

## Εργαστηριακή Άσκηση 11

### Πλοήγηση στον παγκόσμιο ιστό

Ο παγκόσμιος ιστός (World Wide Web) είναι μια αχανής συλλογή εγγράφων που εμφανίζεται με τη μορφή ιστοσελίδων. Η ιστοσελίδα είναι μια συλλογή αντικειμένων, όπως αρχείο HTML, εικόνες JPEG, αρχεία ήχου, βίντεο, κώδικας Javascript, Java applet, κλπ, και μπορεί να περιέχει παραπομπές προς άλλες ιστοσελίδες<sup>1</sup>. Η ανάγνωση των ιστοσελίδων μπορεί να γίνει με τη βοήθεια ενός προγράμματος-πελάτη (client) γνωστού ως πλοηγού ιστού ή φυλλομετρητή (web browser). Οι χρήστες μπορούν να ακολουθήσουν τις υπερζεύξεις με ένα κλικ του ποντικιού, διαδικασία γνωστή, ως πλοήγηση (navigation). Ο πλοηγός ιστού προσκομίζει την ιστοσελίδα και την απεικονίζει στην οθόνη του υπολογιστή, μεταφράζοντας το κείμενο και τις εντολές μορφοποίησης. Οι υπερζεύξεις απεικονίζονται έντονα, είτε με υπογράμμιση είτε με τη χρήση ενός συγκεκριμένου χρώματος, είτε και με τα δύο.

Σε κάθε ιστοσελίδα έχει προσδεθεί ένα μονοσήμαντο όνομα, ώστε να μην υπάρχει καμία αμφιβολία όσον αφορά την αναγνώρισή της. Συγκεκριμένα, σε κάθε σελίδα εκχωρείται μια ταυτότητα, ένας *Ομοιόμορφος Εντοπιστής Πόρων URL (Uniform Resource Locator)* που στην ουσία παίζει τον ρόλο του παγκόσμιου ονόματος της σελίδας. Τα URL αποτελούνται από τρία μέρη: το πρωτόκολλο, το όνομα DNS ή τη διεύθυνση IP της μηχανής στην οποία βρίσκεται η σελίδα και ένα τοπικό όνομα που προσδιορίζει μονοσήμαντα τη συγκεκριμένη σελίδα (συνήθως ένα όνομα καταλόγου ή αρχείου). Για παράδειγμα, το URL <http://edu-dy.cn.ntua.gr/index.html> αποτελείται από τρία μέρη: το πρωτόκολλο (*http*), το όνομα DNS της μηχανής στην οποία φιλοξενείται η σελίδα (*edu-dy.cn.ntua.gr*) και το όνομα του αρχείου που περιέχει τη σελίδα (*index.html*), με κάποια σημεία στίξης που διαχωρίζουν τα κομμάτια. Για να επισκεφθεί, ο χρήστης μια ιστοσελίδα αρκεί να συμπληρώσει σωστά τη διεύθυνση της ιστοσελίδας στη γραμμή διευθύνσεων του πλοηγού ιστού.

Για τη μεταφορά των πληροφοριών που περιέχει η ιστοσελίδα χρησιμοποιείται το πρωτόκολλο στρώματος εφαρμογής [HTTP](#) (hypertext transfer protocol). Η πρώτη έκδοση του πρωτοκόλλου ήταν η HTTP/1.0 που περιγράφεται στο [RFC 1945](#). Η επόμενη και πλέον διαδεδομένη έκδοση HTTP/1.1 ορίστηκε το 1997 στο [RFC 2068](#), που στη συνέχεια αντικαταστάθηκε από το [RFC 2616](#) του 1999. Σε συντομία, η βασική λειτουργία του HTTP είναι η μεταφορά πόρων (resources). Πόρος είναι ένα κομμάτι πληροφορίας που προσδιορίζεται μοναδικά από ένα URL (το R στο ακρωνύμιο URL). Η πιο κοινή περίπτωση πόρου είναι το αρχείο. Όμως, ένας πόρος μπορεί να δημιουργείται δυναμικά ως αποτέλεσμα εντολών (π.χ. από κώδικα PHP), είτε να είναι έγγραφο διαθέσιμο σε πολλές γλώσσες, είτε κώδικας Javascript που θα εκτελεσθεί δυναμικά ή οτιδήποτε άλλο.

Σε κάθε ιστοθέση (Web site) που χρησιμοποιεί το πρωτόκολλο HTTP υπάρχει μία διεργασία εξυπηρετητή (web server), η οποία παρακολουθεί την TCP θύρα (port) 80 περιμένοντας εισερχόμενες συνδέσεις από πελάτες πλοηγούς (web clients). Μόλις εδραιωθεί η σύνδεση, ο πελάτης στέλνει μία αίτηση HTTP (HTTP request) και ο εξυπηρετητής στέλνει μία απόκριση HTTP (HTTP response) που περιέχει διάφορα αντικείμενα. Μετά η σύνδεση απελευθερώνεται, δηλαδή, το HTTP δεν διατηρεί πληροφορία κατάστασης (stateless) μεταξύ διαδοχικών δοσοληψιών (transactions).

Για την πρόσβαση σε εξυπηρετητές ιστού έχει όμως πλέον επικρατήσει<sup>2</sup> η χρήση του πρωτοκόλλου [HTTPS](#) (HTTP Secure). Ο εξυπηρετητής τυπικά ακούει στη θύρα 443 και η επικοινωνία με αυτόν είναι κρυπτογραφημένη ακολουθώντας το πρωτόκολλο TLS ή παλιότερα το SSL (που θα δείτε στην επόμενη εργαστηριακή άσκηση). Τα κύρια κίνητρα για το HTTPS ήταν η πιστοποίηση ταυτότητας της ιστοθέσης καθώς και η προστασία των δεδομένων που ανταλλάσσονται με αυτήν.

<sup>1</sup> Κείμενο που παραπέμπει σε άλλο κείμενο αναφέρεται ως υπερκείμενο (hypertext) και οι παραπομπές είναι γνωστές ως υπερζεύξεις (hypertext links).

<sup>2</sup> Υποστηρίζεται από το 84,7% των ιστοθέσεων (δείτε περισσότερα <https://w3techs.com/technologies/comparison/ce-http2,ce-httpsdefault>)

Οι συνδέσεις HTTP διακρίνονται σε μη επίμονες (Non-persistent HTTP) και σε επίμονες (Persistent HTTP). Το πρωτόκολλο HTTP/1.0 χρησιμοποιεί μη επίμονες συνδέσεις HTTP. Σε αυτές, πάνω από μία σύνδεση TCP μπορεί να αποσταλεί το πολύ ένα αντικείμενο κάθε φορά. Το πρωτόκολλο HTTP/1.1 χρησιμοποιεί επίμονες συνδέσεις HTTP. Σε αυτές, πάνω από την ίδια σύνδεση TCP μπορούν να ζητηθούν και αποσταλούν πολλαπλά αντικείμενα μεταξύ πελάτη-εξυπηρετητή. Έτσι μειώνονται οι επιπτώσεις της καθυστέρησης κατά την τριπλή χειραγία TCP (μία για κάθε αντικείμενο στο HTTP/1.0). Ως εκ τούτου, η μεταφορά των αντικειμένων και η εμφάνιση της ιστοσελίδας γίνεται γρηγορότερα. Με την πάροδο των χρόνων όμως, το μέγεθος των ιστοσελίδων έχει μεγαλώσει, τυπικά ξεπερνούν τα 2 Mb, και ο αριθμός των αντικειμένων που περιέχει η αρχική σελίδα μιας ιστοθέσης άνετα ξεπερνά τα 100. Το HTTP/1.1 δεν αποδίδει καλά και για αυτό αναπτύχθηκαν διάφορα πολύπλοκα τεχνάσματα.

Προκειμένου να παρακαμφτούν αυτά, το 2007 ξεκίνησαν οι εργασίες της ομάδας HTTPbis Working Group που κατέληξαν το 2014 στην αναθεώρηση του HTTP/1.1 σε μια σειρά έξι νέων προτύπων RFC 7230 έως RFC 7235 και το 2015 στη δημοσίευση της δεύτερης έκδοσής του, το HTTP/2, στο πρότυπο [RFC 7540](#). Το [HTTP/2](#) διατηρεί τη συμβατότητα με το HTTP/1.1 χρησιμοποιώντας τις ίδιες μεθόδους (δείτε παρακάτω) με αυτό και προσθέτει νέα λειτουργικότητα ώστε το φόρτωμα των ιστοσελίδων να γίνεται γρηγορότερα. Στο HTTP/2 χρησιμοποιούνται λιγότερες συγκριτικά με το HTTP/1.1 συνδέσεις και η πληροφορία μεταδίδεται σε δυαδική μορφή (όχι ως κείμενο όπως στο HTTP/1.1) σε παράλληλες ροές εντός μιας σύνδεσης TCP, με αποτέλεσμα το ταχύτερο κατέβασμα των ιστοσελίδων. Πλέον όλοι οι πλοηγοί ιστού έχουν ενσωματώσει το HTTP/2 και σύμφωνα με τα πιο πρόσφατα στοιχεία υποστηρίζεται από το 35,8% των ιστοθέσεων (δείτε περισσότερα στην ιστοθέση <https://w3techs.com/technologies/details/ce-http2>).

Το 2018 ξεκίνησαν επίσημα οι εργασίες για την προτυποποίηση της επόμενης έκδοσής του, το [HTTP/3](#). Στο HTTP/3, για τις παράλληλες συνδέσεις χρησιμοποιείται ένα άλλο πρωτόκολλο μεταφοράς, το [QUIC](#), που βασίζεται στο UDP και περιγράφεται στο [RFC 9000](#). Σκοπός του QUIC είναι να ελαχιστοποιήσει το εγγενές στο HTTP/2 φαινόμενο καθυστερήσεων στην κορυφή της ουράς (head-of-line blocking) λόγω της πολυπλεξίας παράλληλων ροών σε μια σύνδεση TCP ώστε οι ιστοσελίδες να φορτώνονται πολύ γρηγορότερα (πάνω από 3 φορές σε σχέση με το HTTP/1.1 σε μερικές περιπτώσεις). Παρότι ήταν ακόμη προσχέδιο, το HTTP/3 υλοποιήθηκε νωρίς (2020) στους δημοφιλείς πλοηγούς ιστού και πλέον υποστηρίζεται από το 27,5% των ιστοθέσεων (δείτε σύγκριση με HTTPS/2 στην ιστοθέση <https://w3techs.com/technologies/comparison/ce-http2,ce-http3>).

Τέλος, πρόσφατα Ιούνιος 2022, δημοσιεύθηκε η σειρά προτύπων RFC 9110 έως 9114 που αντικαθιστά όλα τα προηγούμενα. Αποτελεί την τελική έκδοση για τα HTTP/1.1 ([RFC 9112](#)), HTTP/2 ([RFC 9113](#)) και HTTP/3 ([RFC 9114](#)). Ξεχωρίζει τους σημασιολογικούς ορισμούς HTTP ([RFC 9110](#)) από την προσωρινή αποθήκευση (caching) απαντήσεων ([RFC 9111](#)) επιτρέποντας έτσι την περαιτέρω ανεξάρτητη εξέλιξη των εκδόσεων του πρωτοκόλλου.

Σκοπός αυτής της εργαστηριακής άσκησης είναι η εξέταση του πρωτοκόλλου εφαρμογής HTTP που χρησιμοποιείται ευρύτατα στο διαδίκτυο για την πλοήγηση στον ιστό, με τη βοήθεια του αναλυτή πρωτοκόλλων Wireshark. Εδώ θα χρησιμοποιήσετε τη λειτουργία *Capture* με φίλτρο, ώστε τα πλαίσια που καταγράφονται να περιέχουν κάποια συγκεκριμένα χαρακτηριστικά. Υπενθυμίζεται ότι το φίλτρο απεικόνισης (*Display*), που επιλέγετε από το μενού *Analyze*, μπορεί να (απ)ενεργοποιηθεί οποιαδήποτε στιγμή κατά τη διάρκεια της καταγραφής, καθώς επίσης και μετά την ολοκλήρωση αυτής, προκειμένου να αποκρύπτει (αποκαλύπτει) κάποια από τα συλληφθέντα πλαίσια, ενώ το φίλτρο σύλληψης, που επιλέγετε από το μενού *Capture*, ενεργοποιείται πάντοτε πριν ξεκινήσει η διαδικασία καταγραφής, με αποτέλεσμα να καταγράφεται μόνο ένα μέρος των διερχόμενων πλαισίων. Επιπλέον, ας σημειωθεί ότι η απενεργοποίηση του φίλτρου απεικόνισης γίνεται πιέζοντας το κουμπί *Clear* (η διαγραφή του φίλτρου στο πεδίο εισαγωγής δεν το ακυρώνει!).

**Για την άσκηση μπορείτε να χρησιμοποιήσετε τον προσωπικό σας υπολογιστή ή έναν του PC Lab. Απαντήστε στο συνοδευτικό φυλλάδιο, το οποίο θα υποβάλλετε ως αρχείο pdf.**

## 1. Ανάκτηση HTML σελίδας

Όπως έχει ήδη αναφερθεί, το HTTP είναι πρωτόκολλο επιπέδου εφαρμογής και ακολουθεί το μοντέλο πελάτη/εξυπηρετητή (client/server model). Το HTTP ορίζει ένα σύνολο μεθόδων αίτησης HTTP (HTTP request methods) ως λέξεις της Αγγλικής γλώσσας (συνήθως αναφέρονται ως HTTP verbs) που δηλώνουν την επιθυμητή δράση σε ένα συγκεκριμένο πόρο. Στο HTTP/1.0 ορίστηκαν οι μέθοδοι GET, HEAD και POST. Εξ αυτών, η μέθοδος GET χρησιμοποιείται μόνο για τη λήψη δεδομένων και ζητά την αναπαράσταση ενός συγκεκριμένου πόρου. Η μέθοδος HEAD είναι παρόμοια με την GET αλλά χωρίς την αναπαράσταση του πόρου, π.χ. για να διαπιστωθεί η διαθεσιμότητά του ή να βρεθεί το μέγεθος ενός αρχείου. Η μέθοδος POST χρησιμοποιείται για την αποστολή δεδομένων προς τον καθοριζόμενο πόρο (π.χ. ένα συνθηματικό) προκαλώντας συνήθως κάποια αλλαγή στην πλευρά του εξυπηρετητή. Στο HTTP/1.1 ορίστηκαν επιπλέον μέθοδοι οι: PUT, DELETE, CONNECT, OPTIONS, TRACE και PATCH. Για περισσότερες λεπτομέρειες δείτε [https://en.wikipedia.org/wiki/HTTP#Request\\_methods](https://en.wikipedia.org/wiki/HTTP#Request_methods). Στις μεθόδους αίτησης ο εξυπηρετητής απαντά με αποκρίσεις HTTP (HTTP responses). Οι αποκρίσεις είναι κωδικοί κατάστασης (status codes) που δείχνουν το κατά πόσο μια συγκεκριμένη αίτηση HTTP ικανοποιήθηκε. Ομαδοποιούνται σε πέντε κατηγορίες: informational, successful, redirects, client errors και servers errors. Για παράδειγμα, το 200 OK δηλώνει επιτυχία και το 404 Not Found δείχνει ότι ο εξυπηρετητής δεν βρήκε τον αιτούμενο πόρο. Για την πλήρη λίστα των κωδικών κατάστασης δείτε [https://en.wikipedia.org/wiki/List\\_of\\_HTTP\\_status\\_codes](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes). Οι μορφές των μηνυμάτων αίτησης HTTP και απόκρισης HTTP είναι ταυτόσημες. Και στις δύο περιπτώσεις τα μηνύματα περιλαμβάνουν:

- Μια αρχική γραμμή,
- Καμία ή πολλές γραμμές επικεφαλίδων,
- Μια κενή γραμμή (blank line)<sup>3</sup>, και
- Προαιρετικά το κυρίως σώμα του μηνύματος HTTP (π.χ. ένα αρχείο, τα δεδομένα μιας ερώτησης, κλπ).

Η αρχική γραμμή είναι διαφορετική για τις αιτήσεις και αποκρίσεις. Στις αιτήσεις η αρχική γραμμή των περιέχει τρία μέρη που χωρίζονται με κενά: το όνομα της μεθόδου (πάντα με κεφαλαία γράμματα, π.χ. GET), την τοπική διαδρομή (local path) του αιτούμενου πόρου και τέλος την έκδοση του πρωτοκόλλου HTTP που χρησιμοποιείται. Η αρχική γραμμή των αποκρίσεων αποκαλείται γραμμή κατάστασης. Περιέχει και αυτή τρία μέρη που χωρίζονται με κενά: την έκδοση του HTTP, ένα κωδικό κατάστασης που υποδεικνύει το αποτέλεσμα της αίτησης (π.χ. 200) και μια Αγγλική λέξη που περιγράφει τον κωδικό κατάστασης (π.χ. OK). Οι γραμμές επικεφαλίδων παρέχουν πληροφορίες για την αίτηση ή την απόκριση ή για το αντικείμενο που περιέχει το κυρίως σώμα. Εμφανίζονται όπως το σύνθημα κείμενο: μια επικεφαλίδα ανά γραμμή της μορφής "Header-Name: value" και καταλήγουν σε <CR><LF>. Το κυρίως σώμα του μηνύματος HTTP ακολουθεί τις επικεφαλίδες. Στις αποκρίσεις, το κυρίως σώμα επιστρέφει τον αιτούμενο πόρο στον πελάτη ή κάποιο επεξηγηματικό κείμενο σε περίπτωση λάθους. Στις αιτήσεις, το κυρίως σώμα είναι το μέρος όπου αρχεία (ή δεδομένα που εισάγει ο χρήστης) αποστέλλονται από τον (ή προς τον) εξυπηρετητή. Όταν το μήνυμα HTTP περιέχει σώμα, συνήθως, οι επικεφαλίδες περιγράφουν το σώμα (τον τύπο και το μήκος του).

Σε αυτή την άσκηση θα καταγραφούν τα μηνύματα HTTP που παράγονται κατά την επίσκεψη μιας ιστοσελίδας. Για μια λεπτομερή περιγραφή της ροής μηνυμάτων όλων των εμπλεκόμενων πρωτοκόλλων κατά το κατέβασμα μιας ιστοσελίδας δείτε το παράδειγμα στην ιστοθέση [https://www.eventhelix.com/Networking/http\\_sequence\\_diagram.pdf](https://www.eventhelix.com/Networking/http_sequence_diagram.pdf).

Προτού αρχίσετε την καταγραφή θα χρειαστεί να αδειάσετε την προσωρινή μνήμη (cache) του πλοηγού σας. Ξεκινήστε τον πλοηγό ιστού και πιέστε τα πλήκτρα Shift+Ctrl+Del ή επιλέξτε στο μενού τη σχετική εντολή, π.χ. στον Firefox *History* → *Clear Recent History*. Στο παράθυρο που θα εμφανισθεί προσδιορίστε το είδος (cache) της προς διαγραφή πληροφορίας, δηλώστε ως διάστημα

<sup>3</sup> Το τέλος των γραμμών δηλώνεται με τους χαρακτήρες <CR><LF>.

τουλάχιστον μία ημέρα, επιβεβαιώστε την πρόθεσή σας, περιμένετε να ολοκληρωθεί η διαγραφή και κλείστε τα παράθυρα διαλόγου. Στη συνέχεια ξεκινήστε μια καταγραφή, επισκεφτείτε την ιστοσελίδα <http://edu-dy.cn.ntua.gr/lab11.html> και σταματήστε την καταγραφή όταν φορτωθεί πλήρως η σελίδα. Εφαρμόστε κατάλληλο φίλτρο απεικόνισης ώστε να παραμείνουν μόνο μηνύματα του πρωτοκόλλου HTTP.

- 1.1 Ποιο είναι το φίλτρο απεικόνισης που εφαρμόσατε;
- 1.2 Ποια είναι η έκδοση του πρωτοκόλλου HTTP που χρησιμοποιεί ο πλοηγός ιστού σας;
- 1.3 Ποια είναι η έκδοση του πρωτοκόλλου HTTP που χρησιμοποιεί ο εξυπηρετητής ιστού;
- 1.4 Ποια είναι η ονομασία της μεθόδου στο πρώτο αίτημα HTTP από τον υπολογιστή σας προς τον εξυπηρετητή ιστού; [Υπόδειξη: Για τα ονόματα των διαθέσιμων μεθόδων και επικεφαλίδων HTTP συμβουλευθείτε την ιστοσελίδα <http://www.networksorcery.com/enp/protocol/http.htm> που θα βρείτε στο [Internet Archive](#)].
- 1.5 Ποιος είναι ο κωδικός κατάστασης και ποια η επεξηγηματική λέξη που επιστρέφει ο εξυπηρετητής ιστού στην πρώτη απόκριση HTTP;
- 1.6 Ένα μηχανήμα που λειτουργεί ως εξυπηρετητής μπορεί να φιλοξενεί ιστοθέσεις με διαφορετικά ονόματα. Ο πλοηγός στα αιτήματά του πρέπει να προσδιορίζει τη συγκεκριμένη από την οποία ζητά κάποιο πόρο. Ποιο είναι το όνομα της σχετικής επικεφαλίδας HTTP;
- 1.7 Ο πλοηγός στα αιτήματά του πληροφορεί τον εξυπηρετητή για το είδος του. Ποιο είναι το όνομα της σχετικής επικεφαλίδας HTTP και ποια η έκδοση λογισμικού του πλοηγού σας;
- 1.8 Αντίστοιχα ο εξυπηρετητής ιστού πληροφορεί τον πλοηγό για το είδος του. Ποιο είναι το όνομα της σχετικής επικεφαλίδας HTTP και ποιο είδος εξυπηρετητή ιστού φιλοξενεί την ιστοθέση <http://edu-dy.cn.ntua.gr>;
- 1.9 Εφαρμόστε νέο φίλτρο απεικόνισης ώστε να παραμείνουν μόνο τα αιτήματα HTTP (HTTP requests) προς τον εξυπηρετητή ιστού για τη μέθοδο της ερώτησης 1.4. Ποια είναι η σύνταξή του;
- 1.10 Πόσα αιτήματα HTTP απέστειλε ο υπολογιστής σας προς τον εξυπηρετητή ιστού;
- 1.11 Ανά είδος αιτούμενου πόρου ο πλοηγός δηλώνει τους προτιμώμενους τύπους περιεχομένου. Ποιο είναι το όνομα της σχετικής επικεφαλίδας HTTP; Σημειώστε μία από τις προτιμήσεις για κάθε διαφορετικό είδος πόρου που ζητήθηκε.
- 1.12 Αντίστοιχα δηλώνει τις προτιμώμενες κωδικοποιήσεις και γλώσσα για το περιεχόμενο. Ποιο είναι το όνομα των σχετικών επικεφαλίδων HTTP;
- 1.13 Ποιες γλώσσες εκτός της Αγγλικής δηλώνει ενδεχομένως ο πλοηγός ιστού σας ότι μπορεί να δεχτεί από τον εξυπηρετητή;

Οι μοντέρνοι πλοηγοί χρησιμοποιούν μέχρι έξι (6) συνδέσεις TCP με τον εκάστοτε εξυπηρετητή. Ο εξυπηρετητής κλείνει αυτές τις συνδέσεις αφού στείλει τα δεδομένα. Ο πλοηγός μπορεί να ζητήσει επίμονες συνδέσεις TCP δηλώνοντας ότι αυτές πρέπει να παραμένουν ζωντανές και να μην κλείνουν αμέσως μετά την ολοκλήρωση της μετάδοσης. Έτσι μπορεί να ζητήσει περισσότερους από ένα πόρους (π.χ. μια σελίδα και ένα αρχείο) πάνω από κάθε σύνδεση.

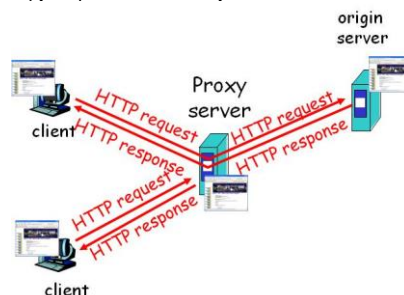
- 1.14 Ποιο είναι το όνομα της επικεφαλίδας HTTP με την οποία ζητούνται επίμονες συνδέσεις και ποια η σχετική τιμή της;
- 1.15 Εφαρμόστε κατάλληλο φίλτρο ώστε να παραμείνουν μόνο τα πρώτα τεμάχια TCP των τριμερών χειραψιών που διεξήχθησαν με τον εξυπηρετητή [edu-dy.cn.ntua.gr](http://edu-dy.cn.ntua.gr). Πόσες συνδέσεις TCP παρατηρείτε στην καταγραφή σας και ποιες οι αντίστοιχες θύρες πηγής;
- 1.16 Εφαρμόστε νέο φίλτρο ώστε να παραμείνουν μόνο τα αιτήματα HTTP (HTTP requests). Ποια είναι η σύνταξή του;
- 1.17 Ζητήθηκε από τον πλοηγό ιστού σας να κατέβουν *παράλληλα* κάποιοι από τους αιτούμενους πόρους; Αιτιολογήστε την απάντησή σας παρατηρώντας το κατά πόσον τα αιτήματα γίνονται με χρήση διαφορετικών συνδέσεων TCP.
- 1.18 Ζητήθηκε από τον πλοηγό ιστού σας να κατέβουν *ακολουθιακά* κάποιοι από τους αιτούμενους πόρους; Αιτιολογήστε την απάντησή σας παρατηρώντας το κατά πόσο χρησιμοποιείται η ίδια σύνδεση TCP για διαφορετικά αιτήματα.



- 1.19 Εφαρμόστε νέο φίλτρο ώστε να παραμείνουν μόνο οι αποκρίσεις HTTP (HTTP responses). Ποια είναι η σύνταξή του;
- 1.20 Για μια επίμονη σύνδεση TCP ο εξυπηρετητής μπορεί να δηλώσει τον χρόνο διακοπής της (timeout) εάν παραμένει αδρανής καθώς και το μέγιστο πλήθος αιτημάτων που μπορούν να γίνουν σε αυτήν. Ποιο είναι το όνομα της σχετικής επικεφαλίδας HTTP;
- 1.21 Στην περίπτωση σύνδεσης όπου κατέβηκαν ακολουθιακά κάποιοι πόροι, τι παρατηρείτε στις αποκρίσεις του εξυπηρετητή όσον αφορά τον χρόνο διακοπής και το μέγιστο πλήθος των αιτημάτων;
- 1.22 Στις αποκρίσεις του ο εξυπηρετητής δηλώνει την ημέρα και ώρα της απάντησης καθώς και την ημέρα και ώρα που τροποποιήθηκε για τελευταία φορά ο σχετικός πόρος σύμφωνα με το δικό του ρολόι. Ποιο είναι το όνομα των σχετικών επικεφαλίδων HTTP;
- 1.23 Επιπλέον ο εξυπηρετητής στις αποκρίσεις του δηλώνει για πόσο χρονικό διάστημα μπορεί να αποθηκευτεί σε προσωρινή μνήμη (cache) η απόκριση. Ποιο είναι το όνομα της σχετικής επικεφαλίδας HTTP και ποια η μέγιστη διάρκεια προσωρινής αποθήκευσης που ορίζεται;
- 1.24 Τόσο στα αιτήματα όσο και στις αποκρίσεις μπορεί να δηλώνεται το είδος του περιεχομένου στο κυρίως σώμα του μηνύματος HTTP. Παρόμοια μπορεί να δηλώνεται το μήκος του περιεχομένου, όπου η τιμή μηδέν σημαίνει ότι δεν υπάρχει σώμα. Ποιο είναι το όνομα των σχετικών επικεφαλίδων HTTP;
- 1.25 Τι μήκος έχει το σώμα της πρώτης απόκρισης που έστειλε ο εξυπηρετητής και τι είδους περιεχόμενο μεταφέρει; Αντίστοιχα για τη δεύτερη απόκριση.
- 1.26 Το περιεχόμενο του σώματος της πρώτης απόκρισης είναι η περιγραφή της ιστοσελίδας σε γλώσσα HTML (HyperText Markup Language) εντός του οποίου υπάρχει και ο τίτλος της ιστοσελίδας. Ποιος είναι αυτός και σε ποιο σημείο εμφανίζεται στον πλοηγό σας;
- 1.27 Το περιεχόμενο του σώματος της δεύτερης απόκρισης είναι η περιγραφή για τη μορφοποίηση της ιστοσελίδας σε γλώσσα CSS (Cascading Style Sheets). Ποιο είναι το μήκος του σε byte και ποιο σε γραμμές;
- 1.28 Εφαρμόστε πάλι φίλτρο απεικόνισης ώστε να παραμείνουν μόνο τα αιτήματα HTTP. Εκτός από την ίδια την ιστοσελίδα (κείμενο σε γλώσσα html) και την περιγραφή αναπαράστασής του (κείμενο σε γλώσσα CCS), ο πλοηγός ιστού ζήτησε και κάποιες εικόνες. Πόσες εικόνες ζητήθηκαν από τον εξυπηρετητή;
- 1.29 Μια από αυτές εμφανίζεται στην καρτέλα πριν από το όνομα της σελίδας. Ποιο είναι το όνομα του αρχείου που την περιέχει; [Υπόδειξη: Αναζητήστε τις λέξεις *favorite icon browser* στο Google.]

## 2. Επανάκτηση μη-τροποποιημένης HTML σελίδας

Η ανάγκη για μείωση του χρόνου εξυπηρέτησης των αιτήσεων των πελατών και η ανάγκη για μείωση του φορτίου στις ζεύξεις πρόσβασης των τοπικών δικτύων οδήγησαν στην επινόηση της τεχνικής της ενδιάμεσης αποθήκευσης (Web caching). Στόχος είναι η ταχύτερη ικανοποίηση του αιτήματος του πελάτη ακόμη και χωρίς την ανάμιξη του εξυπηρετητή πηγής. Ο χρήστης, λοιπόν, ρυθμίζει τον πλοηγό του ώστε η περιήγηση στον παγκόσμιο ιστό να γίνεται μέσω ενδιάμεσου εξυπηρετητή (proxy server). Εάν τα ζητούμενα αντικείμενα περιέχονται εκεί (δηλαδή, τα έχει ήδη αναζητήσει κάποιος προηγούμενος) επιστρέφονται από αυτόν. Διαφορετικά, ο ενδιάμεσος εξυπηρετητής τα ζητά από τον εξυπηρετητή πηγής, κρατά ένα αντίγραφο και τα προωθεί στον πελάτη.



Μια συμπληρωματική λύση στο ίδιο πρόβλημα δόθηκε μέσω των πλοηγών ιστού. Συγκεκριμένα, οι πλοηγοί κάνουν τοπική αποθήκευση, δηλαδή, κρατούν στο δίσκο του υπολογιστή διάφορα αντικείμενα από τις ιστοσελίδες που επισκέπτονται. Όταν λοιπόν ο χρήστης επισκεφτεί ξανά την ίδια σελίδα ή ζητήσει ανανέωση (refresh) αυτής που έχει επισκεφτεί, τότε αντί για την HTTP μέθοδο GET εκτελείται η conditional GET. Σύμφωνα με την τελευταία, το ζητούμενο αντικείμενο μεταφέρεται μόνο εάν ισχύουν οι συνθήκες που περιγράφονται στην επικεφαλίδα της εντολής. Για μια πιο λεπτομερή περιγραφή δείτε το παράδειγμα ανταλλαγής μηνυμάτων μεταξύ ενός πλοηγού ιστού και ενός εξυπηρετητή κατά την ανανέωση (ή τη νέα επίσκεψη) μιας ιστοσελίδας στο ακόλουθο αρχείο [https://www.eventhelix.com/Networking/web\\_browser\\_caching\\_tutorial.pdf](https://www.eventhelix.com/Networking/web_browser_caching_tutorial.pdf).

Αφού ξεκινήσετε μια νέα καταγραφή με το Wireshark, επισκεφθείτε με τον πλοηγό ιστού τη σελίδα: <http://edu-dy.cn.ntua.gr/get2.html>. Αμέσως ανανεώστε τη σελίδα πατώντας το κουμπί Reload. Κατόπιν σταματήστε την καταγραφή και εφαρμόστε κατάλληλο φίλτρο απεικόνισης ώστε να παραμείνουν μόνο μηνύματα του πρωτοκόλλου HTTP.

- 2.1 Παρατηρήστε τις επικεφαλίδες πρωτοκόλλου HTTP του πρώτου αιτήματος GET με το οποίο ζητείται το κατέβασμα της σελίδας get2.html. Υπάρχει η επικεφαλίδα If-Modified-Since;
- 2.2 Ποιος είναι ο κωδικός κατάστασης (status code) που επιστρέφει ο εξυπηρετητής ως απόκριση στο μήνυμα αυτό;
- 2.3 Παρατηρήστε τις επικεφαλίδες πρωτοκόλλου HTTP που περιέχονται στην απόκριση. Ποιο είναι το μέγεθος σε byte του περιεχομένου που επιστρέφεται στον πλοηγό ιστού;
- 2.4 Τι είδους αρχείο είναι το περιεχόμενο της σελίδας get2.html;
- 2.5 Πότε τροποποιήθηκε για τελευταία φορά το αρχείο που κατεβάσατε;
- 2.6 Δηλώνει κάποια τιμή ως μέγιστο χρόνο προσωρινής αποθήκευσης; Εάν ναι, για πόσο διάστημα;
- 2.7 Εκτός από την ημερομηνία και ώρα της τελευταίας μεταβολής του πόρου, ο εξυπηρετητής επιστρέφει και μια επιπλέον σήμανση (Entity tag) που αλλάζει σε κάθε μεταβολή του πόρου. Ποιο είναι το όνομα της σχετικής επικεφαλίδας HTTP και ποια η τιμή για τη σελίδα get2.html;
- 2.8 Σε ποιες γλώσσες είναι γραμμένο το κείμενο της ιστοσελίδας get2.html;
- 2.9 Ποιο σύνολο χαρακτήρων (character set) χρησιμοποιείται για το κείμενο της ιστοσελίδας get2.html; [Υπόδειξη: Αναπτύξτε το περιεχόμενο του πεδίου δεδομένων της απάντησης στην οθόνη με τις λεπτομέρειες επικεφαλίδας.]
- 2.10 Τι παρατηρείτε σχετικά με την κωδικοποίηση των ελληνικών χαρακτήρων της σελίδας get2.html; [Υπόδειξη: Παρατηρήστε στο παράθυρο του Wireshark την αναπαράσταση του περιεχομένου του πεδίου δεδομένων της απάντησης σε δεκαεξαδική και ASCII μορφή.]
- 2.11 Παρατηρήστε τις επικεφαλίδες πρωτοκόλλου HTTP του δεύτερου αιτήματος GET με το οποίο ζητείται και πάλι το κατέβασμα της σελίδας get2.html. Υπάρχει επικεφαλίδα If-Modified-Since; Εάν ναι, ποια ημερομηνία και ώρα αναφέρεται;
- 2.12 Υπάρχει επικεφαλίδα If-None-Match που να προσδιορίζει, όπως καταγράψατε στην ερώτηση 2.7, τον συγκεκριμένο πόρο που επέστρεψε ο εξυπηρετητής στην πρώτη απόκρισή του;
- 2.13 Προσδιορίζει ο πλοηγός τη μέγιστη ηλικία της προσδοκώμενης απάντησης μέσω της επικεφαλίδας Cache-Control; Εάν ναι, ποια είναι;
- 2.14 Ποιος είναι ο κωδικός κατάστασης (status code) που επιστρέφει ο εξυπηρετητής ως απόκριση στο δεύτερο αίτημα για κατέβασμα της σελίδας get2.html;
- 2.15 Ο εξυπηρετητής επέστρεψε ρητά τα περιεχόμενα του αρχείου get2.html; Αιτιολογήστε την απάντησή σας.
- 2.16 Πλην των παραπάνω αιτημάτων HTTP υπήρξε άλλο προς τον εξυπηρετητή ιστού; Εάν ναι, ποιος ήταν ο σκοπός του;

### 3. Επανάκτηση τροποποιημένης HTML σελίδας

Όπως είδατε παραπάνω το ζητούμενο αντικείμενο δεν είχε τροποποιηθεί και γι' αυτό τον λόγο ο πλοηγός ιστού απεικόνισε στον χρήστη το ήδη αποθηκευμένο αντικείμενο. Στην αντίθετη περίπτωση, δηλαδή, εάν το ζητούμενο αντικείμενο είχε τροποποιηθεί, τότε ο πλοηγός ιστού θα κατέβαζε το

τροποποιημένο αντικείμενο από τον εξυπηρετητή, θα το αποθήκευε στην προσωρινή μνήμη (cache) και παράλληλα θα το απεικόνιζε στο χρήστη.

Ξεκινήστε μια νέα καταγραφή με το Wireshark. Με τον πλοηγό ιστού επισκεφθείτε τη σελίδα <http://edu-dy.cn.ntua.gr/get.html>. **Αμέσως** ανανεώστε τη σελίδα πατώντας το κουμπί Reload. Περιμένετε **λίγο περισσότερο** από ένα λεπτό και στη συνέχεια ανανεώστε τη σελίδα πατώντας το κουμπί Reload. Κατόπιν σταματήστε την καταγραφή και εφαρμόστε κατάλληλο φίλτρο απεικόνισης ώστε να παραμείνουν μόνο μηνύματα του πρωτοκόλλου HTTP.

- 3.1 Πόσα και τι τύπου αιτήματα HTTP έστειλε ο πλοηγός ιστού σας;
- 3.2 Ποιος είναι ο κωδικός κατάστασης (status code) που επιστρέφει ο εξυπηρετητής ως απόκριση στο πρώτο αίτημα HTTP τύπου GET του πλοηγού ιστού για το κατέβασμα της σελίδας get.html;
- 3.3 Πότε τροποποιήθηκε για τελευταία φορά το περιεχόμενο σύμφωνα με την απόκριση στο πρώτο αίτημα;
- 3.4 Ποιο είναι το μέγεθος σε byte του περιεχομένου που επιστρέφεται στον πλοηγό ιστού και ποια είναι η σήμανση που δηλώνεται στην επικεφαλίδα Etag;
- 3.5 Παρατηρήστε τις επικεφαλίδες πρωτοκόλλου HTTP του δεύτερου αιτήματος τύπου GET με το οποίο ζητείται το κατέβασμα της σελίδας get.html. Υπάρχουν επικεφαλίδες If-Modified-Since, If-None-Match και/ή Cache-Control;
- 3.6 Ποιος είναι ο κωδικός κατάστασης που επιστρέφει ο εξυπηρετητής ως απόκριση στο δεύτερο αίτημα HTTP τύπου GET του πλοηγού ιστού για το κατέβασμα της σελίδας get.html;
- 3.7 Παρατηρήστε τις επικεφαλίδες πρωτοκόλλου HTTP του τρίτου αιτήματος τύπου GET με το οποίο ζητείται το κατέβασμα της σελίδας get.html. Υπάρχουν επικεφαλίδες If-Modified-Since, If-None-Match και/ή Cache-Control;
- 3.8 Ποιος είναι ο κωδικός κατάστασης που επιστρέφει ο εξυπηρετητής ως απόκριση στο τρίτο αίτημα HTTP τύπου GET του πλοηγού ιστού για το κατέβασμα της σελίδας get.html;
- 3.9 Πότε τροποποιήθηκε για τελευταία φορά το περιεχόμενο σύμφωνα με την απόκριση στο τρίτο αίτημα HTTP τύπου GET;
- 3.10 Ποιο είναι το μέγεθος σε byte του περιεχομένου που επιστρέφεται στον πλοηγό ιστού και ποια είναι η σήμανση που δηλώνεται στην επικεφαλίδα Etag; Είναι διαφορετικά από αυτά που βρήκατε προηγουμένως στην ερώτηση 3.4;
- 3.11 Γιατί σας ζητήθηκε να περιμένετε περισσότερο από ένα λεπτό προτού ανανεώσετε τη σελίδα; (Σημείωση: κάποιοι εξυπηρετητές θέτουν την ώρα τελευταίας τροποποίησης των αρχείων ίση με την τρέχουσα ώρα κάθε λεπτό)

#### 4. Ανάκτηση εκτενούς σελίδας HTML

Υπάρχουν περιπτώσεις όπου το ζητούμενο αρχείο είναι αρκετά μεγάλο και δεν χωράει σε ένα τεμάχιο TCP. Στην περίπτωση αυτή το μήνυμα HTTP τεμαχίζεται στο στρώμα μεταφοράς. Σε **υπολογιστή του εργαστηρίου** ξεκινήστε μια νέα καταγραφή με το Wireshark. Με τον πλοηγό ιστού επισκεφθείτε τη σελίδα <http://edu-dy.cn.ntua.gr/long.html>. Σταματήστε την καταγραφή μόλις ολοκληρωθεί το φόρτωμα της σελίδας.

- 4.1 Πόσες συνδέσεις TCP έγιναν;
- 4.2 Ποιο είναι το μέγεθος του MSS που ανακοινώνει η κάθε πλευρά;
- 4.3 Εφαρμόστε κατάλληλο φίλτρο απεικόνισης ώστε να παραμείνουν μόνο μηνύματα του πρωτοκόλλου HTTP. Πόσα τεμάχια TCP απαιτήθηκαν για να μεταδοθεί το αίτημα HTTP;
- 4.4 Να καταγραφεί η γραμμή κατάστασης (status line) της απόκρισης του εξυπηρετητή.
- 4.5 Πόσα τεμάχια TCP απαιτήθηκαν για να μεταδοθεί η απόκριση HTTP;
- 4.6 Να καταγραφεί το μήκος του περιεχομένου του αρχείου long.html όπως αναφέρεται στην προηγούμενη απόκριση HTTP.
- 4.7 Εφαρμόστε νέο φίλτρο απεικόνισης στο Wireshark ώστε να παραμείνει μόνο η κίνηση IP που προέρχεται από τον εξυπηρετητή ιστού. Ποια είναι η σύνταξη του παραπάνω φίλτρου;

- 4.8 Πιο τεμάχιο TCP από αυτά του ερωτήματος 4.5 περιλαμβάνει τη γραμμή κατάστασης του πρωτοκόλλου HTTP που καταγράψατε προηγουμένως; [Υπόδειξη: Αναζητήστε το περιεχόμενο της γραμμής κατάστασης στο παράθυρο με τα περιεχόμενα του επιλεγμένου πλαισίου.]
- 4.9 Ποιο είναι το μέγεθος του περιεχομένου HTTP που μεταφέρει κάθε ένα από τα τεμάχια αυτά πλην του τελευταίου; [Υπόδειξη: Αναπτύξτε το περιεχόμενο της επικεφαλίδας TCP στο παράθυρο με τις λεπτομέρειες επικεφαλίδας.]
- 4.10 Γιατί το μέγεθος των πλαισίων Ethernet που μεταφέρουν τα προηγούμενα τεμάχια TCP πλην του τελευταίου είναι σταθερό;
- 4.11 Πώς προκύπτει το μέγεθος του τελευταίου εξ αυτών;

## 5. Ανάκτηση HTML σελίδας με ενσωματωμένα αντικείμενα

Μια σελίδα HTML μπορεί να είναι συλλογή αντικειμένων (π.χ. ενός αρχείου HTML, διαφόρων εικόνων JPEG, αρχείων ήχου, βίντεο, κλπ) και μπορεί να περιέχει παραπομπές προς άλλες ιστοσελίδες. Στην άσκηση αυτή θα μελετήσετε την κίνηση που παράγεται στην περίπτωση όπου ο πλοηγός ιστού κατεβάζει μια σελίδα που περιέχει ενσωματωμένα αντικείμενα (embedded objects).

Ξεκινήστε μια νέα καταγραφή με το Wireshark αφού **αδειάσετε** την προσωρινή μνήμη (cache) του πλοηγού σας. Στη συνέχεια επισκεφθείτε τη σελίδα <http://edu-dy.cn.ntua.gr/links.html>. Η σελίδα αυτή περιλαμβάνει διευθύνσεις URL που αναφέρονται σε δύο εικόνες που βρίσκονται σε διαφορετικούς από τον [edu-dy.cn.ntua.gr](http://edu-dy.cn.ntua.gr) εξυπηρετητές ιστού. Μόλις η σελίδα φορτωθεί πλήρως, σταματήστε την καταγραφή.

- 5.1 Εφαρμόστε κατάλληλο φίλτρο ώστε να παραμείνουν μόνο τα αιτήματα HTTP προς εξυπηρετητές ιστού. Πόσες εντολές HTTP τύπου GET έχει καταγράψει το Wireshark;
- 5.2 Εκτός από τη σελίδα, ζητήθηκαν και κάποια αρχεία εικόνων. Καταγράψτε τα ονόματα των αρχείων που ζήτησε ο πλοηγός.
- 5.3 Από τις επικεφαλίδες των αιτημάτων HTTP βρείτε τα ονόματα των εξυπηρετητών από τους οποίους ζητήθηκαν τα αρχεία εικόνων.
- 5.4 Εφαρμόστε κατάλληλο φίλτρο ώστε να παραμείνουν μόνο οι αποκρίσεις HTTP. Τι είδους εξυπηρετητές ιστού απάντησαν στα αιτήματα για τα αρχεία εικόνων;
- 5.5 Πότε τροποποιήθηκε για τελευταία φορά το περιεχόμενο της σελίδας links.html και ποιο το μέγεθός του σε byte;
- 5.6 Πότε τροποποιήθηκαν για τελευταία φορά τα αρχεία εικόνων και ποιο το μέγεθός τους;
- 5.7 Ποιες είναι οι διευθύνσεις IPv4 ή IPv6 των εξυπηρετητών ιστού με τους οποίους επικοινωνήσε ο πλοηγός ιστού ώστε να ολοκληρωθεί το κατέβασμα της σελίδας links.html;
- 5.8 Εφαρμόστε κατάλληλο φίλτρο ώστε να παραμείνουν μόνο τα πρώτα τεμάχια TCP των τριμερών χειρασιών. Πόσες συνδέσεις TCP έγιναν με τους προηγούμενους εξυπηρετητές;
- 5.9 Εάν παρατηρήσατε περισσότερες από μία συνδέσεις προς κάποιον από τους προηγούμενους εξυπηρετητές, για πιο λόγο νομίζετε ότι αυτές έγιναν;



Όνοματεπώνυμο:	Ομάδα:
Όνομα PC/ΛΣ:	Ημερομηνία:     /     /
Διεύθυνση IP:     .     .     .	Διεύθυνση MAC:     -     -     -     -     -

## Εργαστηριακή Άσκηση 11

### Πλοήγηση στον παγκόσμιο ιστό

Απαντήστε στα ερωτήματα στον χώρο που σας δίνεται παρακάτω και στην πίσω σελίδα εάν δεν επαρκεί. Το φυλλάδιο αυτό θα παραδοθεί στον επιβλέποντα.

#### 1

- 1.1 .....
- 1.2 .....
- 1.3 .....
- 1.4 .....
- 1.5 .....
- 1.6 .....
- 1.7 .....
- 1.8 .....
- 1.9 .....
- 1.10 .....
- 1.11 .....
- .....
- .....
- 1.12 .....
- .....
- 1.13 .....
- 1.14 .....
- 1.15 .....
- .....
- .....
- 1.16 .....
- 1.17 .....
- .....
- .....
- 1.18 .....
- .....
- .....
- 1.19 .....

1.20	.....
1.21	.....
.....	.....
1.22	.....
.....	.....
1.23	.....
.....	.....
1.24	.....
.....	.....
1.25	.....
.....	.....
1.26	.....
.....	.....
1.27	.....
1.28	.....
1.29	.....
<b>2</b>	
2.1	.....
2.2	.....
2.3	.....
2.4	.....
2.5	.....
2.6	.....
2.7	.....
2.8	.....
2.9	.....
2.10	.....
.....	.....
2.11	.....
2.12	.....
2.13	.....
2.14	.....
2.15	.....
.....	.....
2.16	.....
.....	.....

**3**

3.1 .....

3.2 .....

3.3 .....

3.4 .....

.....

3.5 .....

3.6 .....

3.7 .....

3.8 .....

3.9 .....

3.10 .....

.....

3.11 .....

.....

**4**

4.1 .....

4.2 .....

4.3 .....

4.4 .....

4.5 .....

4.6 .....

4.7 .....

4.8 .....

4.9 .....

.....

4.10 .....

.....

4.11 .....

.....

.....

**5**

5.1 .....

5.2 .....

5.3 .....

.....

- 5.4 .....  
.....
- 5.5 .....  
.....
- 5.6 .....  
.....  
.....
- 5.7 .....  
.....  
.....
- 5.8 .....  
5.9 .....  
.....