

Kathará

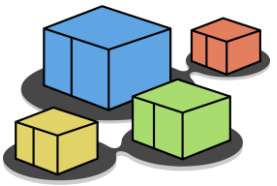
Lab webserver

web server and browser

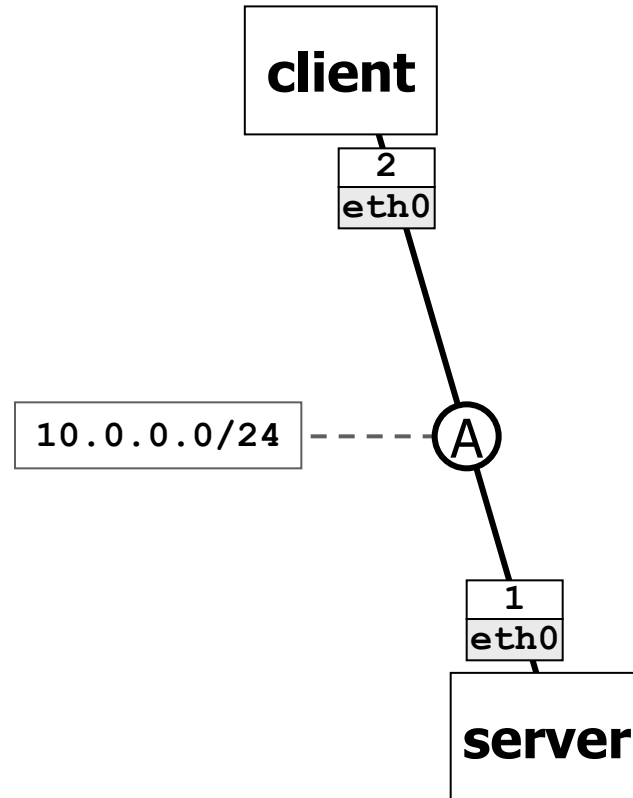
Version	1.4
Author(s)	Lorenzo Ariemma, Tommaso Caiazzzi, Giuseppe Di Battista, Maurizio Patrignani, Massimo Rimondini
E-mail	contact@kathara.org
Web	http://www.kathara.org/
Description	A lab showing the operation of a Web server accessed by a browser client – the TCP perspective – kathara version of a corresponding netkit lab vers. 1.2

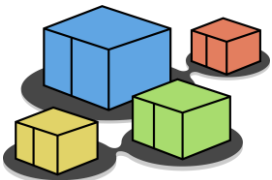
Copyright notice

- All the pages/slides in this presentation, including but not limited to, images, photos, animations, videos, sounds, music, and text (hereby referred to as “material”) are protected by copyright.
- This material, with the exception of some multimedia elements licensed by other organizations, is property of the authors and/or organizations appearing in the first slide.
- This material, or its parts, can be reproduced and used for didactical purposes within universities and schools, provided that this happens for non-profit purposes.
- Information contained in this material cannot be used within network design projects or other products of any kind.
- Any other use is prohibited, unless explicitly authorized by the authors on the basis of an explicit agreement.
- The authors assume no responsibility about this material and provide this material “as is”, with no implicit or explicit warranty about the correctness and completeness of its contents, which may be subject to changes.
- This copyright notice must always be redistributed together with the material, or its portions.



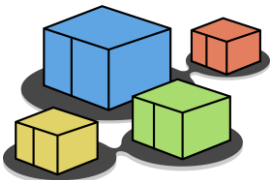
Lab topology





Lab description

- server
 - runs apache2 (with a default configuration)
- client
 - the user can launch a text-based web browser (**links**) to check the server operation



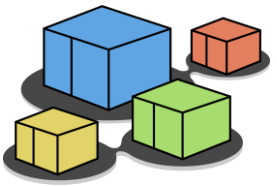
The server

- the user can check that apache2 is up and running by using the following command:

```
root@server:~$ systemctl start apache2
root@server:~$
```

- we put a test html page
 - located in `/var/www/html/index.html`

```
<html>
  <body>
    <h1>Hello!</h1>
  </body>
</html>
```



The client

- the user is supposed to start the web browser **links** on the client

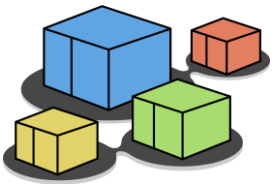
```
root@client:~$ links http://10.0.0.1
```

- you should get a screen saying “Hello!”






let us observe the packets

- perform the following command on the host computer to observe the traffic generated by the http protocol
 - `kathara lconfig -n wireshark --add A`
- what follows is a list of packets observed on the Ethernet link called A



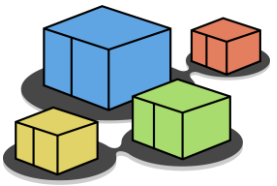
The 13 captured packets

*eth1						
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
						
Apply a display filter ... <Ctrl-/>  +						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000...	5e:61:c3:8e:91:bc	Broadcast	ARP	60	Who has 10.0.0.1? Tell 10.0.0.2[Malformed Packet]
2	0.000572...	ae:eb:54:d8:fd:ab	5e:61:c3:8e:91:bc	ARP	60	10.0.0.1 is at ae:eb:54:d8:fd:ab[Malformed Packet]
3	0.000580...	10.0.0.2	10.0.0.1	TCP	74	60208 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSv...
4	0.000588...	10.0.0.1	10.0.0.2	TCP	74	80 → 60208 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SA...
5	0.000596...	10.0.0.2	10.0.0.1	TCP	66	60208 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3035323436...
6	0.000971...	10.0.0.2	10.0.0.1	HTTP	686	GET / HTTP/1.1
7	0.000980...	10.0.0.1	10.0.0.2	TCP	66	80 → 60208 [ACK] Seq=1 Ack=621 Win=64640 Len=0 TSval=21450453...
8	0.002337...	10.0.0.1	10.0.0.2	HTTP	597	HTTP/1.1 200 OK (text/html)
9	0.002496...	10.0.0.2	10.0.0.1	TCP	66	60208 → 80 [ACK] Seq=621 Ack=532 Win=64128 Len=0 TSval=303532...
10	5.009319...	10.0.0.1	10.0.0.2	TCP	66	80 → 60208 [FIN, ACK] Seq=532 Ack=621 Win=64640 Len=0 TSval=2...
11	5.056332...	10.0.0.2	10.0.0.1	TCP	66	60208 → 80 [ACK] Seq=621 Ack=533 Win=64128 Len=0 TSval=303532...
12	16.95590...	10.0.0.2	10.0.0.1	TCP	66	60208 → 80 [FIN, ACK] Seq=621 Ack=533 Win=64128 Len=0 TSval=3...
13	16.96008...	10.0.0.1	10.0.0.2	TCP	66	80 → 60208 [ACK] Seq=533 Ack=622 Win=64640 Len=0 TSval=214506...

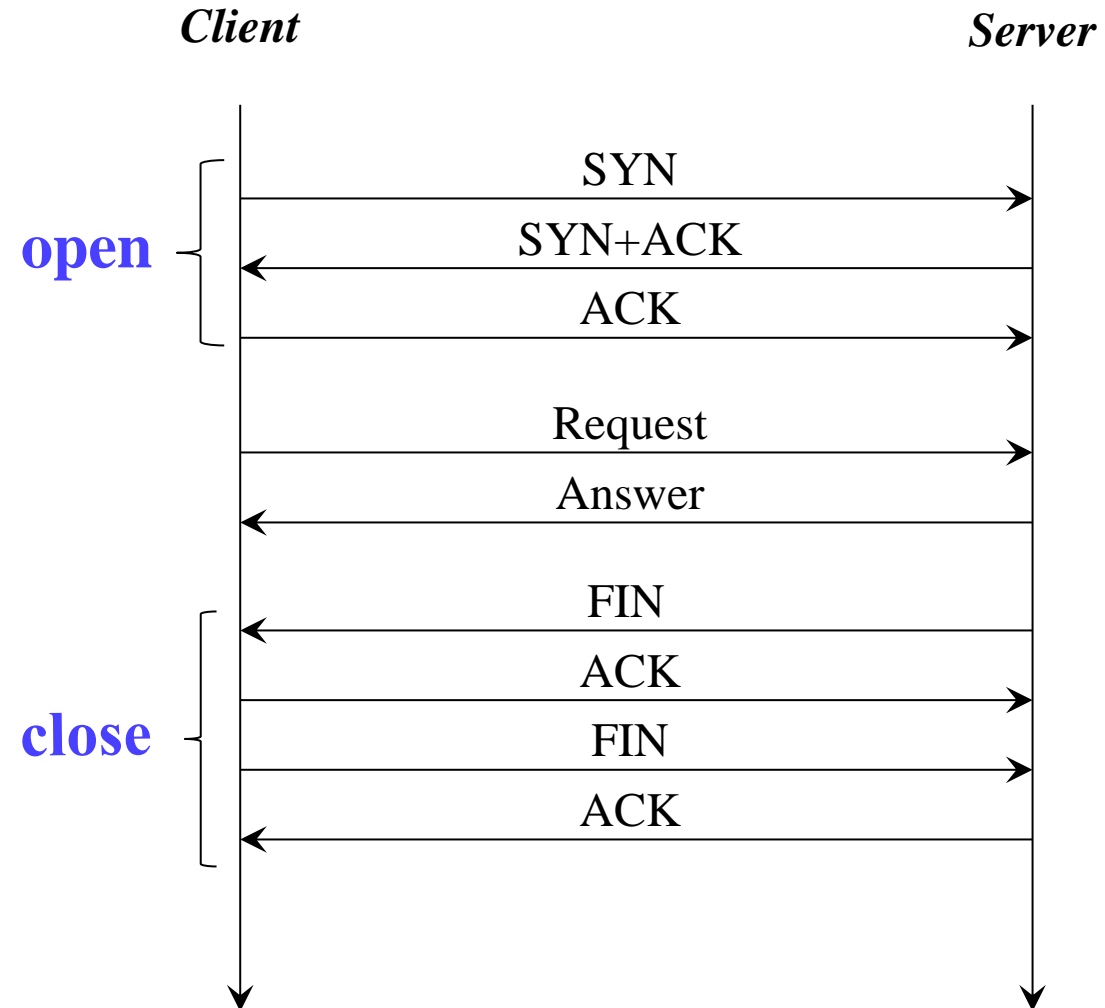
 wireshark_eth1BZ0UG2.pcapng

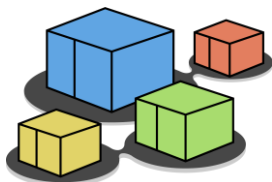
Packets: 13 · Displayed: 13 (100.0%) · Dropped: 0 (0.0%)

Profile: Default



http basic behaviour





pkt 1 – client→bcast – arp request

arp request:
the client looks for the MAC address of the server

The image shows a Wireshark packet capture window titled '*eth1'. The packet list on the left shows four packets. Packet 1 is an ARP request from 5e:61:c3:8e:91:bc to Broadcast (10.0.0.1). The packet details pane on the right shows the structure of the ARP request, including the Ethernet II header, the ARP request structure, and the hardware and protocol types. The packet bytes pane on the right shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000...	5e:61:c3:8e:91:bc	Broadcast	ARP	60	Who has 10.0.0.1? Tell 10.0.0.2[Malformed Packet]
2	0.000572...	ae:eb:54:d8:fd:ab	5e:61:c3:8e:91:bc	ARP	60	10.0.0.1 is at ae:eb:54:d8:fd:ab[Malformed Packet]
3	0.000580...	10.0.0.2	10.0.0.1	TCP	74	60208 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PER...
4	0.000588...	10.0.0.1	10.0.0.2	TCP	74	80 → 60208 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=14...

Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth1

Ethernet II, Src: 5e:61:c3:8e:91:bc (5e:61:c3:8e:91:bc), Dst: ff:ff:ff:ff:ff:ff

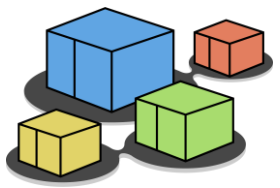
Address Resolution Protocol (request)

- Hardware type: Ethernet (1)
- Protocol type: IPv4 (0x0800)
- Hardware size: 6
- Protocol size: 4
- Opcode: request (1)
- Sender MAC address: 5e:61:c3:8e:91:bc (5e:61:c3:8e:91:bc)
- Sender IP address: 10.0.0.2
- Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
- Target IP address: 10.0.0.1

[Malformed Packet: F5 Ethernet trailer]

wireshark_eth1BZ0UG2.pcapng

Packets: 13 · Displayed: 13 (100.0%) · Dropped: 0 (0.0%) · Profile: Default



pkt 2 – client ← server – arp reply

Wireshark packet capture analysis showing an ARP reply packet (packet 2) from the server to the client.

Packet 2: 0.000572... ae:eb:54:d8:fd:ab → 5e:61:c3:8e:91:bc, ARP, 60 bytes. Info: 10.0.0.1 is at ae:eb:54:d8:fd:ab [Malformed Packet].

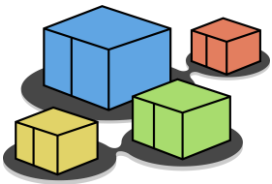
Packet 3: 0.000580... 10.0.0.2 → 10.0.0.1, TCP, 74 bytes. Info: 60208 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PER...

Packet 4: 0.000588... 10.0.0.1 → 10.0.0.2, TCP, 74 bytes. Info: 80 → 60208 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=14...

Frame 2 details: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth1. Ethernet II, Src: ae:eb:54:d8:fd:ab (ae:eb:54:d8:fd:ab), Dst: 5e:61:c3:8e:91:bc (5e:61:c3:8e:91:bc). Address Resolution Protocol (reply). Hardware type: Ethernet (1), Protocol type: IPv4 (0x0800), Hardware size: 6, Protocol size: 4, Opcode: reply (2), Sender MAC address: ae:eb:54:d8:fd:ab (ae:eb:54:d8:fd:ab), Sender IP address: 10.0.0.1, Target MAC address: 5e:61:c3:8e:91:bc (5e:61:c3:8e:91:bc), Target IP address: 10.0.0.2. [Malformed Packet: F5 Ethernet trailer]

Wireshark interface: *eth1. Packets: 13 · Displayed: 13 (100.0%) · Dropped: 0 (0.0%) · Profile: Default.

arp
reply:
the
server
provides
its MAC
address



pkt 3 – client→server – syn

the client starts the three-way-handshake

the port the client is using as source port is 60208

the port the client is "knocking on" is 80

The image shows a Wireshark packet capture window titled '*eth1'. The packet list on the left shows four packets. Packet 3 is selected, showing a TCP SYN packet from 10.0.0.2 to 10.0.0.1. The packet details on the right show the TCP header with the SYN flag set. The packet bytes pane shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000...	5e:61:c3:8e:91:bc	Broadcast	ARP	60	Who has 10.0.0.1? Tell 10.0.0.2[Malformed Packet]
2	0.000572...	ae:eb:54:d8:fd:ab	5e:61:c3:8e:91:bc	ARP	60	10.0.0.1 is at ae:eb:54:d8:fd:ab[Malformed Packet]
3	0.000580...	10.0.0.2	10.0.0.1	TCP	74	60208 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PER...
4	0.000588...	10.0.0.1	10.0.0.2	TCP	74	80 → 60208 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=14...

1010 = Header Length: 40 bytes (10)

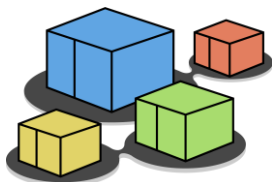
Flags: 0x002 (SYN)

- 000. = Reserved: Not set
- ...0 = Accurate ECN: Not set
- 0... = Congestion Window Reduced: Not set
-0.. = ECN-Echo: Not set
-0. = Urgent: Not set
-0 = Acknowledgment: Not set
-0 = Push: Not set
-0 = Reset: Not set
- ▶1. = Syn: Set
-0 = Fin: Not set

[TCP Flags:S.]

Window: 64240

Packets: 13 · Displayed: 13 (100.0%) · Dropped: 0 (0.0%) · Profile: Default



pkt 3 – client→server – initial seq. numb.

the client
proposes
3723005447
as its
initial
sequence
number

Wireshark capture window titled *eth1 showing packet 3 selected.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000...	5e:61:c3:8e:91:bc	Broadcast	ARP	60	Who has 10.0.0.1? Tell 10.0.0.2 [Malformed Packet]
2	0.000572...	ae:eb:54:d8:fd:ab	5e:61:c3:8e:91:bc	ARP	60	10.0.0.1 is at ae:eb:54:d8:fd:ab [Malformed Packet]
3	0.000580...	10.0.0.2	10.0.0.1	TCP	74	60208 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PER...
4	0.000588...	10.0.0.1	10.0.0.2	TCP	74	80 → 60208 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=14...

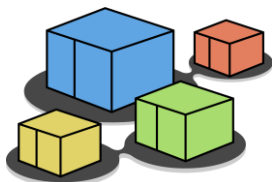
Packet 3 details:

- Source Port: 60208
- Destination Port: 80
- [Stream index: 0]
- [Conversation completeness: Complete, WITH_DATA (31)]
- [TCP Segment Len: 0]
- Sequence Number: 0 (relative sequence number)
- Sequence Number (raw): 3723005447
- [Next Sequence Number: 1 (relative sequence number)]
- Acknowledgment Number: 0
- Acknowledgment number (raw): 0
- 1010 = Header Length: 40 bytes (10)
- Flags: 0x002 (SYN)
- Window: 64240
- [Calculated window size: 64240]

Raw packet data (hex):

```
0000 ae eb 54 d8 fd ab 5e 61 c3 8e 91 bc 08 00 45 00
0010 00 3c c9 05 40 00 40 06 5d b4 0a 00 00 02 0a 00
0020 00 01 eb 30 00 50 dd e8 8e 07 00 00 00 00 a0 02
0030 fa f0 d0 84 00 00 02 04 05 b4 04 02 08 0a b4 eb
0040 5c 2b 00 00 00 00 01 03 03 07
```

Wireshark status bar: Packets: 13 · Displayed: 13 (100.0%) · Dropped: 0 (0.0%) · Profile: Default



pkt 3 – client→server – MSS option

the client
proposes
a
maximum
segment
size of
1460
bytes

Wireshark interface showing packet 3 (client to server) with the MSS option.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000...	5e:61:c3:8e:91:bc	Broadcast	ARP	60	Who has 10.0.0.1? Tell 10.0.0.2[Malformed Packet]
2	0.000572...	ae:eb:54:d8:fd:ab	5e:61:c3:8e:91:bc	ARP	60	10.0.0.1 is at ae:eb:54:d8:fd:ab[Malformed Packet]
3	0.000580...	10.0.0.2	10.0.0.1	TCP	74	60208 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PER...
4	0.000588...	10.0.0.1	10.0.0.2	TCP	74	80 → 60208 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=14...

[TCP Flags:S.]

Window: 64240

[Calculated window size: 64240]

Checksum: 0xd084 [unverified]

[Checksum Status: Unverified]

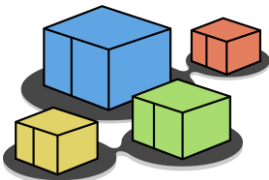
Urgent Pointer: 0

Options: (20 bytes), Maximum segment size, SACK permitted, Time

- TCP Option - Maximum segment size: 1460 bytes
- TCP Option - SACK permitted
- TCP Option - Timestamps
- TCP Option - No-Operation (NOP)
- TCP Option - Window scale: 7 (multiply by 128)
- [Timestamps]

Hex dump of packet 3 (client to server):

```
0000  ae eb 54 d8 fd ab 5e 61  c3 8e 91 bc 08 00 45 00
0010  00 3c c9 05 40 00 40 06  5d b4 0a 00 00 02 0a 00
0020  00 01 eb 30 00 50 dd e8  8e 07 00 00 00 00 a0 02
0030  fa f0 d0 84 00 00 02 04  05 b4 04 02 08 0a b4 eb
0040  5c 2b 00 00 00 00 01 03  03 07
```



pkt 4 – client ← server – syn ack

second
packet
of the
three-
way-
handsha
ke

The image shows a Wireshark packet capture window titled '*eth1'. The packet list on the left shows four packets. Packet 4 is selected, showing details for a TCP SYN-ACK. The packet bytes pane on the right displays the raw data in hexadecimal.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000...	5e:61:c3:8e:91:bc	Broadcast	ARP	60	Who has 10.0.0.1? Tell 10.0.0.2[Malformed Packet]
2	0.000572...	ae:eb:54:d8:fd:ab	5e:61:c3:8e:91:bc	ARP	60	10.0.0.1 is at ae:eb:54:d8:fd:ab[Malformed Packet]
3	0.000580...	10.0.0.2	10.0.0.1	TCP	74	60208 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PER...
4	0.000588...	10.0.0.1	10.0.0.2	TCP	74	80 → 60208 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=14...

Flags: 0x012 (SYN, ACK)

- 000. = Reserved: Not set
- ...0 = Accurate ECN: Not set
- 0... = Congestion Window Reduced: Not set
-0.. = ECN-Echo: Not set
-0. = Urgent: Not set
-1 = Acknowledgment: Set
- 0... = Push: Not set
-0.. = Reset: Not set
- ▶1. = Syn: Set
-0 = Fin: Not set

[TCP Flags:A..S.]

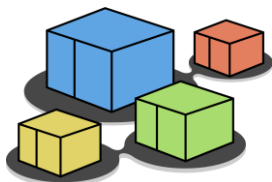
Window: 65160

[Calculated window size: 65160]

0000 5e 61 c3 8e 91 bc ae eb 54 d8 fd ab 08 00 45 00
0010 00 3c 00 00 40 00 40 06 26 ba 0a 00 00 01 0a 00
0020 00 02 00 50 eb 30 aa 03 30 ed dd e8 8e 08 a0 12
0030 fe 88 a6 cb 00 00 02 04 05 b4 04 02 08 0a 7f da
0040 cb 44 b4 eb 5c 2b 01 03 03 07

wireshark_eth1BZ0UG2.pcapng

Packets: 13 · Displayed: 13 (100.0%) · Dropped: 0 (0.0%) Profile: Default



pkt 4 – client ← server – MSS option

the server
proposes
a
maximum
segment
size of
1460
bytes too

Wireshark packet capture analysis of a TCP connection. The packet list shows four packets:

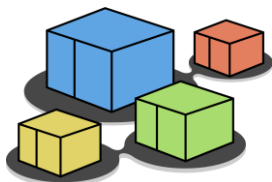
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000...	5e:61:c3:8e:91:bc	Broadcast	ARP	60	Who has 10.0.0.1? Tell 10.0.0.2 [Malformed Packet]
2	0.000572...	ae:eb:54:d8:fd:ab	5e:61:c3:8e:91:bc	ARP	60	10.0.0.1 is at ae:eb:54:d8:fd:ab [Malformed Packet]
3	0.000580...	10.0.0.2	10.0.0.1	TCP	74	60208 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PER...
4	0.000588...	10.0.0.1	10.0.0.2	TCP	74	80 → 60208 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=14...

The packet details for packet 4 show the following information:

- Window: 65160
- [Calculated window size: 65160]
- Checksum: 0xa6cb [unverified]
- [Checksum Status: Unverified]
- Urgent Pointer: 0
- Options: (20 bytes), Maximum segment size, SACK permitted, Time
- TCP Option - Maximum segment size: 1460 bytes
 - Kind: Maximum Segment Size (2)
 - Length: 4
 - MSS Value: 1460
- TCP Option - SACK permitted
- TCP Option - Timestamps
- TCP Option - No-Operation (NOP)
- TCP Option - Window scale: 7 (multiply by 128)

The packet bytes pane shows the raw data for packet 4, with the MSS option highlighted in blue:

```
0000  5e 61 c3 8e 91 bc ae eb 54 d8 fd ab 08 00 45 00
0010  00 3c 00 00 40 00 40 06 26 ba 0a 00 00 01 0a 00
0020  00 02 00 50 eb 30 aa 03 30 ed dd e8 8e 08 a0 12
0030  fe 88 a6 cb 00 00 02 04 05 b4 04 02 08 0a 7f da
0040  cb 44 b4 eb 5c 2b 01 03 03 07
```

pkt 4 – client ← server – initial seq. numb.

the server proposes 2852335853 as initial sequence number and acks the sequence number proposed by the client

Wireshark interface showing packet 4 (TCP SYN, ACK) from client to server.

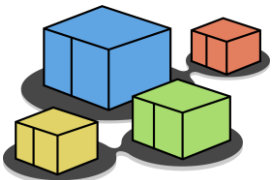
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000...	5e:61:c3:8e:91:bc	Broadcast	ARP	60	Who has 10.0.0.1? Tell 10.0.0.2 [Malformed Packet]
2	0.000572...	ae:eb:54:d8:fd:ab	5e:61:c3:8e:91:bc	ARP	60	10.0.0.1 is at ae:eb:54:d8:fd:ab [Malformed Packet]
3	0.000580...	10.0.0.2	10.0.0.1	TCP	74	60208 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PER...
4	0.000588...	10.0.0.1	10.0.0.2	TCP	74	80 → 60208 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=14...

Packet 4 details:

- Source Port: 80
- Destination Port: 60208
- [Stream index: 0]
- [Conversation completeness: Complete, WITH_DATA (31)]
- [TCP Segment Len: 0]
- Sequence Number: 0 (relative sequence number)
- Sequence Number (raw): 2852335853
- [Next Sequence Number: 1 (relative sequence number)]
- Acknowledgment Number: 1 (relative ack number)
- Acknowledgment number (raw): 3723005448
- 1010 = Header Length: 40 bytes (10)
- Flags: 0x012 (SYN, ACK)
- Window: 65160
- [Calculated window size: 65160]

Packet 4 hex dump:

```
0000  5e 61 c3 8e 91 bc ae eb 54 d8 fd ab 08 00 45 00
0010  00 3c 00 00 40 00 40 06 26 ba 0a 00 00 01 0a 00
0020  00 02 00 50 eb 30 aa 03 30 ed dd e8 8e 08 a0 12
0030  fe 88 a6 cb 00 00 02 04 05 b4 04 02 08 0a 7f da
0040  cb 44 b4 eb 5c 2b 01 03 03 07
```



pkt 5 – client→server – ack

third
packet
of the
three-
way-
handsha
ke

Wireshark capture window titled *eth1 showing a packet capture on the eth1 interface. The packet list shows four packets. Packet 5 is selected, showing its details and raw data.

No.	Time	Source	Destination	Protocol	Length	Info
3	0.000580...	10.0.0.2	10.0.0.1	TCP	74	60208 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PER...
4	0.000588...	10.0.0.1	10.0.0.2	TCP	74	80 → 60208 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=14...
5	0.000596...	10.0.0.2	10.0.0.1	TCP	66	60208 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=30353...
6	0.000971...	10.0.0.2	10.0.0.1	HTTP	686	GET / HTTP/1.1

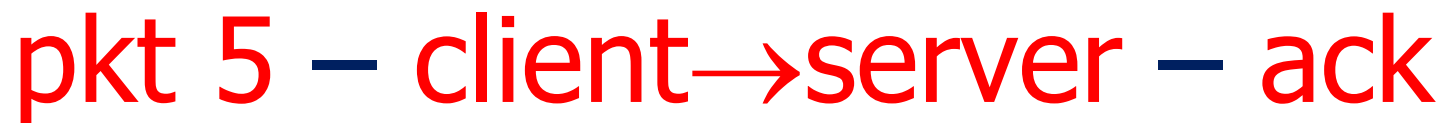
Details for Packet 5 (TCP):

- Flags: 0x010 (ACK)
- 000. = Reserved: Not set
- ...0 = Accurate ECN: Not set
- 0... = Congestion Window Reduced: Not set
-0.. = ECN-Echo: Not set
-0. = Urgent: Not set
-1 = Acknowledgment: Set
- 0... = Push: Not set
-0.. = Reset: Not set
-0. = Syn: Not set
-0 = Fin: Not set
- [TCP Flags:A.....]
- Window: 502
- [Calculated window size: 64256]

Raw data (hex):

```
0000 ae eb 54 d8 fd ab 5e 61 c3 8e 91 bc 08 00 45 00
0010 00 34 c9 06 40 00 40 06 5d bb 0a 00 00 02 0a 00
0020 00 01 eb 30 00 50 dd e8 8e 08 aa 03 30 ee 80 10
0030 01 f6 d2 29 00 00 01 01 08 0a b4 eb 5c 2c 7f da
0040 cb 44
```

Wireshark status bar: Packets: 13 · Displayed: 13 (100.0%) · Dropped: 0 (0.0%) · Profile: Default



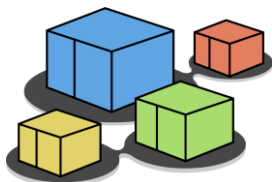
The image shows a Wireshark packet capture analysis of a TCP SYN-ACK packet. The packet list at the top shows four packets. The selected packet (No. 5) is a TCP segment from 10.0.0.1 to 10.0.0.2, sequence number 1, acknowledgment number 1, and window size 502. The packet details pane shows the TCP segment structure, including the source and destination ports, sequence and acknowledgment numbers, and window size. The packet bytes pane shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Length	Info
2	0.000572...	ae:eb:54:d8:fd:ab	5e:61:c3:8e:91:bc	ARP	60	10.0.0.1 is at ae:eb:54:d8:fd:ab[Malformed Packet]
3	0.000580...	10.0.0.2	10.0.0.1	TCP	74	60208 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PER...
4	0.000588...	10.0.0.1	10.0.0.2	TCP	74	80 → 60208 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=14...
5	0.000596...	10.0.0.2	10.0.0.1	TCP	66	60208 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=30353...

Source Port: 60208
Destination Port: 80
[Stream index: 0]
[Conversation completeness: Complete, WITH_DATA (31)]
[TCP Segment Len: 0]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 3723005448
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 2852335854
1000 = Header Length: 32 bytes (8)
Flags: 0x010 (ACK)
Window: 502
[Calculated window size: 64256]

0000 ae eb 54 d8 fd ab 5e 61 c3 8e 91 bc 08 00 45 00
0010 00 34 c9 06 40 00 40 06 5d bb 0a 00 00 02 0a 00
0020 00 01 eb 30 00 50 dd e8 8e 08 aa 03 30 ee 80 10
0030 01 f6 d2 29 00 00 01 01 08 0a b4 eb 5c 2c 7f da
0040 cb 44

Packets: 13 · Displayed: 13 (100.0%) · Dropped: 0 (0.0%) Profile: Default



pkt 6 – client→server – http GET

http
GET
with
http
version
1.1

Wireshark packet capture window titled *eth1. The packet list shows four packets. Packet 6 is selected, showing an HTTP GET request from 10.0.0.2 to 10.0.0.1.

No.	Time	Source	Destination	Protocol	Length	Info
3	0.000580...	10.0.0.2	10.0.0.1	TCP	74	60208 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PER...
4	0.000588...	10.0.0.1	10.0.0.2	TCP	74	80 → 60208 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=14...
5	0.000596...	10.0.0.2	10.0.0.1	TCP	66	60208 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=30353...
6	0.000971...	10.0.0.2	10.0.0.1	HTTP	686	GET / HTTP/1.1

Packet 6 details: Hypertext Transfer Protocol, GET / HTTP/1.1\r\n

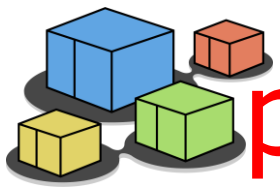
[Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]

Request Method: GET
Request URI: /
Request Version: HTTP/1.1
Host: 10.0.0.1\r\nUser-Agent: Links (2.28; Linux 5.10.102.1-microsoft-standard-WS
Accept: */*\r\nAccept-Language: en,*;q=0.1\r\nAccept-Encoding: gzip, deflate, br, zstd, bzip2, lzma, lzma2, 1
[truncated]Accept-Charset: us-ascii,ISO-8859-1,ISO-8859-2,ISO-
Connection: keep-alive\r\n\r\n

Packet bytes (hex): 0040 cb 44 47 45 54 20 2f 20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 31 30 2e 30 2e 30 2e 31 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4c 69 6e 6b 73 20 28 32 2e 32 38 3b 20 4c 69 6e 75 78 20 35 2e 31 30 2e 31 30 32 2e 31 2d 6d 69 63 72 6f 73 6f 66 74 2d 73 74 61 6e 64 61 72 64 2d 57 53 4c 32 20 78 38 36 5f 36 34 3b 20 47 4e 55 20 43 20 31 32 2e 32 3b 20 74 65 78 74 29 0d 0a 41 63 63 65 70 74 2d 4c 61 6e 67 75 61 67 65 3a 20 65 6e 2c 2a 3b 71 3d 30 2e 31 0d 0a 41 63 63 65 70 74 2d 45 6e 63 6f 64 69 6e 67 3a 20 67 7a 69 70 2c 20 64 65 66 6c 61 74 65 2c 20 62 72 2c 20 7a 73 74 64 2c 20 62 7a 69 70 32 2c 20 6c 7a 6d 61 2c 20

Text item (text), 16 byte(s)

Packets: 13 · Displayed: 13 (100.0%) · Dropped: 0 (0.0%) Profile: Default



pkt 7 – client ← server – bytes received

tcp acks
the
receipt
of the
bytes of
the GET

Wireshark capture window titled *eth1 showing network traffic analysis.

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
6	0.000971...	10.0.0.2	10.0.0.1	HTTP	686	GET / HTTP/1.1
7	0.000980...	10.0.0.1	10.0.0.2	TCP	66	80 → 60208 [ACK] Seq=1 Ack=621 Win=64640 Len=0 TSval=214...
8	0.002337...	10.0.0.1	10.0.0.2	HTTP	597	HTTP/1.1 200 OK (text/html)
9	0.002496...	10.0.0.2	10.0.0.1	TCP	66	60208 → 80 [ACK] Seq=621 Ack=532 Win=64128 Len=0 TSval=3...

Frame 7: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface eth1

Ethernet II, Src: ae:eb:54:d8:fd:ab (ae:eb:54:d8:fd:ab), Dst: 5e:6b:43:9d:8a:00

Internet Protocol Version 4, Src: 10.0.0.1, Dst: 10.0.0.2

Transmission Control Protocol, Src Port: 80, Dst Port: 60208, Seq: 1, Len: 0

Source Port: 80

Destination Port: 60208

[Stream index: 0]

[Conversation completeness: Complete, WITH_DATA (31)]

[TCP Segment Len: 0]

Sequence Number: 1 (relative sequence number)

Sequence Number (raw): 2852335854

[Next Sequence Number: 1 (relative sequence number)]

Acknowledgment Number: 621 (relative ack number)

Acknowledgment number (raw): 3723006068

0000 5e 61 c3 8e 91 bc ae eb 54 d8 fd ab 08 00 45 00

0010 00 34 ef 55 40 00 40 06 37 6c 0a 00 00 01 0a 00

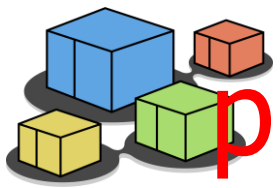
0020 00 02 00 50 eb 30 aa 03 30 ee dd e8 90 74 80 10

0030 01 f9 cf b9 00 00 01 01 08 0a 7f da cb 45 b4 eb

0040 5c 2c

wireshark_eth1BZ0UG2.pcapng

Packets: 13 · Displayed: 13 (100.0%) · Dropped: 0 (0.0%) Profile: Default



pkt 8 – client ← server – resource moves

the
requested
resource

Wireshark capture window titled *eth1 showing network traffic analysis.

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
6	0.000971...	10.0.0.2	10.0.0.1	HTTP	686	GET / HTTP/1.1
7	0.000980...	10.0.0.1	10.0.0.2	TCP	66	80 → 60208 [ACK] Seq=1 Ack=621 Win=64640 Len=0 TSval=214...
8	0.002337...	10.0.0.1	10.0.0.2	HTTP	597	HTTP/1.1 200 OK (text/html)
9	0.002496...	10.0.0.2	10.0.0.1	TCP	66	60208 → 80 [ACK] Seq=621 Ack=532 Win=64128 Len=0 TSval=3...

Frame 8: 597 bytes on wire (4776 bits), 597 bytes captured (4776 bits) on interface eth1

Ethernet II, Src: ae:eb:54:d8:fd:ab (ae:eb:54:d8:fd:ab), Dst: 5e:eb:54:d8:fd:ab (5e:eb:54:d8:fd:ab)

Internet Protocol Version 4, Src: 10.0.0.1, Dst: 10.0.0.2

Transmission Control Protocol, Src Port: 80, Dst Port: 60208, Seq: 1, Len: 0

Hypertext Transfer Protocol

HTTP/1.1 200 OK\r\n

Date: Thu, 04 Jan 2024 17:07:29 GMT\r\n

Server: Apache/2.4.57 (Debian)\r\n

Last-Modified: Thu, 04 Jan 2024 17:02:45 GMT\r\n

ETag: "228-60e21b34ad740-gzip"\r\n

Accept-Ranges: bytes\r\n

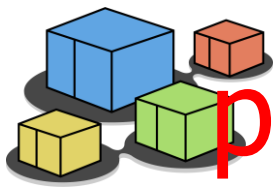
Vary: Accept-Encoding\r\n

Content-Encoding: gzip\r\n

Content-Length: 194\r\n

Frame (597 bytes) Uncompressed entity body (552 bytes)

Packets: 13 · Displayed: 13 (100.0%) · Dropped: 0 (0.0%) Profile: Default



pkt 8 – client ← server – resource moves

Wireshark packet capture analysis showing a successful HTTP GET request and response.

Packet List:

No.	Time	Source	Destination	Protocol	Length	Info
6	0.000971...	10.0.0.2	10.0.0.1	HTTP	686	GET / HTTP/1.1
7	0.000980...	10.0.0.1	10.0.0.2	TCP	66	80 → 60208 [ACK] Seq=1 Ack=621 Win=64640 Len=0 TSval=214...
8	0.002337...	10.0.0.1	10.0.0.2	HTTP	597	HTTP/1.1 200 OK (text/html)
9	0.002496...	10.0.0.2	10.0.0.1	TCP	66	60208 → 80 [ACK] Seq=621 Ack=532 Win=64128 Len=0 TSval=3...

Packet 8 Details:

- Line-based text data: text/html (30 lines)
- Content:

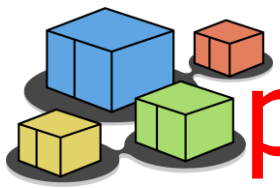
```
<html>\n<body>\n\nHello!\n\n<pre>\n\n      _\n     /\n    | o |\n    \  /\n      ) (\n     |\n    _\n   _\n  _\n _\n
```

Packet 8 Hex Data (Selected):

Offset	Hex
0040	5c 2c 48 54 54 50 2f 31 2e 31 20 32 30 30 20 4f
0050	4b 0d 0a 44 61 74 65 3a 20 54 68 75 2c 20 30 34
0060	20 4a 61 6e 20 32 30 32 34 20 31 37 3a 30 37 3a
0070	32 39 20 47 4d 54 0d 0a 53 65 72 76 65 72 3a 20
0080	41 70 61 63 68 65 2f 32 2e 34 2e 35 37 20 28 44
0090	65 62 69 61 6e 29 0d 0a 4c 61 73 74 2d 4d 6f 64
00a0	69 66 69 65 64 3a 20 54 68 75 2c 20 30 34 20 4a
00b0	61 6e 20 32 30 32 34 20 31 37 3a 30 32 3a 34 35
00c0	20 47 4d 54 0d 0a 45 54 61 67 3a 20 22 32 32 38
00d0	2d 36 30 65 32 31 62 33 34 61 64 37 34 30 2d 67
00e0	7a 69 70 22 0d 0a 41 63 63 65 70 74 2d 52 61 6e
00f0	67 65 73 3a 20 62 79 74 65 73 0d 0a 56 61 72 79
0100	3a 20 41 63 63 65 70 74 2d 45 6e 63 6f 64 69 6e

Summary: The capture shows a client (10.0.0.2) sending a GET request to a server (10.0.0.1). The server responds with a 200 OK status and an HTML document. The HTML content includes a body with a pre-formatted text block.

the requested resource



pkt 9 – client→server – bytes received

tcp acks
the bytes
of the
resource

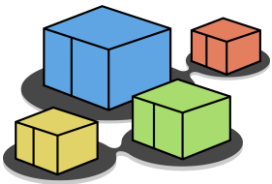
Wireshark interface showing packet 9 selected. The packet list shows a TCP ACK from 10.0.0.2 to 10.0.0.1. The packet details show the TCP segment with sequence number 621 and acknowledgment number 532. The packet bytes are displayed in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
6	0.000971...	10.0.0.2	10.0.0.1	HTTP	686	GET / HTTP/1.1
7	0.000980...	10.0.0.1	10.0.0.2	TCP	66	80 → 60208 [ACK] Seq=1 Ack=621 Win=64640 Len=0 TSval=214...
8	0.002337...	10.0.0.1	10.0.0.2	HTTP	597	HTTP/1.1 200 OK (text/html)
9	0.002496...	10.0.0.2	10.0.0.1	TCP	66	60208 → 80 [ACK] Seq=621 Ack=532 Win=64128 Len=0 TSval=3...

Frame 9: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface eth1
Ethernet II, Src: 5e:61:c3:8e:91:bc (5e:61:c3:8e:91:bc), Dst: ae:50:00:00:00:00
Internet Protocol Version 4, Src: 10.0.0.2, Dst: 10.0.0.1
Transmission Control Protocol, Src Port: 60208, Dst Port: 80, Seq: 621, Len: 0
[Stream index: 0]
[Conversation completeness: Complete, WITH_DATA (31)]
[TCP Segment Len: 0]
Sequence Number: 621 (relative sequence number)
Sequence Number (raw): 3723006068
[Next Sequence Number: 621 (relative sequence number)]
Acknowledgment Number: 532 (relative ack number)
Acknowledgment number (raw): 2852336385

0000 ae eb 54 d8 fd ab 5e 61 c3 8e 91 bc 08 00 45 00
0010 00 34 c9 08 40 00 40 06 5d b9 0a 00 00 02 0a 00
0020 00 01 eb 30 00 50 dd e8 90 74 aa 03 33 01 80 10
0030 01 f5 cd a8 00 00 01 01 08 0a b4 eb 5c 2d 7f da
0040 cb 46

wireshark_eth1BZ0UG2.pcapng Packets: 13 · Displayed: 13 (100.0%) · Dropped: 0 (0.0%) Profile: Default



pkt 10 – client ← server – fin

request
to finish

Wireshark capture window showing network traffic on interface *eth1. The packet list displays four packets, with packet 10 highlighted in blue. The packet details pane shows the structure of the selected packet, including the TCP header and flags.

No.	Time	Source	Destination	Protocol	Length	Info
10	5.009319...	10.0.0.1	10.0.0.2	TCP	66	80 → 60208 [FIN, ACK] Seq=532 Ack=621 Win=64640 Len=0 TS...
11	5.056332...	10.0.0.2	10.0.0.1	TCP	66	60208 → 80 [ACK] Seq=621 Ack=533 Win=64128 Len=0 TSval=3...
12	16.95590...	10.0.0.2	10.0.0.1	TCP	66	60208 → 80 [FIN, ACK] Seq=621 Ack=533 Win=64128 Len=0 TS...
13	16.96008...	10.0.0.1	10.0.0.2	TCP	66	80 → 60208 [ACK] Seq=533 Ack=622 Win=64640 Len=0 TSval=2...

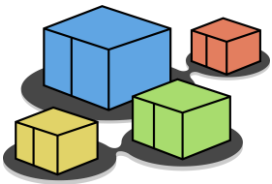
Packet 10 details:

- Acknowledgment number (raw): 3723006068
- 1000 = Header Length: 32 bytes (8)
- Flags: 0x011 (FIN, ACK)
 - 000. = Reserved: Not set
 - ...0 = Accurate ECN: Not set
 - 0... = Congestion Window Reduced: Not set
 -0.. = ECN-Echo: Not set
 -0. = Urgent: Not set
 -1 = Acknowledgment: Set
 -0... = Push: Not set
 -0.. = Reset: Not set
 -0. = Syn: Not set
 -1 = Fin: Set
- [TCP Flags:A...F]

Raw packet data (hex):

```
0000 5e 61 c3 8e 91 bc ae eb 54 d8 fd ab 08 00 45 00
0010 00 34 ef 57 40 00 40 06 37 6a 0a 00 00 01 0a 00
0020 00 02 00 50 eb 30 aa 03 33 01 dd e8 90 74 80 11
0030 01 f9 ba 15 00 00 01 01 08 0a 7f da de d4 b4 eb
0040 5c 2d
```

Wireshark status bar: Packets: 13 · Displayed: 13 (100.0%) · Dropped: 0 (0.0%) · Profile: Default



pkt 11 – client→server – ack

ack to
finish

Wireshark packet capture window showing packet 11 selected. The packet is a TCP ACK from 10.0.0.2 to 10.0.0.1.

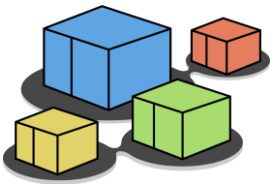
No.	Time	Source	Destination	Protocol	Length	Info
10	5.009319...	10.0.0.1	10.0.0.2	TCP	66	80 → 60208 [FIN, ACK] Seq=532 Ack=621 Win=64640 Len=0 TS...
11	5.056332...	10.0.0.2	10.0.0.1	TCP	66	60208 → 80 [ACK] Seq=621 Ack=533 Win=64128 Len=0 TSval=3...
12	16.95590...	10.0.0.2	10.0.0.1	TCP	66	60208 → 80 [FIN, ACK] Seq=621 Ack=533 Win=64128 Len=0 TS...
13	16.96008...	10.0.0.1	10.0.0.2	TCP	66	80 → 60208 [ACK] Seq=533 Ack=622 Win=64640 Len=0 TSval=2...

Packet 11 details:

- Acknowledgment number (raw): 2852336386
- 1000 = Header Length: 32 bytes (8)
- Flags: 0x010 (ACK)
 - 000. = Reserved: Not set
 - ...0 = Accurate ECN: Not set
 - 0... = Congestion Window Reduced: Not set
 -0.. = ECN-Echo: Not set
 -0. = Urgent: Not set
 -1 = Acknowledgment: Set
 - 0... = Push: Not set
 -0.. = Reset: Not set
 -0. = Syn: Not set
 -0 = Fin: Not set

Packet 11 hex dump:

```
0000  ae eb 54 d8 fd ab 5e 61  c3 8e 91 bc 08 00 45 00
0010  00 34 c9 09 40 00 40 06  5d b8 0a 00 00 02 0a 00
0020  00 01 eb 30 00 50 dd e8  90 74 aa 03 33 02 80 10
0030  01 f5 a6 5b 00 00 01 01  08 0a b4 eb 6f eb 7f da
0040  de d4
```



pkt 12 – client→server – fin

request
to finish

Wireshark packet capture window showing packet 12 (client→server) with details of the TCP FIN, ACK flag.

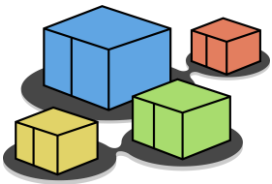
No.	Time	Source	Destination	Protocol	Length	Info
10	5.009319...	10.0.0.1	10.0.0.2	TCP	66	80 → 60208 [FIN, ACK] Seq=532 Ack=621 Win=64640 Len=0 TS...
11	5.056332...	10.0.0.2	10.0.0.1	TCP	66	60208 → 80 [ACK] Seq=621 Ack=533 Win=64128 Len=0 TSval=3...
12	16.95590...	10.0.0.2	10.0.0.1	TCP	66	60208 → 80 [FIN, ACK] Seq=621 Ack=533 Win=64128 Len=0 TS...
13	16.96008...	10.0.0.1	10.0.0.2	TCP	66	80 → 60208 [ACK] Seq=533 Ack=622 Win=64640 Len=0 TSval=2...

Packet 12 details:

- Acknowledgment number (raw): 2852336386
- 1000 = Header Length: 32 bytes (8)
- Flags: 0x011 (FIN, ACK)
 - 000. = Reserved: Not set
 - ...0 = Accurate ECN: Not set
 - 0... = Congestion Window Reduced: Not set
 -0.. = ECN-Echo: Not set
 -0. = Urgent: Not set
 -1 = Acknowledgment: Set
 - 0... = Push: Not set
 -0.. = Reset: Not set
 -0. = Syn: Not set
 -1 = Fin: Set
- [TCP Flags:A...F]

Packet 12 raw data (hex):

```
0000 ae eb 54 d8 fd ab 5e 61 c3 8e 91 bc 08 00 45 00
0010 00 34 c9 0a 40 00 40 06 5d b7 0a 00 00 02 0a 00
0020 00 01 eb 30 00 50 dd e8 90 74 aa 03 33 02 80 11
0030 01 f5 77 e1 00 00 01 01 08 0a b4 eb 9e 64 7f da
0040 de d4
```



pkt 13 – client ← server – ack

ack to
finish

Wireshark capture window showing network traffic on interface *eth1. The packet list displays four packets, with packet 13 highlighted in blue, indicating it is the selected packet.

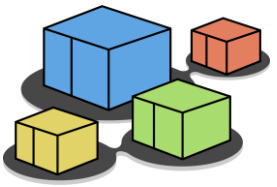
No.	Time	Source	Destination	Protocol	Length	Info
10	5.009319...	10.0.0.1	10.0.0.2	TCP	66	80 → 60208 [FIN, ACK] Seq=532 Ack=621 Win=64640 Len=0 TS...
11	5.056332...	10.0.0.2	10.0.0.1	TCP	66	60208 → 80 [ACK] Seq=621 Ack=533 Win=64128 Len=0 TSval=3...
12	16.95590...	10.0.0.2	10.0.0.1	TCP	66	60208 → 80 [FIN, ACK] Seq=621 Ack=533 Win=64128 Len=0 TS...
13	16.96008...	10.0.0.1	10.0.0.2	TCP	66	80 → 60208 [ACK] Seq=533 Ack=622 Win=64640 Len=0 TSval=2...

The packet details pane for packet 13 shows the following information:

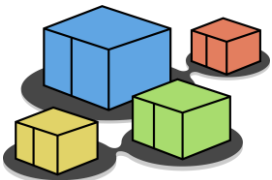
- Acknowledgment number (raw): 3723006069
- 1000 = Header Length: 32 bytes (8)
- Flags: 0x010 (ACK)
 - 000. = Reserved: Not set
 - ...0 = Accurate ECN: Not set
 - 0... = Congestion Window Reduced: Not set
 -0.. = ECN-Echo: Not set
 -0. = Urgent: Not set
 -1 = Acknowledgment: Set
 -0... = Push: Not set
 -0.. = Reset: Not set
 -0. = Syn: Not set
 -0 = Fin: Not set
- [TCP Flags:A.....]

The packet bytes pane shows the raw data of the packet, starting with 0000 5e 61 c3 8e 91 bc ae eb 54 d8 fd ab 08 00 45 00.

Wireshark status bar: Packets: 13 · Displayed: 13 (100.0%) · Dropped: 0 (0.0%) · Profile: Default



extras



The server (again)

- to monitor accesses to the web server you can use the following command (on the server):

```
root@server:~$ tail -f /var/log/apache2/access.log
10.0.0.2 - - [19/Oct/2011:08:04:08 +0000] "GET / HTTP/1.1" 200 56
 "-" "Links (2.2; Linux; 80x39)"
```

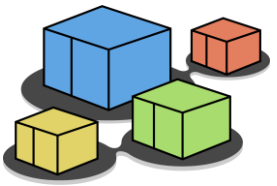


The server (again)

- to monitor errors on the web server you can use the following command (on the server):

```
root@server:~$ tail -f /var/log/apache2/error.log
[Wed Nov 14 15:57:58 2019] [notice] Apache/2.2.9 (Debian)
configured -- resuming normal operations
[Wed Nov 14 16:14:07 2019] [notice] caught SIGTERM, shutting down
```

- very useful when debugging configurations



Apache modules

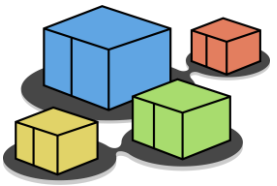
- most of apache's functionalities are built-in
 - retrieve the list using `apache2 -l`
- others can be added by enabling modules
 - to enable a module:

```
root@server:~$ a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
    service apache2 restart
root@server:~$
```



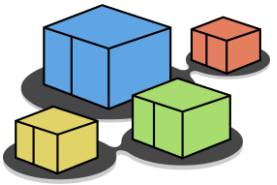

apache modules

- available modules are located in:
 - `/etc/apache2/mods-available`
- enabled modules are located in:
 - `/etc/apache2/mods-enabled`
- `a2enmod` puts a symbolic link from the relevant file(s) in:
 - `/etc/apache2/mods-available` to `/etc/apache2/mods-enabled`
- `a2dismod` removes these symbolic links



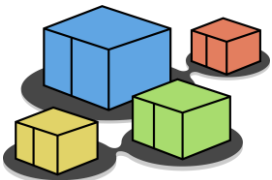
some useful apache modules

<code>userdir</code>	enables per-user web sites (this feature does not work with Kathará)
<code>rewrite</code>	implements URL rewriting
<code>proxy</code>	implements a proxy/gateway
<code>cgi/cgid</code>	supports execution of CGI scripts



per-directory configuration

- apache allows configuration changes on a per-directory basis
- creating a special file `/some/path/.htaccess` with apache configuration statements applies those statements to all files and subdirectories inside `/some/path`
 - `.htaccess` files can be nested in a directory tree
 - nested files override their parents



per-directory configuration

■ sample configuration statements:

■ restrict access from specific hosts

`Deny from example.org test.com 10.0.0 192.168.0.0/24`

■ perform URL rewriting

■ (transparently) redirect to other sites

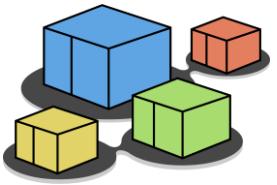
■ restrict access to a specific subdirectory

■ change name of file containing the default page

`DirectoryIndex pippo.html`

■ enable/disable directory indexing

`Options -Indexes`



Exercise: per-directory configuration

- when a resource name is not specified in the URL, apache serves `index.html` from the requested path
- hands-on:
 - edit file `/var/www/html/.htaccess` and add the following directive:
`DirectoryIndex custom_file.html`
 - rename previously created file `/var/www/html/index.html` to `custom_file.html`
 - try accessing `http://10.0.0.1/` from client
 - rename `custom_file.html` back to `index.html` and try accessing the page again