

# Exercise session 2

## Application Layer

Lakiotakis Emmanouil  
Spring 2017

# Exercise 1

Υποθέστε ότι μέσα στον web browser κάνετε κλικ σ' έναν δεσμό για να πάρετε μία ιστοσελίδα. Η διεύθυνση IP για το σχετικό URL δεν βρίσκεται στην cache στον τοπικό σας υπολογιστή, οπότε είναι απαραίτητη μία αναζήτηση DNS για να πάρετε την διεύθυνση IP. Υποθέστε ότι γίνονται επισκέψεις σε  $n$  DNS servers πριν πάρει ο υπολογιστής σας την διεύθυνση IP από το DNS, οι διαδοχικές επισκέψεις γίνονται σε  $RTT_1, \dots, RTT_n$ . Ακόμη, υποθέστε ότι η ιστοσελίδα που σχετίζεται με την ζεύξη περιέχει ακριβώς ένα αντικείμενο, αποτελούμενο από μία μικρή ποσότητα κειμένου HTML. Έστω ότι το  $RTT_0$  συμβολίζει το RTT ανάμεσα στον τοπικό υπολογιστή και στον server που περιέχει το αντικείμενο. Υποθέτοντας μηδενικό χρόνο μετάδοσης του αντικειμένου, πόσος χρόνος παρέρχεται από την στιγμή που ο πελάτης κάνει κλικ στον σύνδεσμο μέχρι την στιγμή που λαμβάνει το αντικείμενο;

# Exercise 1 - Solution

Για την εύρεση της IP διεύθυνσης απαιτείται χρόνος

$$RTT_1 + RTT_2 + \dots + RTT_n \quad (1)$$

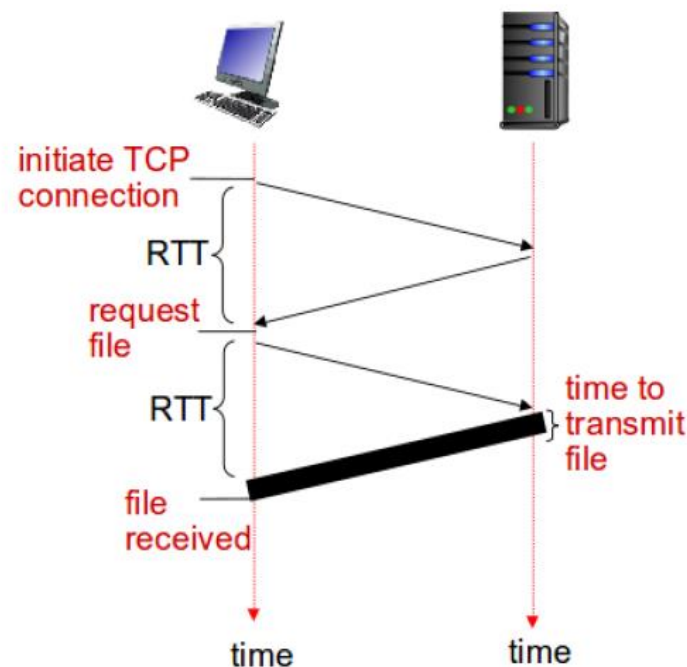
Για την λήψη του αντικειμένου:

- $RTT_0$  χρόνος για το εγκαθίδρυση της TCP σύνδεσης (2)
- $RTT_0$  χρόνος για την διαδικασία request-reply του μικρού αντικειμένου (από την εκφώνηση ο χρόνος μετάδοσης θεωρείται αμελητέος) (3)

Άρα από (1), (2), (3) **συνολικά** απαιτούνται:

$$2 \times RTT_0 + RTT_1 + RTT_2 + \dots + RTT_n$$

χρόνος για την λήψη του αντικειμένου



# Exercise 2

Αναφερόμενοι στο προηγούμενο πρόβλημα (1), υποθέστε ότι το αρχείο HTML αναφέρεται σε **οκτώ** πολύ μικρά αντικείμενα στον ίδιο server. Αγνοώντας τους χρόνους μετάδοσης, πόσος χρόνος παρέρχεται με

a) **μη παραμένουσες συνδέσεις HTTP χωρίς παράλληλες συνδέσεις TCP;**

Solution:  $RTT_1 + \dots + RTT_n + 2 \times RTT_0 + 8 \times 2 \times RTT_0$

b) **μη παραμένουσες συνδέσεις HTTP με τον browser ρυθμισμένο για 5 παράλληλες συνδέσεις;**

Solution:  $RTT_1 + \dots + RTT_n + 2 \times RTT_0 + 2 \times 2 \times RTT_0$

c) **παραμένουσες συνδέσεις, θεωρώντας μη πεπερασμένο πλήθος συνδέσεων;**

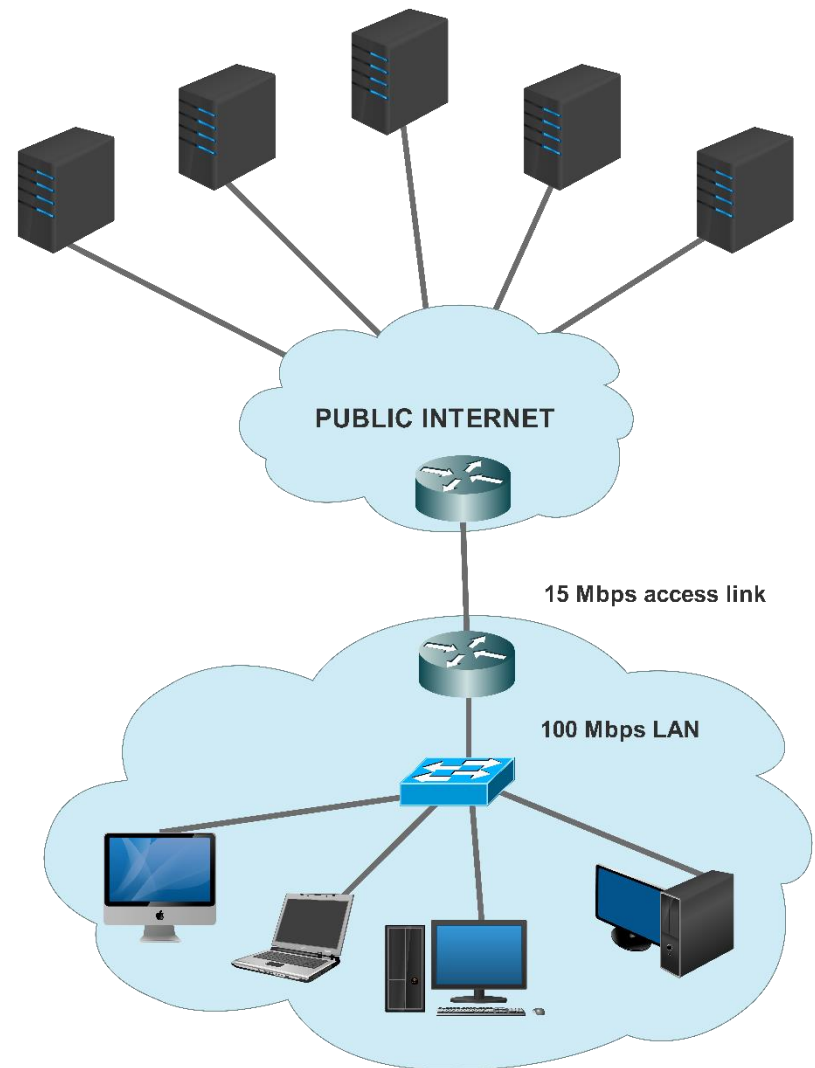
Solution:  $RTT_1 + \dots + RTT_n + 2 \times RTT_0 + RTT_0$

# Exercise 3

Δείτε την παρακάτω εικόνα, στην οποία υπάρχει ένα ιδρυματικό δίκτυο συνδεδεμένο στο Διαδίκτυο. Υποθέστε ότι το μέσο μέγεθος αντικειμένου είναι **850.000 bits** και ότι ο μέσος ρυθμός αιτήσεων από ιδρυματικούς browsers μέχρι τους origin servers είναι **16 αιτήσεις** ανά δευτερόλεπτο. Επίσης, υποθέστε ότι ο χρόνος που χρειάζεται από την στιγμή που ο δρομολογητής στην πλευρά του διαδικτύου της ζεύξης προσπέλασης προωθεί μία αίτηση HTTP, μέχρι την στιγμή που λαμβάνει την απόκριση είναι **3 δευτερόλεπτα** κατά μέσον όρο. Μοντελοποιήστε τον συνολικό μέσο χρόνο απόκρισης ως το άθροισμα των μέσων καθυστερήσεων προσπέλασης (δηλ. την καθυστέρηση από τον δρομολογητή του διαδικτύου μέχρι τον ιδρυματικό δρομολογητή) και την μέση καθυστέρηση διαδικτύου. Για την μέση καθυστέρηση προσπέλασης, χρησιμοποιήστε το  $\frac{\Delta}{1-\Delta \times \beta}$ , όπου  $\Delta$  είναι ο μέσος χρόνος που απαιτείται για αποστολή ενός αντικειμένου μέσω μιας ζεύξης προσπέλασης και  $\beta$  ο ρυθμός αφίξεων αντικειμένων στην ζεύξη προσπέλασης.

# Exercise 3 – Solution

- 16 requests ανά δευτερόλεπτο  $\rightarrow \beta$
- Χρόνος από το HTTP request (router) μέχρι το response  $\rightarrow$  3 seconds
- Μέσο μέγεθος αντικειμένου  $\rightarrow$  850000 bits
- Μέσος χρόνος που απαιτείται για αποστολή ενός αντικειμένου μέσω μιας ζεύξης προσπέλασης  $\rightarrow \Delta$



# Exercise 3 - Solution

a) **Βρείτε το συνολικό μέσον χρόνο απόκρισης**

Solution: Γνωρίζουμε από εκφώνηση ότι ο συνολικός μέσος χρόνος απόκρισης είναι,  $t_{\text{Total}}$  ισούται με:

$$t_{\text{Total}} = t_{\text{μέση καθυσ προσπ}} + t_{\text{μέση καθυσ δικτύου}} \quad (1)$$

με

- $t_{\text{μέση καθυσ προσπ}} = \frac{\Delta}{1 - \Delta \times \beta} \quad (2)$
- $t_{\text{μέση καθυσ δικτύου}} = 3 \text{ seconds}$

Γνωρίζουμε ότι ο χρόνος που απαιτείται για να μεταδοθεί ένα αντικείμενο μεγέθους  $L$  μέσα από ένα κανάλι με ρυθμό  $R$  ισούται με:

$$t = \frac{L}{R}$$

# Exercise 3 - Solution

$$\text{Άρα } \Delta = \frac{850000 \text{ bits}}{15000000 \text{ bits/sec}} = 0.0567 \text{ sec}$$

Το traffic intensity στο link ισούται με

$$\beta \times \Delta = (16 \text{ requests/sec}) (0.0567 \text{ sec/request}) = 0.907$$

Συνεπώς με αντικατάσταση στην (2) προκύπτει ότι

$$t_{\text{μέση καθυσ προσπ}} = \frac{\Delta}{1 - \Delta \times \beta} = 0.6 \text{ sec}$$

$$\text{Άρα από την (1) έχουμε ότι τελικά } t_{\text{Total}} = 0.6 \text{ sec} + 3 \text{ sec} = 3.6 \text{ sec}$$



# Exercise 3 - Solution

b) Τώρα υποθέστε ότι μια cache έχει τοποθετηθεί στο LAN του ιδρύματος. Υποθέστε ότι ο ρυθμός αποτυχίας είναι 0.4. Βρείτε τον συνολικό χρόνο απόκρισης.

Solution: Λόγω της χρήσης της cache το traffic intensity στο link προσπέλασης μειώνεται κατά 60% διότι το 60% των requests ικανοποιούνται μέσα στο δίκτυο του ιδρύματος. Άρα

$$t_{\text{μέση καθυσ προσπ}} = \frac{\Delta}{1 - (0.4) \times \Delta \times \beta} = 0.089 \text{ sec}$$

- Σε περίπτωση που το request ικανοποιηθεί από την cache, τότε ο χρόνος απόκρισης είναι 0. Το ενδεχόμενο αυτό συγκεντρώνει πιθανότητα 0.6.
- Σε περίπτωση που το request δεν ικανοποιηθεί από την cache, τότε ο χρόνος απόκρισης θα είναι

$$0.089 \text{ sec} + 3 \text{ sec} = 3.089 \text{ sec}$$

με πιθανότητα 0.4.

- Άρα συνδυάζοντας τα 2 ενδεχόμενα προκύπτει ότι ο μέσος χρόνος απόκρισης είναι

$$(0.6) \times (0 \text{ sec}) + (0.4) \times (3.089 \text{ sec}) = 1.24 \text{ sec} < \text{3.6 sec}$$

# Exercise 4

Έστω το παρακάτω Wireshark trace κατά την διάρκεια που ο browser σας στέλνει ένα HTTP GET μήνυμα. Απαντήστε τα παρακάτω ερωτήματα:

**α) Ποιο είναι το URL του αρχείου που ζητήθηκε από τον browser?**

Sol: Το request για το αρχείο είναι <http://gaia.cs.umass.edu/cs453/index.html>. Το πεδίο Host: υποδεικνύει το όνομα του server και η κατάληξη [/cs453/index.html](http://gaia.cs.umass.edu/cs453/index.html) υποδεικνύει το όνομα του αρχείου.

```
GET /cs453/index.html HTTP/1.1<cr><lf>Host: gai
a.cs.umass.edu<cr><lf>User-Agent: Mozilla/5.0 (
Windows;U; Windows NT 5.1; en-US; rv:1.7.2) Gec
ko/20040804 Netscape/7.2 (ax) <cr><lf>Accept:ex
t/xml, application/xml, application/xhtml+xml, text
/html;q=0.9, text/plain;q=0.8,image/png,*/*;q=0.5
<cr><lf>Accept-Language: en-us,en;q=0.5<cr><lf>Accept-
Encoding: zip,deflate<cr><lf>Accept-Charset: ISO
-8859-1,utf-8;q=0.7,*;q=0.7<cr><lf>Keep-Alive: 300<cr>
<lf>Connection:keep-alive<cr><lf><cr><lf>
```

# Exercise 4

**b) Ποια έκδοση του HTTP τρέχει ο browser?**

Sol: Ο browser τρέχει την έκδοση HTTP 1.1.

**c) Ο browser ζητάει non-persistent ή persistent σύνδεση?**

Sol: Ο browser ζητάει persistent σύνδεση όπως φαίνεται από το πεδίο Connection: keep-alive.

**d) Ποια είναι η IP διεύθυνση του host στον οποίο τρέχει ο browser?**

Sol: Η απάντηση μπορεί να δοθεί μόνο από τα IP datagrams μελετώντας τα TCP segments του HTTP GET request.

```
GET /cs453/index.html HTTP/1.1<cr><lf>Host: gai
a.cs.umass.edu<cr><lf>User-Agent: Mozilla/5.0 (
Windows;U; Windows NT 5.1; en-US; rv:1.7.2) Gec
ko/20040804 Netscape/7.2 (ax) <cr><lf>Accept:ex
t/xml, application/xml, application/xhtml+xml, text
/html;q=0.9, text/plain;q=0.8,image/png,*/*;q=0.5
<cr><lf>Accept-Language: en-us,en;q=0.5<cr><lf>Accept-
Encoding: zip,deflate<cr><lf>Accept-Charset: ISO
-8859-1,utf-8;q=0.7,*;q=0.7<cr><lf>Keep-Alive: 300<cr>
<lf>Connection:keep-alive<cr><lf><cr><lf>
```

# Exercise 4

**ε) Ποιος τύπος browser στέλνει το παραπάνω μήνυμα? Και γιατί ο τύπος browser χρησιμοποιείται σε ένα μήνυμα HTTP request?**

**Sol:** Ο browser είναι Mozilla/5.0. Η πληροφορία αυτή χρησιμοποιείται από τον server για να στέλνει διαφορετικές εκδόσεις του ίδιου αντικειμένου σε διαφορετικούς τύπους browsers.

```
GET /cs453/index.html HTTP/1.1<cr><lf>Host: gai
a.cs.umass.edu<cr><lf>User-Agent: Mozilla/5.0 (
Windows;U; Windows NT 5.1; en-US; rv:1.7.2) Gec
ko/20040804 Netscape/7.2 (ax) <cr><lf>Accept:ex
t/xml, application/xml, application/xhtml+xml, text
/html;q=0.9, text/plain;q=0.8,image/png,*/*;q=0.5
<cr><lf>Accept-Language: en-us,en;q=0.5<cr><lf>Accept-
Encoding: zip,deflate<cr><lf>Accept-Charset: ISO
-8859-1,utf-8;q=0.7,*;q=0.7<cr><lf>Keep-Alive: 300<cr>
<lf>Connection:keep-alive<cr><lf><cr><lf>
```



# Exercise 5

Το παρακάτω Wireshark trace περιέχει την απάντηση του server στο HTTP GET μήνυμα της προηγούμενης άσκησης. Απαντήστε τα παρακάτω ερωτήματα:

**a) Ο server μπόρεσε επιτυχώς να βρει το αρχείο ή όχι? Πότε εστάλη η απάντηση?**

Sol: Από το code status 200 και την λέξη OK συμπεραίνουμε ότι ο server βρήκε επιτυχώς το αρχείο. Η ημερομηνία αποστολής της απάντησης είναι Tuesday, 07 Mar 2008 12:39:45 Greenwich Mean Time (Date: field).

```
HTTP/1.1 200 OK<cr><lf>Date: Tue, 07 Mar 2008
12:39:45GMT<cr><lf>Server: Apache/2.0.52 (Fedora)
<cr><lf>Last-Modified: Sat, 10 Dec2005 18:27:46
GMT<cr><lf>ETag: "526c3-f22-a88a4c80"<cr><lf>Accept-
Ranges: bytes<cr><lf>Content-Length: 3874<cr><lf>
Keep-Alive: timeout=max=100<cr><lf>Connection:
Keep-Alive<cr><lf>Content-Type: text/html; charset=
ISO-8859-1<cr><lf><cr><lf><!doctype html public "-
//w3c//dtd html 4.0 transitional//en"><lf><html><lf>
<head><lf> <meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1"><lf> <meta
name="GENERATOR" content="Mozilla/4.79 [en] (Windows NT
5.0; U) Netscape]"><lf> <title>CMPSCI 453 / 591 /
NTU-ST550A Spring 2005 homepage</title><lf></head><lf>
<much more document text following here (not shown)>
```

# Exercise 5

**b) Πότε το αρχείο τροποποιήθηκε για τελευταία φορά?**

Sol: Η τελευταία τροποποίηση του αρχείου ήταν Saturday 10 Dec 2005 18:27:46 GMT (Last Modified: field).

**c) Πόσα bytes περιέχει το επιστρεφόμενο αρχείο?**

Sol: Το αρχείο περιέχει 3874 bytes (Content-Length field).

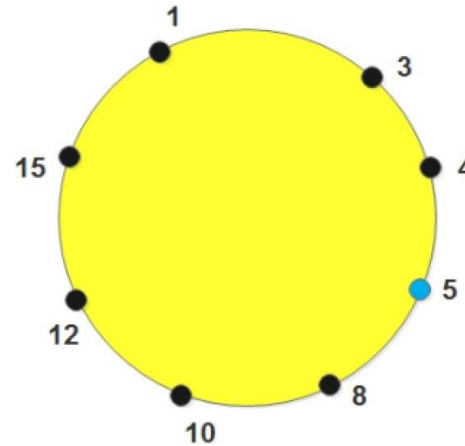
**d) Ο server συμφώνησε σε persistent σύνδεση?**

Sol: Ο server συμφώνησε για persistent σύνδεση όπως φαίνεται από το πεδίο Connection: Keep-Alive.

```
HTTP/1.1 200 OK<cr><lf>Date: Tue, 07 Mar 2008
12:39:45GMT<cr><lf>Server: Apache/2.0.52 (Fedora)
<cr><lf>Last-Modified: Sat, 10 Dec2005 18:27:46
GMT<cr><lf>ETag: "526c3-f22-a88a4c80"<cr><lf>Accept-
Ranges: bytes<cr><lf>Content-Length: 3874<cr><lf>
Keep-Alive: timeout=max=100<cr><lf>Connection:
Keep-Alive<cr><lf>Content-Type: text/html; charset=
ISO-8859-1<cr><lf><cr><lf><!doctype html public "-
//w3c//dtd html 4.0 transitional//en"><lf><html><lf>
<head><lf> <meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1"><lf> <meta
name="GENERATOR" content="Mozilla/4.79 [en] (Windows NT
5.0; U) Netscape]><lf> <title>CMPSCI 453 / 591 /
NTU-ST550A Spring 2005 homepage</title><lf></head><lf>
<much more document text following here (not shown)>
```

# Exercise 6

- Υποθέστε πως έχετε την παρακάτω διάταξη peers. Κάθε peer γνωρίζει τους 2 επόμενους peers (peer churn). Έστω ότι ο κόμβος 5 αποχωρεί. Ποια διαδικασία θα ακολουθηθεί για να ανανεώσει ο κόμβος 3 τις πληροφορίες για τους successors του;



What happens when node 5 leaves?

# Exercise 6 - Solution

- Όταν ο peer 3 πληροφορηθεί πως ο peer 5 αποχώρησε:
  - Ο peer 3 ρωτάει τον peer 4 (πρώτος successor) για την ταυτότητα του επόμενού του (peer 8).
  - Έπειτα, ο peer 8 θα γίνει ο 2ος successor του 3, αφού πρώτα ο peer 4 κάνει update μετατρέποντας τον peer 8 σε πρώτο successor δικό του.