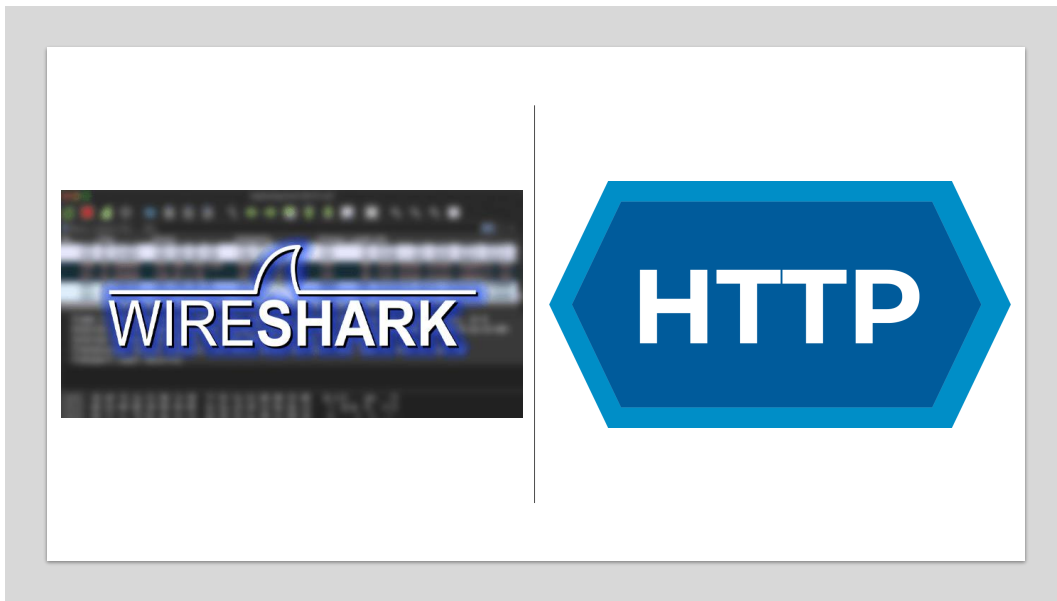


Πρωτόκολλα και Αρχιτεκτονικές δικτύων

Εργαστήριο 5

Τμήμα Πληροφορικής και τηλεπικοινωνιών
Πανεπιστήμιο Ιωαννίνων, Άρτα

2023-05-16



Περιεχόμενα

1	Εισαγωγή	2
2	GET and POST requests	2
3	Η Βασική Αλληλεπίδραση GET/Απόκριση στο HTTP	3
4	Έγγραφα HTML με Ενσωματωμένα Αντικείμενα	5

1 Εισαγωγή

Το πρωτόκολλο HTTP (Hypertext Transfer Protocol) αποτελεί τη βάση του Παγκόσμιου Ιστού και είναι το πρωτόκολλο που χρησιμοποιείται για τη μεταφορά δεδομένων μέσω του Διαδικτύου. Η βασική αλληλεπίδραση μεταξύ ενός πελάτη και ενός διακομιστή στο HTTP είναι η αλληλεπίδραση GET/Απάντηση. Αυτή η αλληλεπίδραση είναι θεμελιώδης για τη λειτουργία του Παγκόσμιου Ιστού και αποτελεί τη βάση όλων των εφαρμογών και υπηρεσιών του Παγκόσμιου Ιστού. Η αλληλεπίδραση GET/Response ξεκινά με έναν πελάτη, όπως ένα πρόγραμμα περιήγησης ιστού, που στέλνει ένα αίτημα GET σε έναν διακομιστή. Το αίτημα GET περιέχει μια διεύθυνση URL (Uniform Resource Locator) που προσδιορίζει τον πόρο στον οποίο ο πελάτης θέλει να έχει πρόσβαση. Στη συνέχεια, ο διακομιστής επεξεργάζεται το αίτημα και στέλνει ένα μήνυμα απάντησης πίσω στον πελάτη. Το μήνυμα απόκρισης περιέχει τον αιτούμενο πόρο, μαζί με μεταδεδομένα και πληροφορίες κατάστασης. Ο κωδικός κατάστασης αποτελεί βασικό μέρος του μηνύματος απόκρισης και υποδεικνύει αν η αίτηση ήταν επιτυχής ή όχι. Υπάρχουν πολλοί διαφορετικοί κωδικοί κατάστασης, αλλά οι πιο συνηθισμένοι περιλαμβάνουν 200 OK (επιτυχής αίτηση), 404 Not Found (πόρος δεν βρέθηκε) και 500 Internal Server Error (σφάλμα διακομιστή). Εκτός από τον κωδικό κατάστασης, το μήνυμα απάντησης μπορεί να περιέχει πρόσθετες επικεφαλίδες που παρέχουν περισσότερες πληροφορίες σχετικά με τον πόρο ή την απάντηση. Για παράδειγμα, η κεφαλίδα Content-Type καθορίζει τη μορφή του πόρου που αποστέλλεται (π.χ. text/html για μια σελίδα HTML). Η αλληλεπίδραση GET/Response είναι η βάση για όλες τις εφαρμογές και υπηρεσίες ιστού. Χρησιμοποιείται για την ανάκτηση δεδομένων από διακομιστές, την ενημέρωση δεδομένων σε διακομιστές και την εκτέλεση άλλων λειτουργιών. Άλλες μέθοδοι HTTP, όπως οι POST, PUT και DELETE, χρησιμοποιούνται για πιο σύνθετες αλληλεπιδράσεις, αλλά όλες βασίζονται στη βασική αλληλεπίδραση GET/Response.

2 GET and POST requests

Στο πρωτόκολλο HTTP (Hypertext Transfer Protocol), οι μέθοδοι GET και POST είναι δύο ευρέως χρησιμοποιούμενες μέθοδοι αίτησης που μπορούν να χρησιμοποιήσουν οι πελάτες για να αλληλεπιδράσουν με τους διακομιστές. Αυτές οι μέθοδοι εξυπηρετούν διαφορετικούς σκοπούς και έχουν διακριτά χαρακτηριστικά.

Αιτήματα GET: Ένα αίτημα GET χρησιμοποιείται για την ανάκτηση ενός πόρου από τον διακομιστή. Πρόκειται για μια απλή και ξεκάθαρη μέθοδο που χρησιμοποιείται κυρίως για την αίτηση δεδομένων. Όταν ένας πελάτης στέλνει ένα αίτημα GET, προσθέτει τις παραμέτρους (εάν υπάρχουν) στην ίδια τη διεύθυνση URL. Για παράδειγμα:

```
GET /api/resource?param1=value1&param2=value2 HTTP/1.1
Host: example.com
```

Σε αυτή την αίτηση, οι παράμετροι `param1` και `param2` περιλαμβάνονται στη διεύθυνση URL. Ορισμένα βασικά χαρακτηριστικά των αιτήσεων GET περιλαμβάνουν:

- **Αιτήματα GET:** Αυτό σημαίνει ότι πολλαπλά πανομοιότυπα αιτήματα GET θα πρέπει να έχουν το ίδιο αποτέλεσμα με ένα μόνο αίτημα. Δεν θα πρέπει να έχουν παρενέργειες στο διακομιστή ή στον πόρο που ζητείται. Τα αιτήματα GET μπορούν να αποθηκευτούν στην προσωρινή μνήμη από τον πελάτη ή τους ενδιάμεσους διακομιστές (όπως οι πληρεξούσιοι), επειδή είναι λειτουργίες μόνο για ανάγνωση. Είναι περιορισμένες ως προς τον όγκο των δεδομένων που μπορούν να αποσταλούν. Τα δεδομένα προσαρτώνται στη διεύθυνση URL, η οποία έχει περιορισμένο μήκος. Δεν πρέπει να χρησιμοποιούνται για ευαίσθητα ή εμπιστευτικά δεδομένα, καθώς οι παράμετροι είναι ορατές στη διεύθυνση URL.
- **Αιτήσεις POST:** Ένα αίτημα POST χρησιμοποιείται για την αποστολή δεδομένων στον διακομιστή για τη δημιουργία ή την ενημέρωση ενός πόρου. Σε αντίθεση με τα αιτήματα GET, τα δεδομένα που αποστέλλονται σε ένα αίτημα POST περιλαμβάνονται στο σώμα του αιτήματος και όχι στη διεύθυνση URL. Για παράδειγμα:

```
POST /api/resource HTTP/1.1
Host: example.com
Content-Type: application/json

{"param1": "value1", "param2": "value2"}
```

Σε αυτή την αίτηση, οι παράμετροι και οι τιμές τους αποστέλλονται ως δεδομένα JSON στο σώμα της αίτησης.

- **Τα βασικά χαρακτηριστικά των αιτήσεων POST περιλαμβάνουν:** Δεν είναι ιδιόσυστατικά. Πολλαπλές πανομοιότυπες αιτήσεις POST μπορεί να έχουν διαφορετικά αποτελέσματα στο διακομιστή ή στον πόρο που τροποποιείται. Τα αιτήματα POST δεν αποθηκεύονται συνήθως στην προσωρινή μνήμη επειδή περιλαμβάνουν τροποποίηση ή ενημέρωση δεδομένων στον διακομιστή. Μπορούν να διαχειριστούν μεγαλύτερες ποσότητες δεδομένων σε σύγκριση με τα αιτήματα GET. Μπορούν να χρησιμοποιηθούν για την αποστολή ευαίσθητων ή εμπιστευτικών δεδομένων, καθώς οι παράμετροι δεν είναι ορατές στη διεύθυνση URL. Συνοπτικά, τα αιτήματα GET χρησιμοποιούνται για την ανάκτηση δεδομένων από τον διακομιστή, ενώ τα αιτήματα POST χρησιμοποιούνται για την αποστολή δεδομένων στον διακομιστή για τη δημιουργία ή την τροποποίηση ενός πόρου. Η κατανόηση των διαφορών μεταξύ αυτών των μεθόδων αίτησης είναι ζωτικής σημασίας για το σχεδιασμό και την υλοποίηση αποτελεσματικών και ασφαλών εφαρμογών ιστού.

Τα βασικά χαρακτηριστικά των αιτήσεων POST περιλαμβάνουν: Δεν είναι ιδιόσυστατικά. Πολλαπλές πανομοιότυπες αιτήσεις POST μπορεί να έχουν διαφορετικά αποτελέσματα στο διακομιστή ή στον πόρο που τροποποιείται. Τα αιτήματα POST δεν αποθηκεύονται συνήθως στην προσωρινή μνήμη επειδή περιλαμβάνουν τροποποίηση ή ενημέρωση δεδομένων στον διακομιστή. Μπορούν να διαχειριστούν μεγαλύτερες ποσότητες δεδομένων σε σύγκριση με τα αιτήματα GET. Μπορούν να χρησιμοποιηθούν για την αποστολή ευαίσθητων ή εμπιστευτικών δεδομένων, καθώς οι παράμετροι δεν είναι ορατές στη διεύθυνση URL.

Συνοπτικά, τα αιτήματα GET χρησιμοποιούνται για την ανάκτηση δεδομένων από τον διακομιστή, ενώ τα αιτήματα POST χρησιμοποιούνται για την αποστολή δεδομένων στον διακομιστή για τη δημιουργία ή την τροποποίηση ενός πόρου. Η κατανόηση των διαφορών μεταξύ αυτών των μεθόδων αίτησης είναι ζωτικής σημασίας για το σχεδιασμό και την υλοποίηση αποτελεσματικών και ασφαλών εφαρμογών ιστού.

3 Η Βασική Αλληλεπίδραση GET/Απόκριση στο HTTP

Ας ξεκινήσουμε την διερεύνηση του HTTP φορτώνοντας ένα πολύ απλό αρχείο HTML, ένα αρχείο το οποίο είναι πολύ μικρό και δεν περιέχει ενσωματωμένα αντικείμενα.

Ακολουθήστε τα παρακάτω βήματα:

- Ξεκινήστε τον web browser σας
- Ξεκινήστε τον packet sniffer Wireshark όπως περιγράφεται στο εισαγωγικό εργαστήριο (αλλά όμως μην ξεκινήσετε τη σύλληψη πακέτων ακόμη). Εισάγετε “http” (χωρίς τα εισαγωγικά) στο παράθυρο προδιαγραφών του φίλτρου παρουσίασης ώστε στο παράθυρο καταλόγου πακέτων να παρουσιάζονται αργότερα μόνο τα συλλαμβανόμενα μηνύματα HTTP. (Στο σημείο αυτό ενδιαφερόμαστε μόνο για το πρωτόκολλο HTTP και δεν θέλουμε να δούμε όλα τα πακέτα που συλλαμβάνονται.)
- Περιμένετε λίγο περισσότερο από ένα λεπτό (σύντομα θα διαπιστώσετε για ποιο λόγο) και μετά αρχίστε τη σύλληψη πακέτων από το Wireshark.
- Εισάγετε το ακόλουθο URL στον browser https://hclab.dit.uoi.gr/Megatron/http_ether1.html.
- Διακόψτε τη σύλληψη πακέτων από το Wireshark.

1654	15.700904	195.130.74.155	34.107.221.82	HTTP	357 GET /canonical.html HTTP/1.1
1670	15.746472	34.107.221.82	195.130.74.155	HTTP	352 HTTP/1.1 200 OK (text/html)
1682	15.812492	195.130.74.155	34.107.221.82	HTTP	359 GET /success.txt?ipv4 HTTP/1.1
1686	15.855529	34.107.221.82	195.130.74.155	HTTP	270 HTTP/1.1 200 OK (text/plain)

Εξετάζοντας την πληροφορία των μηνυμάτων HTTP GET και απόκριση, απαντήστε στις ακόλουθες ερωτήσεις. Εκτυπώστε πρώτα το μήνυμα GET και το μήνυμα απόκρισης (ο τρόπος με τον οποίο μπορείτε να πάρετε μία εκτύπωση εξηγείται στο εισαγωγικό εργαστήριο Wireshark). Σε κάθε απάντησή σας να υποδεικνύετε το σημείο του μηνύματος που περιέχει την πληροφορία που την αιτιολογεί.

1. Ποια έκδοση του HTTP τρέχει στον browser σας; Ποια έκδοση του HTTP τρέχει στον server;

```

Hypertext Transfer Protocol
  GET /canonical.html HTTP/1.1\r\n
    Host: detectportal.firefox.com\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/113.0\r\n
    Accept: */*\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Cache-Control: no-cache\r\n
    Pragma: no-cache\r\n
    Connection: keep-alive\r\n
    \r\n
    [Full request URI: http://detectportal.firefox.com/canonical.html]
    [HTTP request 1/1]
    [Response in frame: 1670]

```

Figure 1: Έκδοση Http Client

```

Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
    Server: nginx\r\n
  Content-Length: 90\r\n
    Via: 1.1 google\r\n
    Date: Tue, 16 May 2023 02:52:41 GMT\r\n
    Age: 31438\r\n
    Content-Type: text/html\r\n
    Cache-Control: public,must-revalidate,max-age=0,s-maxage=3600\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.045568000 seconds]
    [Request in frame: 1654]
    [Request URI: http://detectportal.firefox.com/canonical.html]
    File Data: 90 bytes

```

Figure 2: Έκδοση Http Server

2. Ποιες γλώσσες υποδεικνύει ο browser στον server ότι μπορεί να αποδεχθεί;

Απ: Επιλέγουμε ένα πακέτο ενός GET request και κάνουμε expand το πεδίο **Hypertext Transfer Protocol**

```

Hypertext Transfer Protocol
  GET /success.txt?ipv4 HTTP/1.1\r\n
    Host: detectportal.firefox.com\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/113.0\r\n
    Accept: */*\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
    Pragma: no-cache\r\n
    Cache-Control: no-cache\r\n
    \r\n
    [Full request URI: http://detectportal.firefox.com/success.txt?ipv4]
    [HTTP request 1/1]
    [Response in frame: 1686]

```

```

Transmission Control Protocol, Src Port: 80, Dst Port: 63047, Seq: 1, Ack: 304, Len:
  Source Port: 80
  Destination Port: 63047
  [Stream index: 85]
  [Conversation completeness: Incomplete, DATA (15)]
  [TCP Segment Len: 298]
  Sequence Number: 1 (relative sequence number)
  Sequence Number (raw): 1724289737
  [Next Sequence Number: 299 (relative sequence number)]
  Acknowledgment Number: 304 (relative ack number)
  Acknowledgment number (raw): 4094599826
  0101 .... = Header Length: 20 bytes (5) ←
  > Flags: 0x018 (PSH, ACK)
  Window: 261
  [Calculated window size: 66816]
  [Window size scaling factor: 256]
  Checksum: 0x384a [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0

```

3. Ποια είναι η διεύθυνση IP του υπολογιστή σας; Ποια είναι η διεύθυνση IP του server;

Aπ1: Για την IP του υπολογιστή μας: ipconfig

Aπ2: Για την IP του server: nslookup hcilab.dit.uoi.gr

4. Πότε τροποποιήθηκε για τελευταία φορά στον server το αρχείο HTML το οποίο ανακτήσατε;

Φιλτράρουμε χρησιμοποιώντας το φίλτρο **http.response** και παίρνουμε το αποτέλεσμα της παρακάτω εικόνας.

```

Hypertext Transfer Protocol
  > HTTP/1.1 200 OK\r\n
  Server: nginx\r\n
  > Content-Length: 90\r\n
  [Content length: 90]
  Via: 1.1 google\r\n
  Date: Mon, 15 May 2023 19:04:32 GMT\r\n
  Age: 60537\r\n
  Content-Type: text/html\r\n
  Cache-Control: public,must-revalidate,max-age=0,s-maxage=3600\r\n
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.043307000 seconds]
  [Request in frame: 3828]
  [Request URI: http://detectportal.firefox.com/canonical.html]
  File Data: 90 bytes

```

5. Πόσα bytes περιεχομένου επιστρέφονται στον browser σας;

6. Εξετάζοντας τα ανεπεξέργαστα δεδομένα στο παράθυρο περιεχομένων πακέτου, διαπιστώνετε ότι μέσα στα δεδομένα περιλαμβάνονται επικεφαλίδες οι οποίες δεν εμφανίζονται στο παράθυρο καταλόγου πακέτων; Εάν υπάρχουν τέτοιες επικεφαλίδες, κατονομάστε μία.

4 Έγγραφα HTML με Ενσωματωμένα Αντικείμενα

Ας εξετάσουμε τώρα, τι συμβαίνει όταν ο browser σας φορτώνει ένα αρχείο με ενσωματωμένα αντικείμενα, δηλαδή ένα αρχείο που περιλαμβάνει άλλα αντικείμενα (για παράδειγμα, αρχεία εικόνων) τα οποία είναι αποθηκευμένα σε έναν ή περισσότερους διαφορετικούς servers.

Ακολουθήστε τα παρακάτω βήματα:

- Ξεκινήστε τον web browser σας και βεβαιωθείτε ότι η cache του είναι άδεια όπως συζητήθηκε παραπάνω.

- Ξεκινήστε τον web browser σας και βεβαιωθείτε ότι η cache του είναι άδεια όπως συζητήθηκε παραπάνω.
- Εισάγετε το ακόλουθο URL στον browser https://hcilab.dit.uoi.gr/Megatron/http_ethereal2.html
- Σταματήστε τη σύλληψη πακέτων από το Wireshark και εισάγετε “http” στο παράθυρο προδιαγραφών του φίλτρου παρουσίασης ώστε να παρουσιαστούν μόνο τα συλλαμβανόμενα μηνύματα HTTP.

Απαντήστε στις ακόλουθες ερωτήσεις:

- Πόσα μηνύματα αιτήσεων HTTP GET στάλθηκαν από τον browser σας; Σε ποιες διευθύνσεις IP στάλθηκαν αυτές οι αιτήσεις GET;
- Μπορείτε να διακρίνετε εάν ο browser σας φόρτωσε τις δύο εικόνες σειριακά ή αν οι εικόνες φορτώθηκαν παράλληλα από τους δύο ιστοτόπους; Εξηγείστε.

Η απάντηση εξαρτάται από την ανάλυση των χρονοσφραγίδων των αιτήσεων HTTP GET για τις εικόνες στο Wireshark. Εάν οι χρονοσφραγίδες των αιτήσεων είναι κοντά η μία στην άλλη ή επικαλύπτονται, αυτό υποδηλώνει ότι οι εικόνες φορτώθηκαν παράλληλα. Από την άλλη πλευρά, εάν υπάρχει αξιοσημείωτη χρονική διαφορά μεταξύ των χρονοσφραγίδων των αιτήσεων, αυτό υποδηλώνει ότι οι εικόνες φορτώθηκαν σειριακά. Επομένως, η απάντηση στο αν οι εικόνες φορτώθηκαν σειριακά ή παράλληλα μπορεί να προσδιοριστεί αναλύοντας τις χρονοσφραγίδες των αιτήσεων στο Wireshark και παρατηρώντας τη χρονική σχέση μεταξύ τους.