptive Streaming over HTTP)

- يقطع الفيديو أجزاء (chunks) ثم يرسلهم ويستخدم (manifest file: provides URLs for different chunks)
- لاحظ: ("intelligence" at client)
- بث الفيديوها: Streaming video = encoding + DASH + playout buffering

content distribution networks (CDNs):

1. إذا أراد مستخدم أن يشاهد فيديو على الشبكة نحتاج الى استخدام إحدى طريقتين لكي نبث إليه هذا المحتوى من (geographically distributed sites)
 - a. enter deep: نجعل كل جهاز قريبا من جهاز رئيسي akamai company



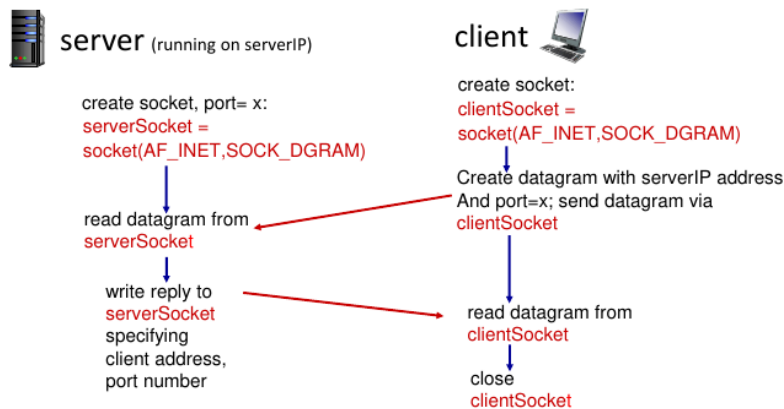
- b. bring home: smaller number (10's) of larger clusters in POPs near access nets -> Limelight company

content distribution networks are (OTT = over the top) of internet as a service.

Socket programming

I won't talk about TCP and UDP now as they are the topic of the next lecture but maybe these two images will clear the topic:

Client/server socket interaction: UDP



Client/server socket interaction: TCP

