**Лабораторийн ажлийн тайлан №2**

HTTP ПРОТОКОЛ

**МТЭС, Програм хангамж**

**О.Энэрэл 22B1NUM0506**

# Ажлын зорилго

Энэхүү лабораторийн ажлаар TCP/IP сүлжээний хэрэглээний давхаргын протокол болох HTTP, түүний хувилбарууд, ажиллагааны зарчмын талаар судална.

# Үндсэн ойлголт

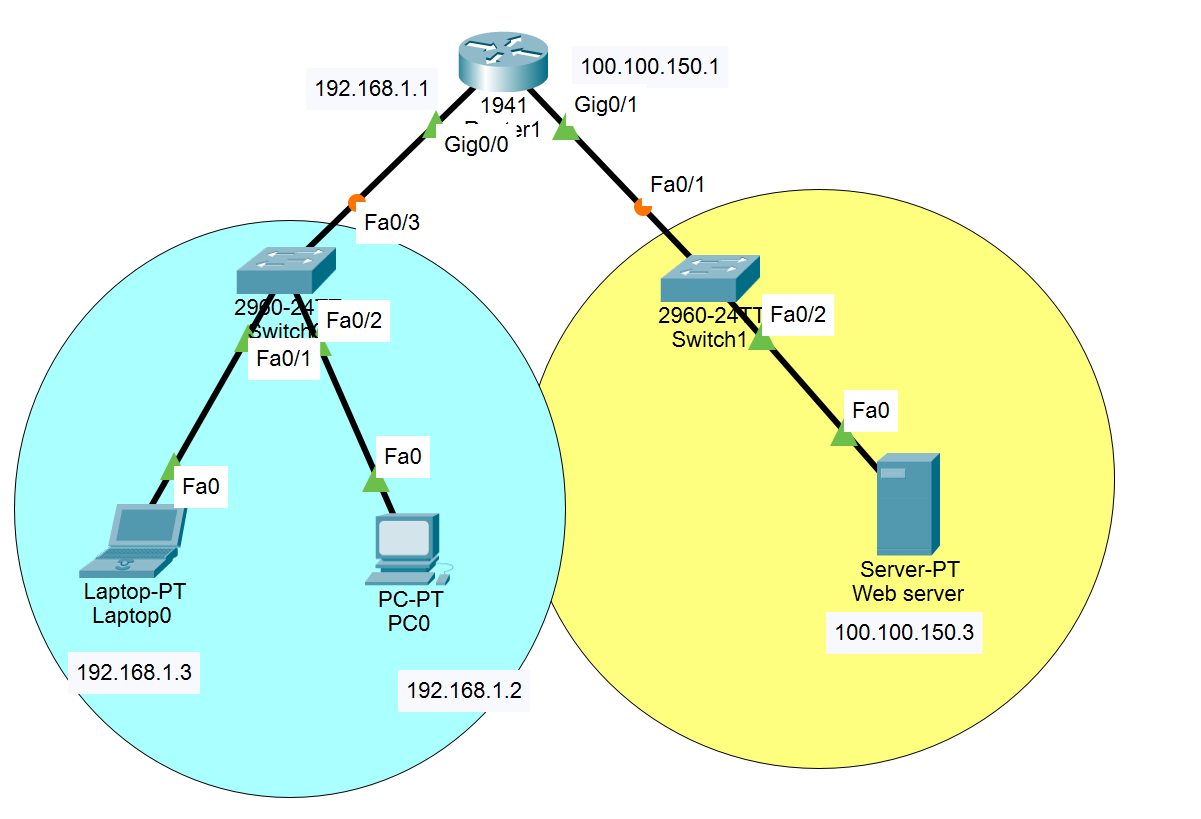
HTTP (Hypertext Transfer Protocol) протокол нь веб сервер болон веб хөтөч хооронд HTTP мессежүүдийн тусламжтайгаар вебийн өгөгдлийг тээвэрлэх, солилцох зорилготой холболтын протокол юм. HTTP-ийн анхны хувилбар болох 1.1 нь 1997 онд гарч байсан бол дараа нь HTTP/2 болон HTTP/3 гэсэн хувилбарууд бий болсон. HTTPS (HTTP over TLS/SSL) нь веб сервер болон веб хөтөч хооронд дамжуулж буй өгөгдлийг шифрлэж аюулгүй байдлыг хангадаг протокол юм. HTTP протокол нь хүсэлт-хариу (request-response) гэсэн клиент-сервер архитектураар ажилладаг. TCP/IP сүлжээний тээвэрлэлтийн давхарга (Transport layer)-д TCP протоколыг ашиглах ба клиент нь сервер рүү three-way handshaking ашиглан TCP холболт үүсгэж холболтыг эхлүүлдэг. Үүний дараа сокет (socket) холболт хийгдэнэ. Клиент нь HTTP GET хүсэлтийг HTTP сервер рүү илгээдэг бол хүсэлт ирсний дараа HTTP сервер нь хүсэлтийн дагуу HTTP response илгээдэг. Ингэхдээ харгалзан TCP протокол, портын дугаарын дагуу хүсэлт болон хариултаа илгээнэ. HTTP протокол нь TCP 80 портыг, HTTPS нь TCP 443 портыг тус тус ашигладаг. HTTP нь веб хөтөч болон веб сервер хоорондын өгөгдөл солилцоход GET, POST, PUT, DELETE, HEAD, OPTIONS зэрэг мессежүүдийг ашиглана.

HTTPS нь HTTP-ийн аюулгүй байдлыг сайжруулсан хувилбар юм. Тээвэрлэлтийн давхарга дээр TCP 443 портыг ашигладаг. HTTPS нь HTTP мессежүүдийг TLS (Transfer Layer Security) протоколын тусламжтайгаар шифрлэж дамжуулдаг бөгөөд гoл зорилго нь хандсан веб хуудсын мэдээллийг нууцлал солилцох, түүний аюулгүй байдал, бүрэн бүтэн байдлыг хангах юм. Жишээлбэл, клиент болон сервер хоёpын хооронд eavesdrop, man-in- the-middle халдлагуудаас болон агуулгыг хуурамчаар үйлдэх зэргээс хамгаалж, холболтын нууц шифрлэлт үүсгэдэг. HTTP протоколтой холбоотой зайлшгүй хоёр зүйл нь URL болон DNS юм.

URL (Uniform Resource Locator) - Веб хуудасны файл, зураг, видеоны байршлыг заахад URL-ийг ашигладаг. DNS (Domain Name System) - веб ажиллаж байгаа серверийн IP хаягийг домэйн нэрийн хэлбэрт шилжүүлснээр хэрэглэгчийн веб хандалтыг хялбаршуулдаг.

# Туршилт

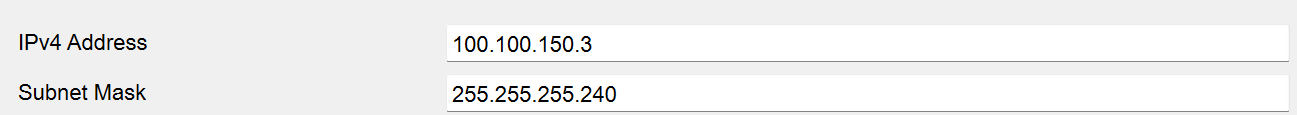
Packet Tracer программ дээр Зураг 2.1-т харуулсан топологийг байгуулаарай.



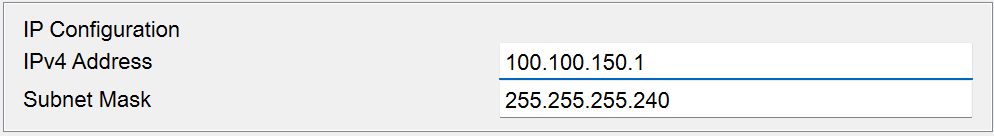
*Figure 1Туршилтын топологи*

Хүснэгт 2.1 Хаягийн хүснэгт

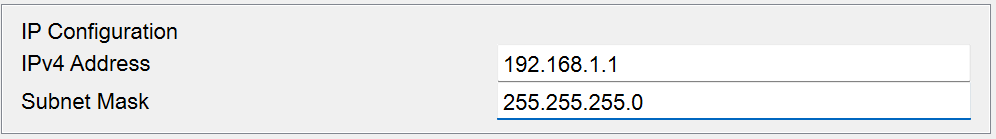
|  |  |  |
| --- | --- | --- |
| **Төхөөрөмжийн нэр** | **IP хаяг** | **Subnet Мask** |
| Web server | 100.100.150.3 | 255.255.255.240 |
| R1, Gig0/1 | 100.100.150.1 | 255.255.255.240 |
| R1, Gig0/0 | 192.168.1.1 | 255.255.255.0 |
| PC0 | 192.168.1.2 | 255.255.255.0 |
| Laptop | 192.168.1.3 | 255.255.255.0 |
|  |  |  |

Төхөөрөмжийн IP хаягийг тохируулах

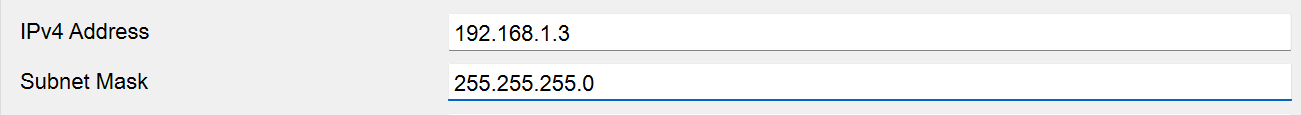
*Figure 2 web server*



*Figure 3 R1, Gig0/1*



*Figure 4R1, Gig0/0*



*Figure 5 Laptop*

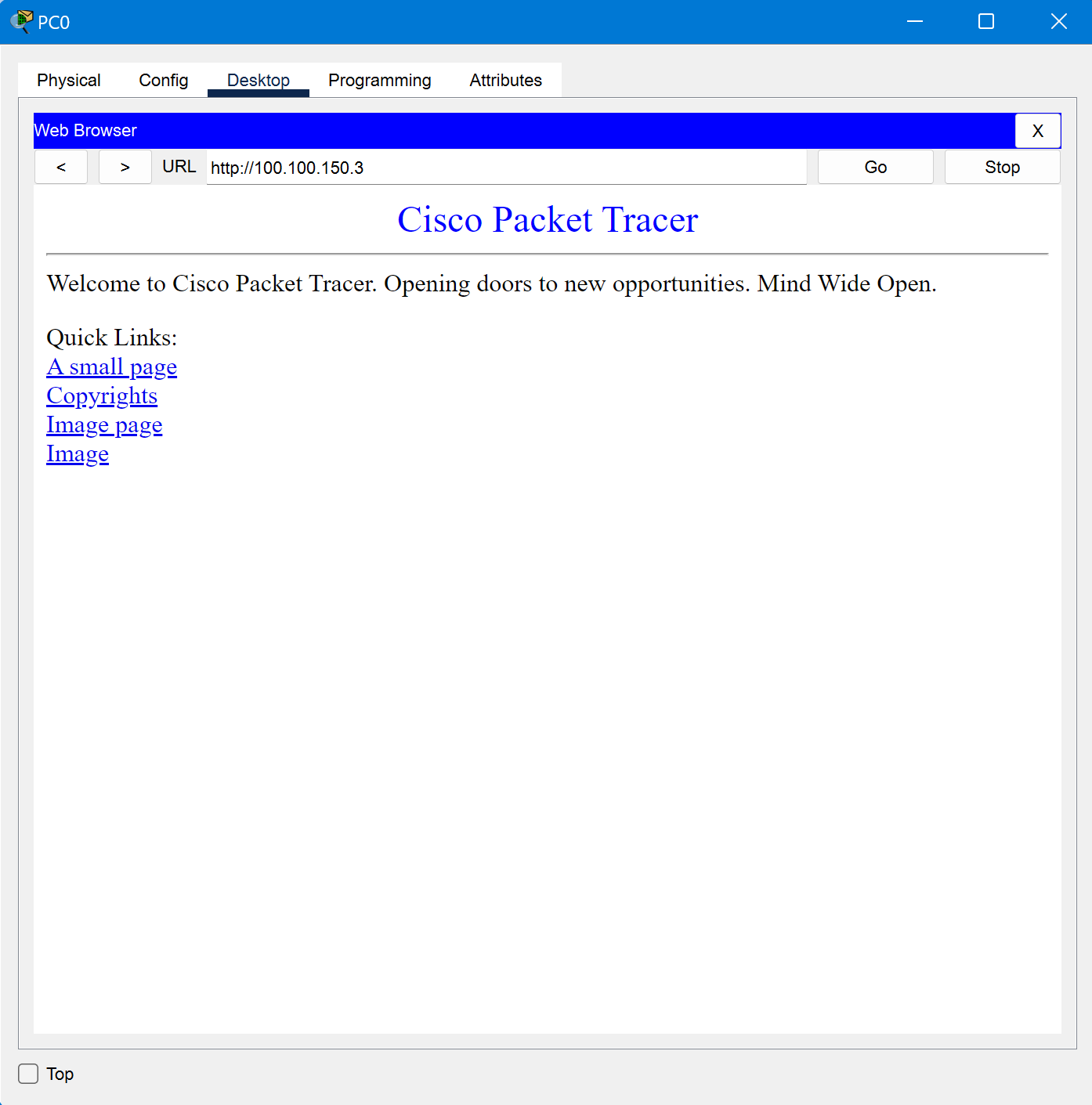


*Figure 6 PC*

# Даалгавар

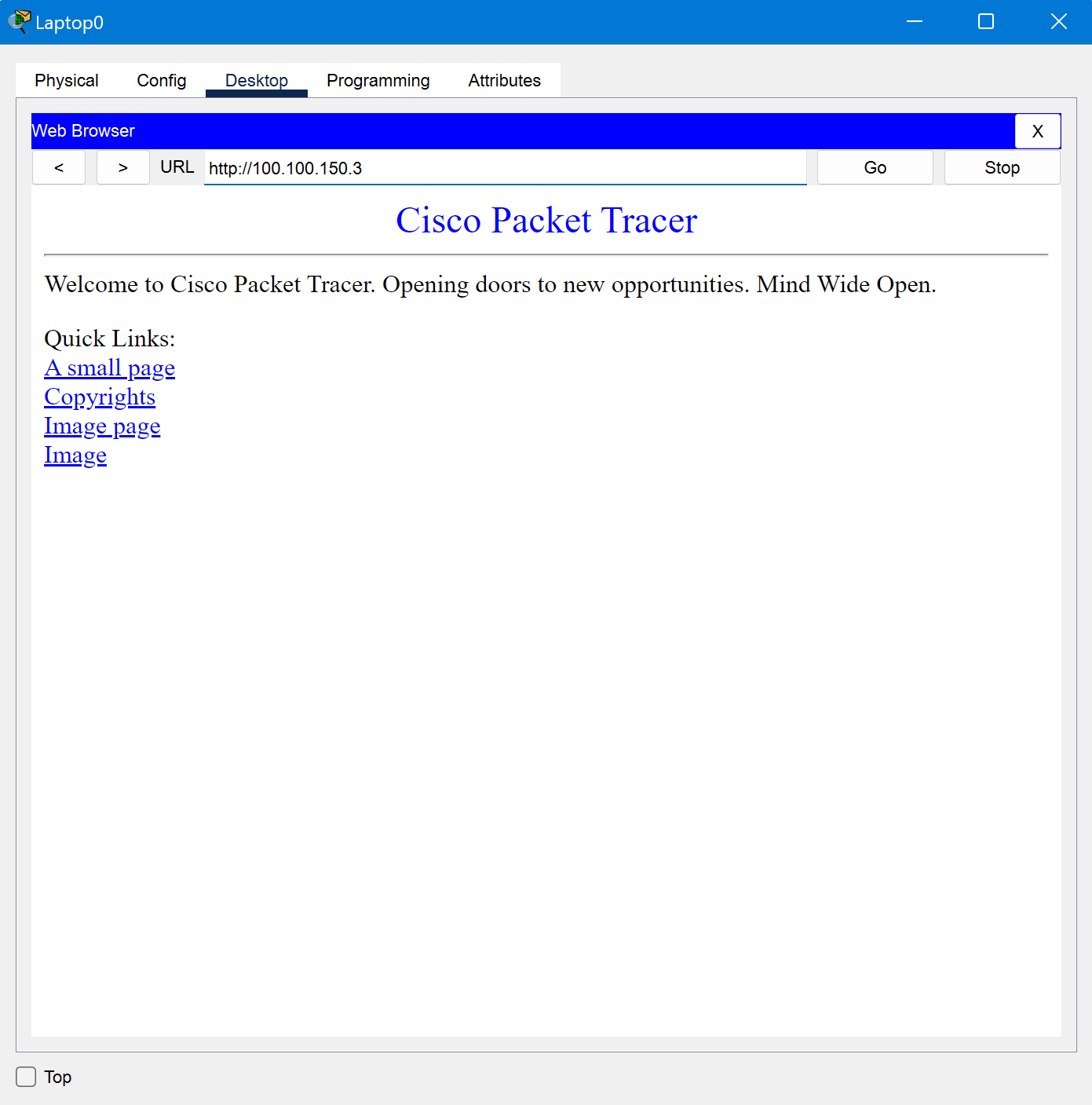
Дээрх топологийг байгуулсны дараа *Desktop -> Web browser* хэсэгт PC0-ээс Web server рүү HTTP -ээр хандах, Laptop -оос Web server рүү хандаж веб хуудасны ажиллагааг шалгах бөгөөд Зураг 2.3-т үзүүлсэнтэй төсөөтэй үр дүн гарна.

PC0-ээс Web server рүү HTTP -ээр хандах

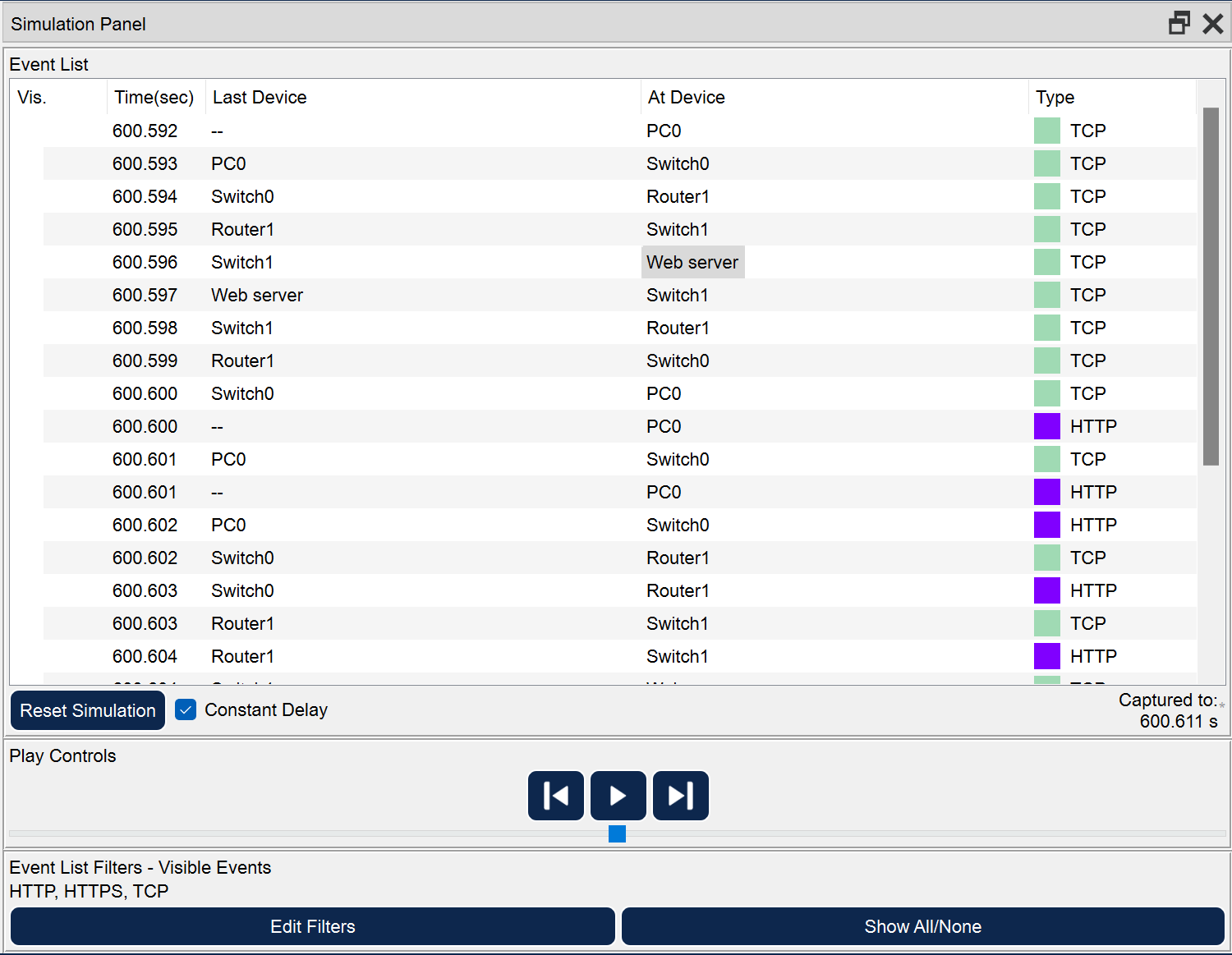


*Figure 7PC0-ээс Web server рүү HTTP -ээр хандасан үр дүн*

Laptop -оос Web server рүү хандана



*Figure 8 Laptop -оос Web server рүү хандсан үр дүн*



*Figure 9 Simulation Panel-ийн үр дүн*

Simulation Panel-ийг ажиглаж TCP холболтын процессыг тайлбарласан нь:

PC0 SYN flagтай TCP segment-ийг web server руу илгээсэн.

Switch0 пакетийг дараагийн порт руу илгээсэн

Router destination IP хаягийг CEF table-ээс хайгаад пакетийг дараагийн порт руу чиглүүлсэн.

Switch1 хүлээн авагчийн IP хаягтай харгалзах МАС хаягийг МАС table-ээс хайгаад олсонгүй тул бүх порт руу пакетийг илгээсэн.

WebServer-ийн IP хаяг нь хүлээн авагчийн IP хаягтай ижилхэн байсан тул пакетийг задалсан.

WebServer нь портийн дугаар нь 80 болох TCP SYN segment – ийг хүлээн авч TCP SYN+ACK segment-ийг пакет илгээгч руу илгээсэн.

PC0 TCP SYN+ACK segment-ийг хүлээн авч TCP холболт үүссэн. PC0 TCP ACK segment-ийг WebServer-руу илгээсэн. HTTP request-ийг server руу илгээсэн.

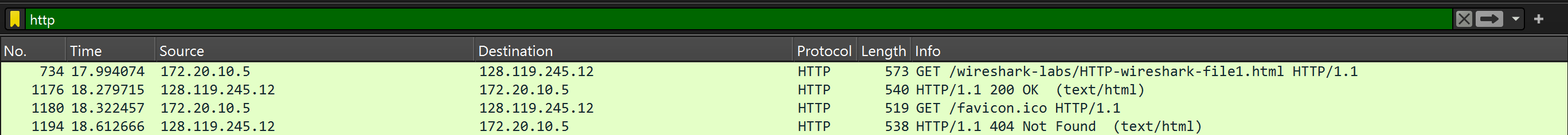
WebServer нь TCP ACK segment-ийг хүлээн авсан. TCP PUSH+ACK segment-ийг РС0 руу илгээж РС0 хүлээн авсан.

РС0 80 порт дээр 100.100.150.30 -ийн TCP холболтоо салгаж, TCP FIN+ACK segment-ийг илгээсэн.

# Даалгавар

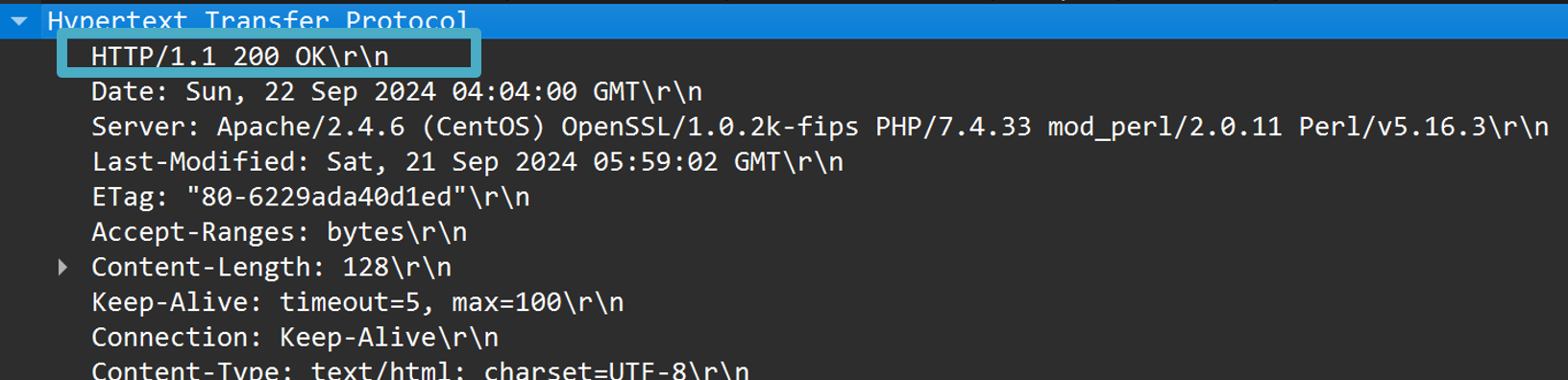
*Туршилт 1. Үндсэн HTTP GET/response мессеж*

Wireshark программ ажиллуулж холбогдох интерфейсийн сүлжээний урсгалыг цуглуулж эхлээрэй. Веб хөтчийг нээж <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html> хаяг руу хандаж энгийн HTML хуудсыг дуудах ба Wireshark програмыг зогсооно. Хоёр HTTP мессежийг сонгож дараах асуултад хариулаарай.



*Figure 10 хаяг руу хандсан үр дүн*

1. Веб хөтөч HTTP 1.0, 1.1 эсвэл 2.0 аль хувилбар дээр ажиллаж байна вэ? Сервер талд HTTP ийн аль хувилбарыг ашиглаж байна вэ? Эдгээр нь хоорондоо ямар ялгаатай болох, 1.1 дээр ямар давуу талууд бий болсон бэ?



*Figure 11 http хувилбар*

Веб хөтөч нь HTTP 1.1 хувилбар дээр ажиллаж байна, Сервер талд бас HTTP 1.1 ииг хувилбарыг ашиглаж байна.

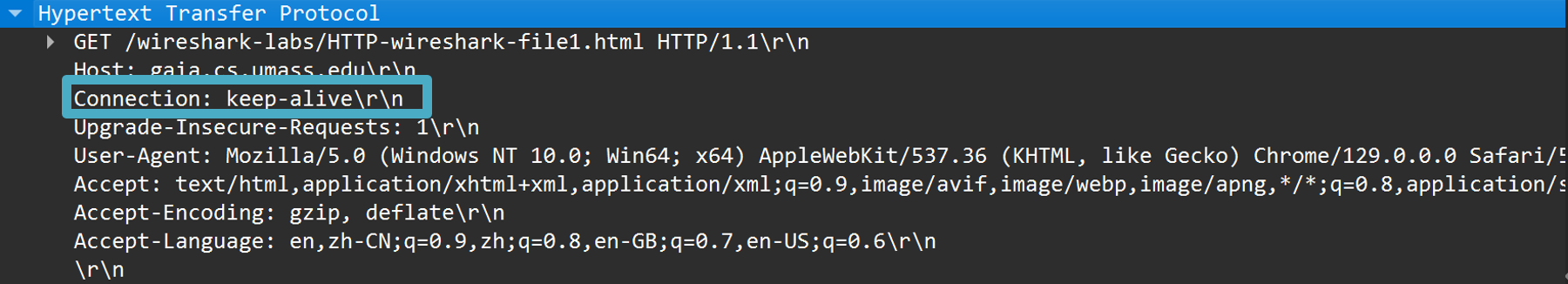
HTTP хувилбруудын ялгаа:

|  |  |
| --- | --- |
| HTTP 1.0 | Клентээс сервэр руу хийсэн хүсэлт бүрд шинэ холболт үүсгэдэг. Холболт нь хүсэлт илгээж, хариултыг авсны дараа салдаг. GET, POST, HEAD хүсэлтүүдийг дэмждэг. |
| HTTP 1.1 | keep-alive холболт ашигланолон хүсэлтийг нэг холболтоор явуулах боломжтой. PUT, DELETE, болон OPTIONS зэрэг шинэ аргууд нэмэгдсэн. Кэшийн хяналтын механизмууд сайжирсан |
| HTTP/2 | Multiplexing нь нэг холболтоор олон хүсэлт, хариулт явуулах боломжийг олгодог. Толгойн мэдээллийг шахах нь ачааллыг багасгадаг. |
| HTTP/3 | QUIC протоколыг ашигладаг, QUIC нь UDP-ийн дээр суурилсан протокол юм. QUIC нь 0-RTT (Zero Round Trip Time) технологиор дамжуулан өмнө тогтоосон холболтоо ашиглаж, илүү хурдан холболт үүсгэж эхэлдэг. |

HTTP 1.1 дээр ямар давуу талууд бий болсон бэ?

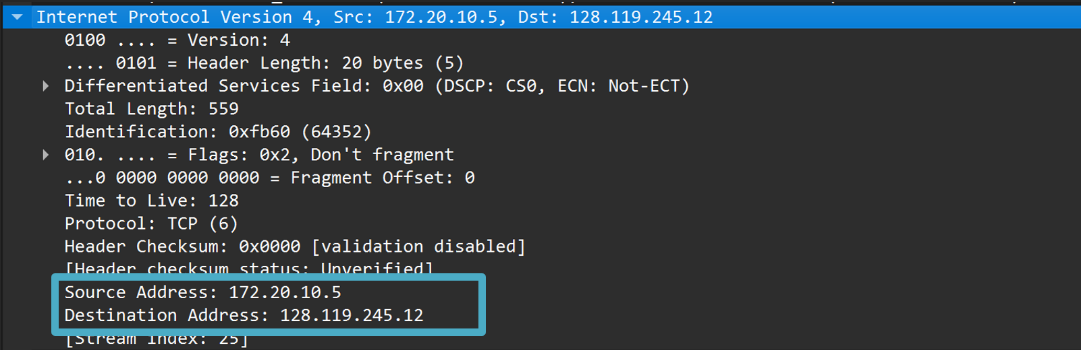
Нэг холболтоор олон хүсэлт явуулж, сервер болон клиент хооронд өгөгдөл дамжуулах хурд сайжирсан. Chunked Transfer Encoding нь динамик веб хуудсыг илүү хурдан ачаалахад тусалдаг. Өгөгдлийг илүү нарийн кэшлэх боломжтой болсон нь вэб ачаалал болон сүлжээний нөөцийг үр ашигтай ашиглахад нөлөөлсөн.

1. HTTP 1.1 хувилбар дээр TCP холболт дамжууллын туршид хадгалагддаг бол, үүнийг ямар мессежний аль талбарын утгаас харж болох вэ?



*Figure 12 keep-alive message*

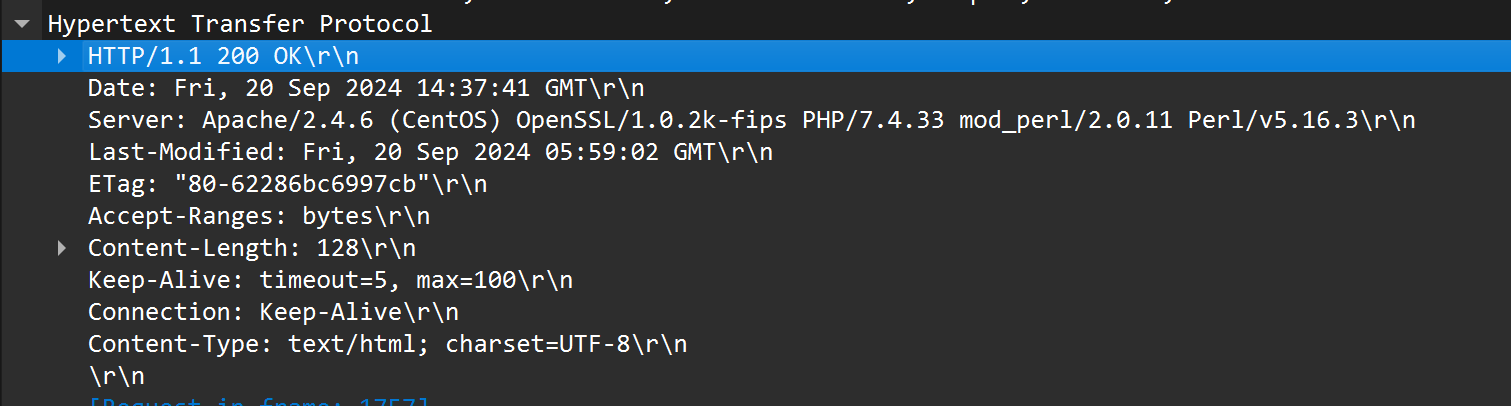
HTTP 1.1 хувилбар дээр TCP холболт дамжууллын туршид хадгалагддагийг Hypertext Transfer Protcol-ийн Connection: keep-alive мессежээс харах боломжтой.

1. ****Өөрийн компьютерын болон gaia.cs.umass.edu серверийн IP хаяг ямар байна вэ?

*Figure 13 sourceIP, DestinationIP*

Миний компьютерын IP хаяг нь 172.20.10.5 байж, gaia.cs.umass.edu серверийн IP хаяг нь 128.119.245.12 байна.

1. Серверээс ирж байгаа хуудасны төлөвийн код (status code) ямар байна вэ? Энэ нь ямар утгыг илэрхийлж байна? Бусад төлөвийн кодыг интернэтээс бие даан судалж тайланд тайлбарлаж оруулна уу.



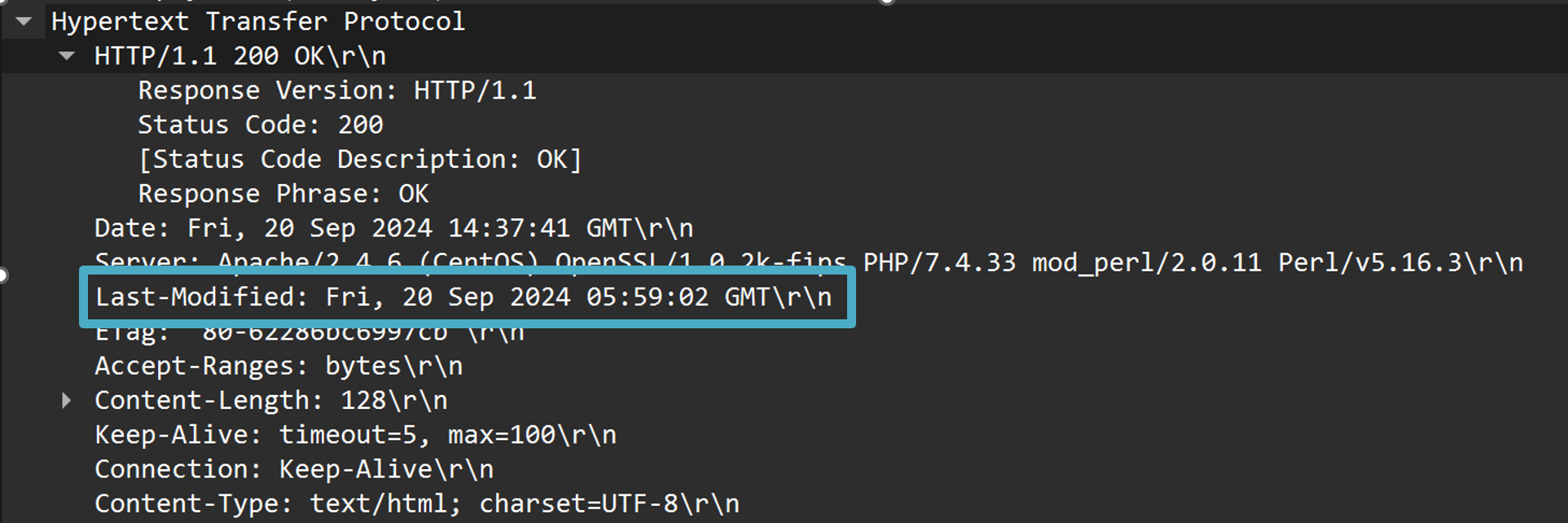
*Figure 14 Status code*

Серверээс ирж байгаа хуудасны төлөвийн код 200OK байна, Энэ нь Хүсэлт амжилттай боловсруулагдсан бөгөөд сервер хариуг илгээж байгааг илэрхийлнэ.

Бусад төлөвийн код

|  |  |
| --- | --- |
| Status code | Тайлбар |
| 100 Continue | Хүсэлт амжилттай эхэлсэн бөгөөд клиент дараагийн хэсгээ илгээж болно. |
| 101 Switching Protocols | Сервер протоколыг өөрчлөх хүсэлтийг хүлээн авсан. |
| 200 OK | Хүсэлт амжилттай боловсруулагдсан бөгөөд сервер хариуг илгээж байна. |
| 201 Created | Хүсэлт амжилттай байсан бөгөөд сервер шинэ нөөцийг үүсгэсэн. |
| 202 Accepted | Хүсэлт хүлээн авсан ч сервер түүнийг шууд боловсруулж дуусгаагүй байна. |
| 204 No Content | Хүсэлт амжилттай байсан ч хариуд агуулгагүй байна (жишээ нь PUT, DELETE хүсэлтэнд ашиглагддаг). |
| 301 Moved Permanently | Хүсэлт хийгдсэн URL байнгын өөр хаяг руу шилжсэн. |
| 302 Found | Түр зуур өөр хаяг руу шилжсэн. Хуучин хаяг нь хүчинтэй хэвээр байна. |
| 304 Not Modified | Клиентын кэш дээрх нөөц өөрчлөгдөөгүй тул серверээс дахин татах шаардлагагүй. |
| 400 Bad Request | Хүсэлт буруу бүтэцтэй эсвэл сервер ойлгож чадаагүй. |
| 401 Unauthorized | Нэвтрэлт хийгээгүй буюу буруу нэвтрэлттэй хүсэлт. |
| 403 Forbidden | Клиент эрхгүй нөөц рүү хандахыг оролдсон. |
| 404 Not Found | Хүсэлтэнд заагдсан нөөцийг сервер олж чадаагүй. |
| 405 Method Not Allowed | Хүсэлтэд ашигласан HTTP арга нь тухайн нөөц дээр боломжгүй. |
| 500 Internal Server Error | Сервер дотоод алдаа гаргасан бөгөөд хүсэлтийг боловсруулж чадаагүй. |
| 501 Not Implemented | Сервер хүсэлтэнд ашигласан аргыг дэмждэггүй. |
| 502 Bad Gateway | Сервер өөр серверээс буруу хариу хүлээн авсан |
| 503 Service Unavailable | Сервер түр зуур ачаалалтай эсвэл засвартай тул хариу өгөх боломжгүй. |
| 504 Gateway Timeout | Сервер өөр серверээс хариу авах хугацаа хэтэрсэн. |

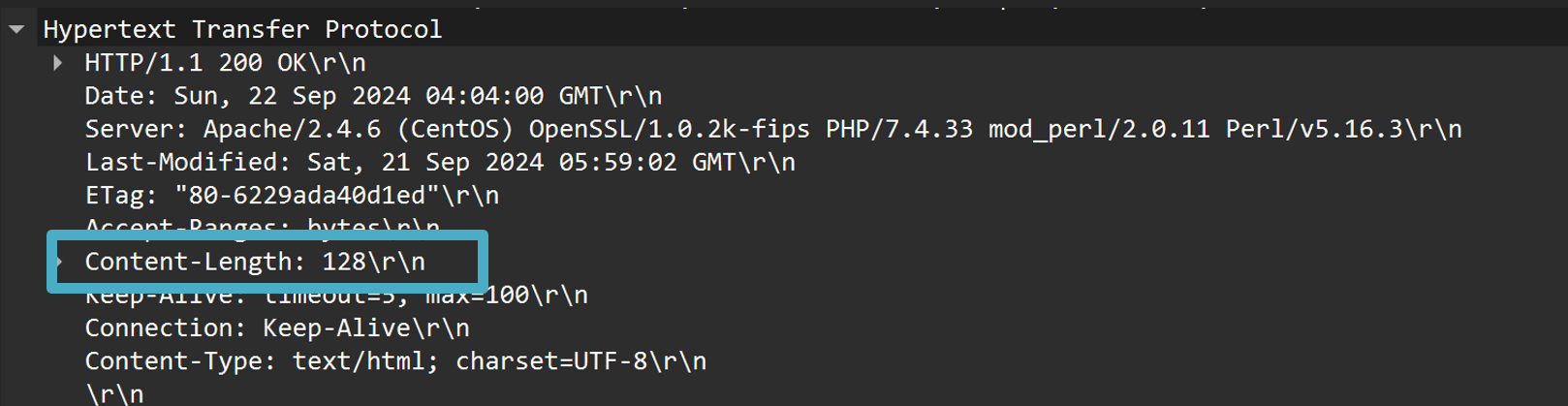
1. Серверээс ирж байгаа html хуудас сервер дээр хамгийн сүүлд хэзээ өөрчлөлт орсон байна вэ?



*Figure 15 Last modified*

HTTP түвшины Last modified талбараас харвал хамгийн сүүлд өөрчлөлт орсон онгноо нь Fri, 20 Sep 2024 05:59:02.

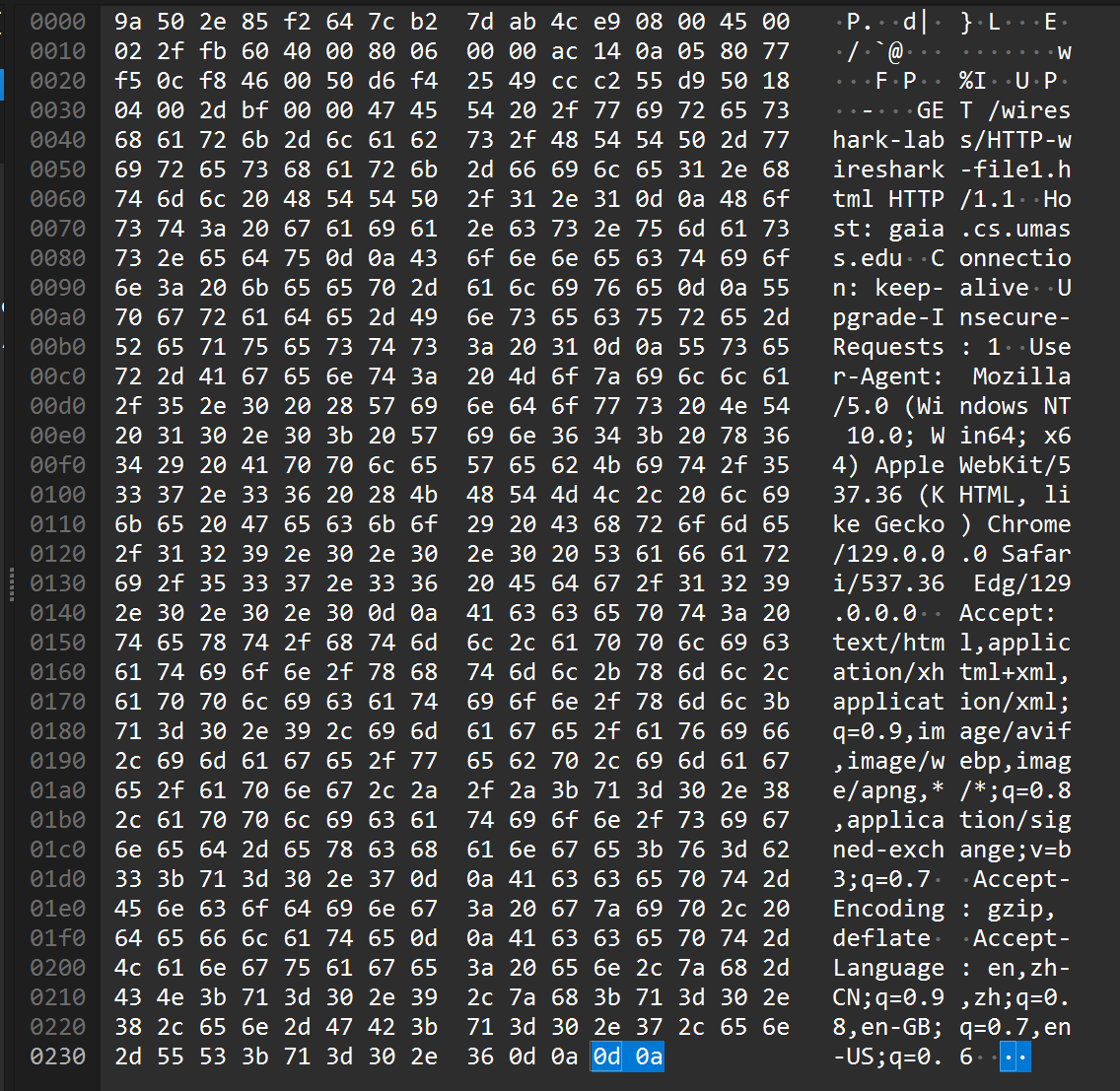
1. Веб хөтөч дээр (Application layer) хэдэн байтын өгөгдөл ирсэн байна бэ?



*Figure 16 Content-length*

Веб хөтөч дээр 128 байтын өгөгдөл ирсэн байна.

1. Пакетыг хоёрт эсвэл арван зургаатаар харах боломжтой цонхонд харуулж байгаа (packet content window) түүхий өгөгдлийг (raw data) сайтар ажиглаж, уг өгөгдөл дотор TCP/IP -аар задалж харуулсан цонхонд (packet-details window) агуулагдаагүй хэсгийн мэдээллүүдээс багадаа нэгийг олж, тайланд жишээгээр оруулна уу.



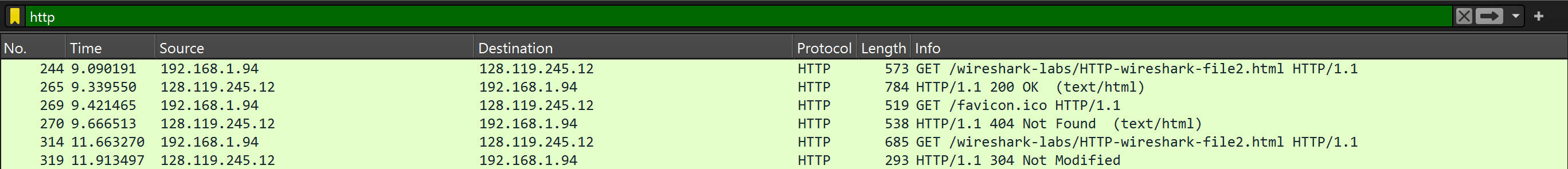
*Figure 17 packet content window*

47 45 54 нь ASCII-д GET гэж бууна (эхлэх байрлал 0030).

48 54 54 50 нь ASCII-д HTTP (эхлэх байрлал 003C).

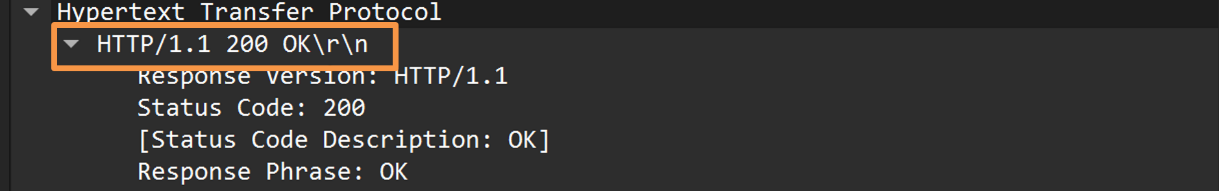
*Туршилт 2. CONDITIONAL GET/response мессеж*

Эхлээд туршилт хийхээс өмнө веб хөтчийн кэшийг цэвэрлэх шаардлагатай. F5 дарж хуудсыг дахин дуудахад ямар процесс болдог талаар судална. Wireshark программ ажиллуулж, холбогдох интерфейсийн сүлжээний урсгалыг цуглуулж эхлээрэй. Веб хөтчийг нээж <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html> хандаж хуудсыг дуудах ба F5 товчийг дарж уг хуудсыг дахин дуудаарай. Wireshark програмыг зогсоож шүүлтүүр талбарт “http” утгыг оруулах бөгөөд дараах асуултад хариулна уу. Энд зөвхөн HTTP протоколоор дамжсан сүлжээний пакетууд харагдана.



1. Эхний HTTP GET хүсэлт мессежний агуулгыг ажиглан “IF-MODIFIED-SINCE” мөр байна уу?

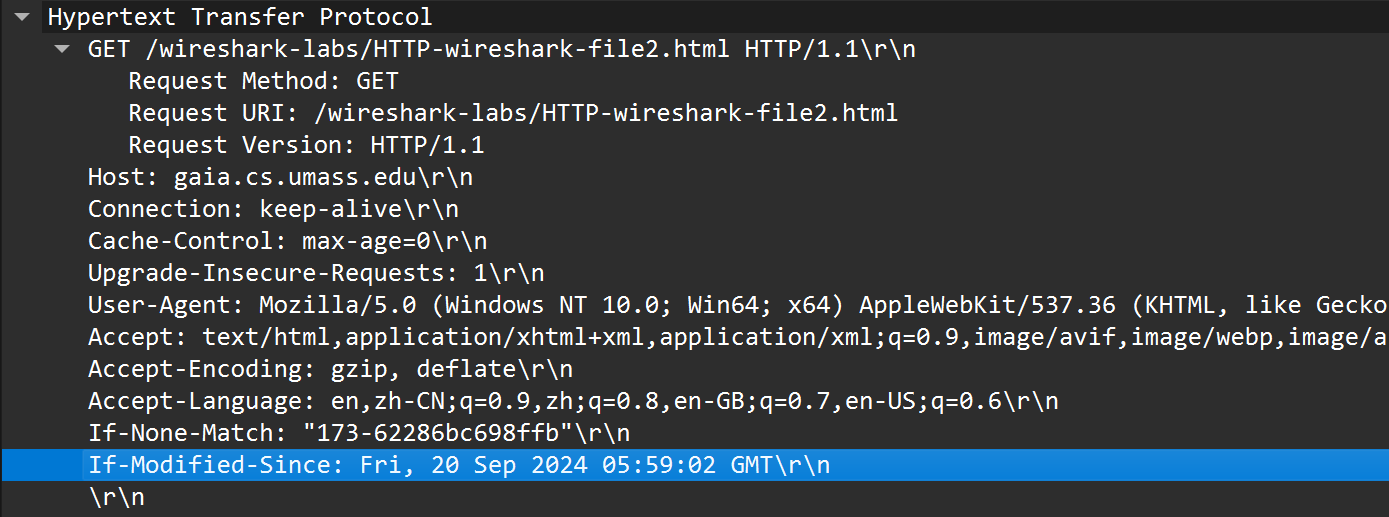
“IF-MODIFIED-SINCE” мөр байхгүй байна.

1. Серверээс хариу файлыг амжилттай авсан уу? Үүнийг хэрхэн мэдэх вэ?

*Figure 18 Status code*

Status code нь 200 байсан тул серверээс хариу файлыг амжилттай авсан.

1. F5 дарах үеийн буюу хоёр дахь HTTP GET хүсэлтийн агуулгыг ажиглаж, “IF- MODIFIED-SINCE” талбар байна уу, байвал ямар утга агуулж байгааг тайлбарлана уу.



*Figure 19 if modified-since*

If-Modified-Since: Fri, 20 Sep 2024 05:59:02 ээс хойш вэб хуудас өөрчлөгдсөн эсэхийг шалгаж кэшлэгдсэн нөөцийг дахин татах шаардлагатай эсэхийг шалгахад хэрэглэдэг.

1. Хоёр дахь HTTP GET мессежний хариу мессежинд ирсэн төлөвийн код болон нэршил (status code, phrase) ямар байна вэ? Энэ код нь ямар утгыг илэрхийлэх вэ?

304 Not Modified клиентын кэш дээрх өгөгдөл өөрчлөгдөөгүй тул серверээс дахин татах шаардлагагүйг илэрхийлнэ.

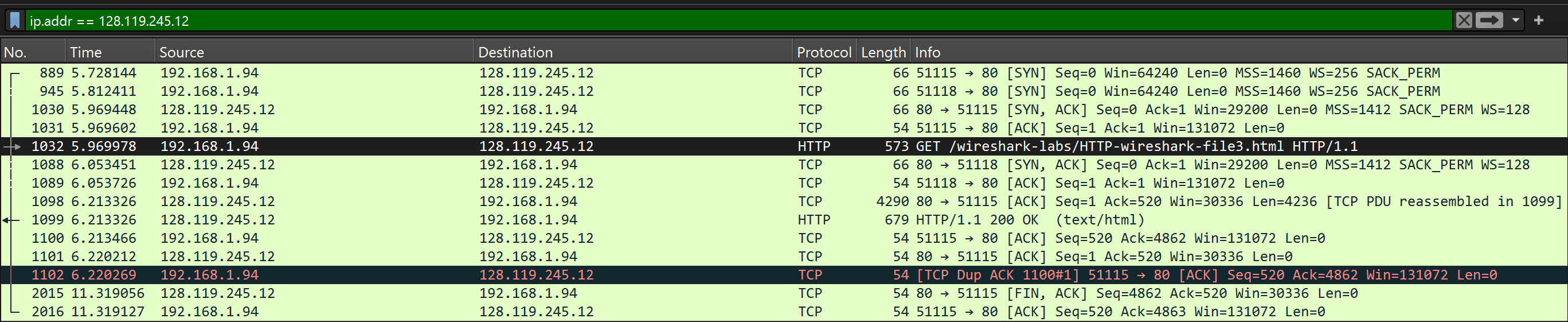
1. Серверээс хоёр дахь хүсэлтэд хариу файл буцаасан уу? Үүнийг хэрхэн мэдэх вэ?

Серверээс хоёр дахь хүсэлтэд хариу файл буцаасангүй, харин клиентын кешээс ирсэн үүнийг төлвийн кодоос мэдэх боломжтой.

*Туршилт 3. Их хэмжээний өгөгдөл татах*

Одоо их хэмжээний өгөгдөл татаж авах тухай судална. Wireshark програмыг ажиллуулж, холбогдох интерфейсийн сүлжээний урсгалыг цуглуулж эхлээрэй. Веб хөтчийг нээж http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html хандаж хуудсыг дуудна. Энд http протоколоор веб рүү хандсан эсэхийг нягтална уу. Wireshark програмыг зогсоож, шүүлтүүр талбарт “ip.addr = = [*server IP*]” утгыг оруулах бөгөөд дараах асуултад хариулна уу. Уг HTTP-wireshark-file-3.html хуудас нь 4500 байтын хэмжээтэй тул нэг TCP пакетаар дамжуулах боломжгүй. Тиймээс ямар хэмжээтэй хэд хэдэн TCP сегментэд хуваагдаж дамжсан, мөн фремийн дугаартай байгаа зэргийг ажиглаад дараах асуултад хариулна.

1. Веб хөтчөөс хэдэн HTTP GET мессеж илгээгдсэн байна вэ?

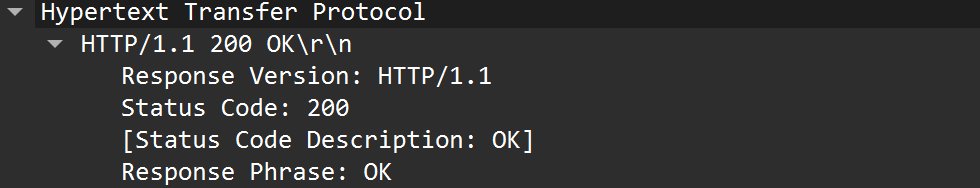


Веб хөтчөөс нэг HTTP GET мессеж илгээгдсэн.

1. Хэд дугаартай пакетад HTTP GET мессежний хариутай холбоотой төлөвийн код, нэршил агуулагдаж байна вэ?

1099-р пакетад HTTP GET мессежний хариутай холбоотой төлөвийн код, нэршил агуулагдаж байна.

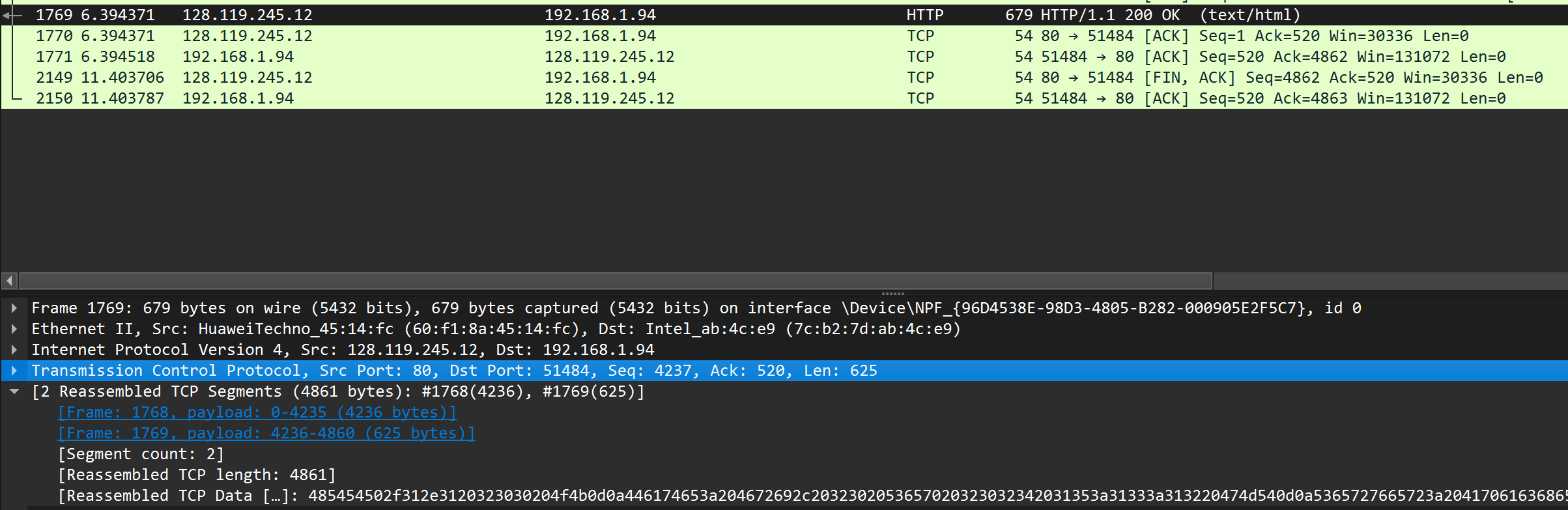
1. Хариу мессежид агуулагдаж байгаа төлөвийн код, нэршил ямар байна вэ?



*Figure 20 status code*

Төлвийн код нь 200 ОК.

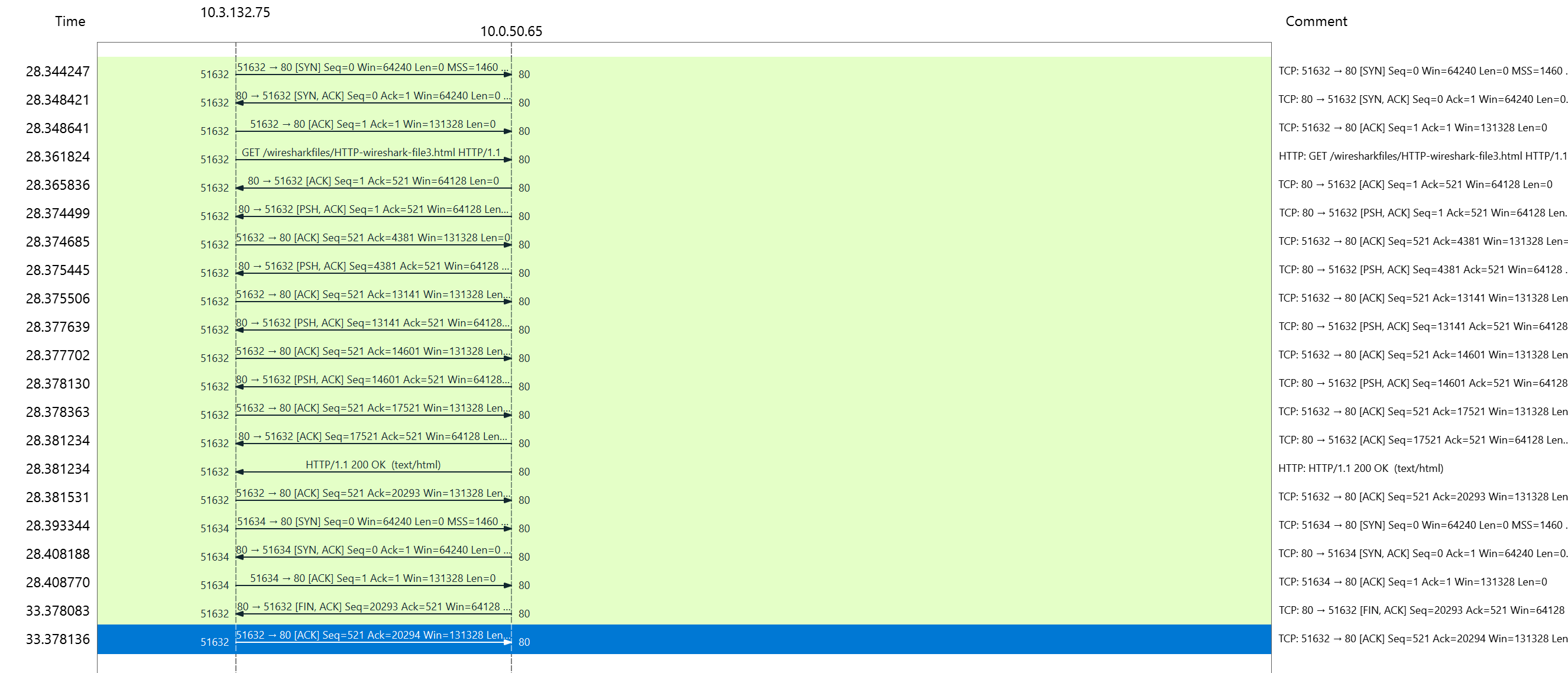
1. Нэг HTTP мессежийг дамжуулахад хэдэн TCP сегмент ашигласан байна вэ? Хэдэн байтын хэмжээтэй ямар дугаартай фремүүдэд хуваагдсан байна вэ?



*Figure 21 TCP segment*

2 TCP сегмент ашигласан байна.

Уг сервертэй дамжуулалт хийсэн процесс диаграмыг (Statistics->Flow graph) зур.



*Туршилт 4. Эмбеддэд объект татаж авах*

Эмбеддэд объект татаж авах талаар судална. Эхлээд туршилт хийхээс өмнө веб хөтчийн кэшийг цэвэрлэх шаардлагатай. Wireshark программ ажиллуулж, холбогдох интерфейсийн сүлжээний урсгалыг цуглуулж эхлээрэй. Веб хөтчийг нээж <http://netconf.num.edu.mn/wiresharkfiles/HTTP-wireshark-file4.html> хуудсыг дуудна. Wireshark програмыг зогсоож, шүүлтүүр талбарт “http” утгыг оруулах бөгөөд дараах асуултад хариулна уу. Уг хуудсанд текст болон 2 зураг агуулагдаж байгаа.

1. Веб хөтчөөс хэдэн HTTP GET хүсэлт илгээгдэж байна вэ? Эдгээр GET хүсэлтүүд ямар IP хаяг руу илгээгдэж байгааг ажиглаж, тайландаа тусгаарай.



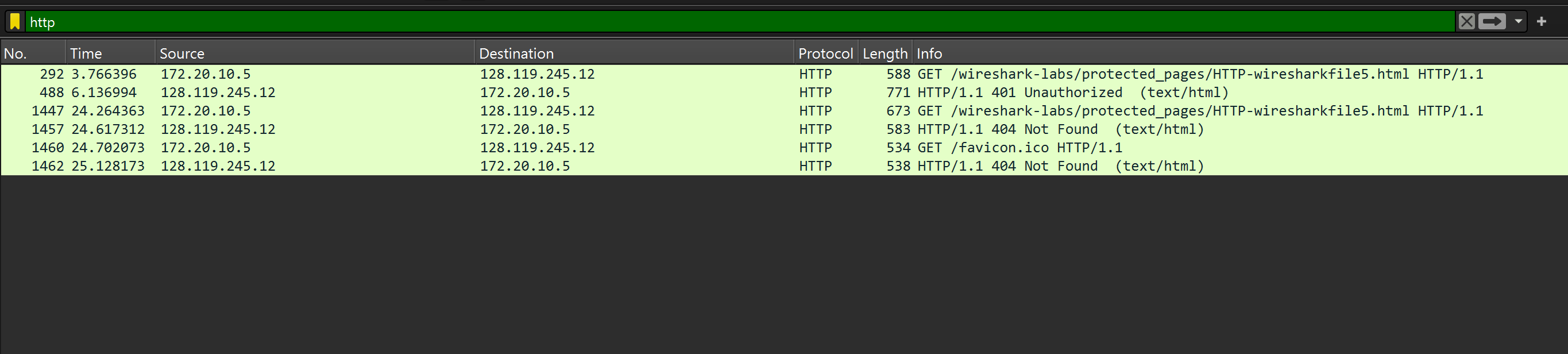
Веб хөтчөөс 3 HTTP GET хүсэлт илгээгдэж байна, эдгээр хүсэлтүүд 10.0.50.65 IP хаяг руу илгээгдэж байна. Нэг нь вэб хуудас руу нөгөө хоёр нь зураг руу GET хүсэлтийг илгээж байна.

1. Веб хөтөч уг хоёр вебээс хоёр зургийг татахдаа цуваа эсвэл зэрэгцээ аль хэлбэрээр татсан бэ? Тайлбарла.

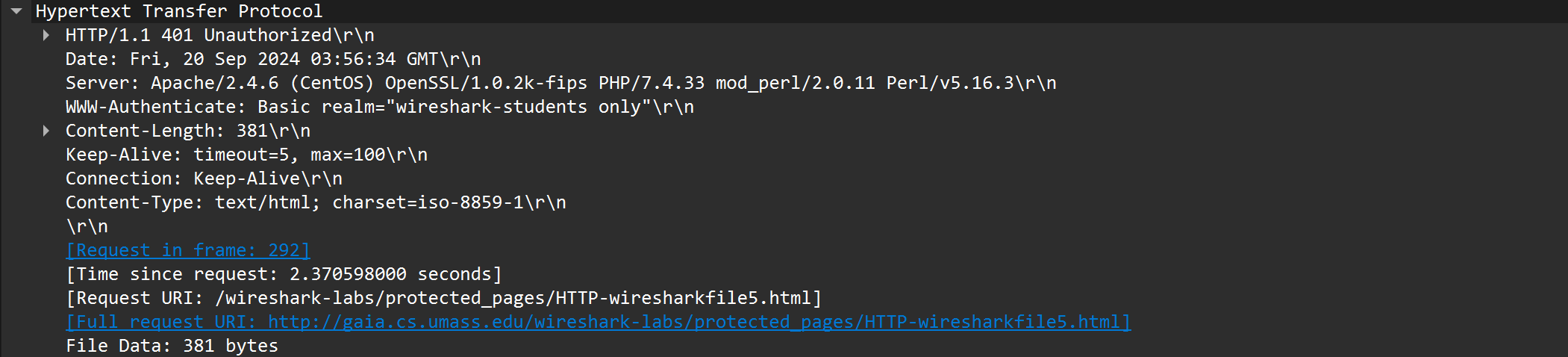
Веб хөтөч уг хоёр вебээс хоёр зургийг татахдаа зэрэгцээ хэлбэрээр татсан. Эхний хүсэлтийн хариуг хүлээхгүйгээр дараагийн хүсэлтийг илгээсэн байна.

*Туршилт 5. HTTP хэрэглэгчийн эрхээр нэвтрэх*

Веб хуудас руу хэрэглэгчийн эрхээр нэвтэрч орох үед юу болдог талаар судална. Wireshark программ ажиллуулж, холбогдох интерфейсийн сүлжээний урсгалыг цуглуулж эхлээрэй. Веб хөтчийг нээж <http://netconf.num.edu.mn/wiresharkfiles/HTTP-wireshark-> file5.html хандаж, хэрэглэгчийн нэр: “*wireshark-students*”, нууц үг: “*network*” байна. Wireshark програмыг зогсоож, шүүлтүүр талбарт “http” утгыг оруулах бөгөөд дараах асуултад хариулна уу.



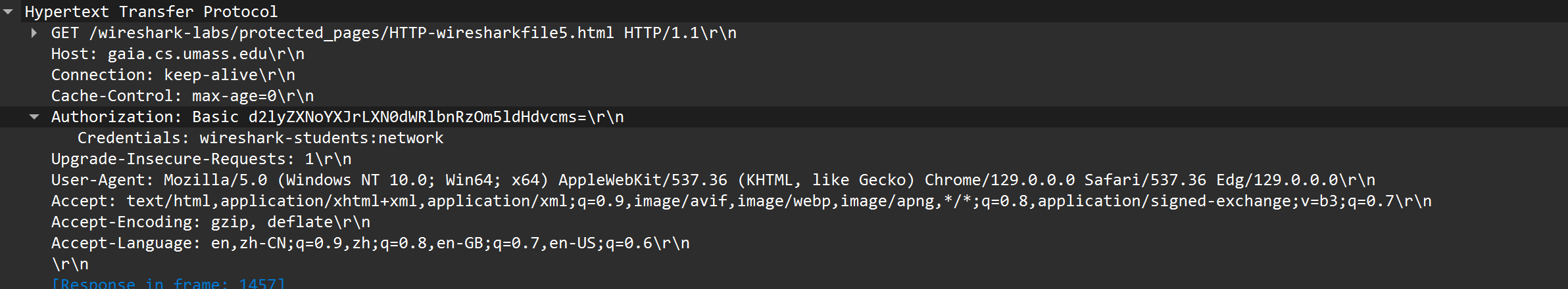
1. Эхний HTTP GET мессежний хариу мессежний төлөвийн код болон нэршил ямар байна вэ?



*Figure 22 Status code*

401 Unauthorized required нэвтрэх эрх шаардлагатай.

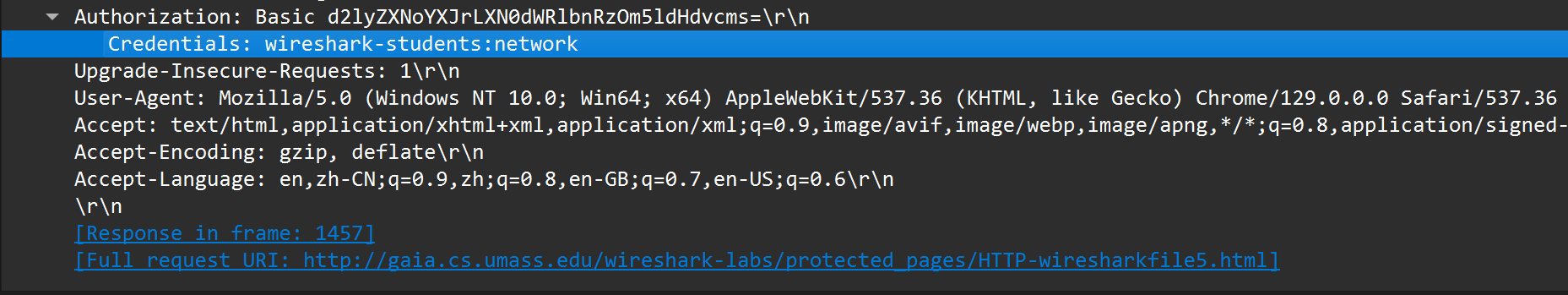
1. Хоёр дахь удаа HTTP GET мессеж илгээхэд HTTP GET мессежид ямар шинэ талбар нэмэгдсэн байна вэ?



*Figure 23 Authorization*

Энэ нь бид хүсэлт явуулахдаа сервер рүү хэрэглэгчийн нэр болон нууц үгийг илгээж, хуудас авах эрхтэй гэдгээ мэдэгдсэн тул authorization талбар нэмэгдсэн.

1. HTTP протоколоор холбогдох үед хэрэглэгчийн нэр, нууц үгийг дамжуулалтын явцад дундаас барих боломжтой байдаг. Тэгвэл эдгээр мэдээллийг ямар мессежийн аль талбараас харах боломжтой байна вэ?



*Figure 24 Credentials*

**Дүгнэлт**

Энэхүү лабораторийн хүрээнд cisco packet tracer програм ашиглан жижиг веб сервер болон клиентуудтай топологи үүсгэн сүлжээний тохиргоо хийн веб сервэрийг тестэлж үзлээ. Wireshark програм ашиглан төрөл бүрийн HTTP GET хүсэлт явуулах хэрхэн серверийн cache ажиладаг, том хэмжээний өгөгдөл хэрхэн дамждаг болон