

Wireshark-II

Contents

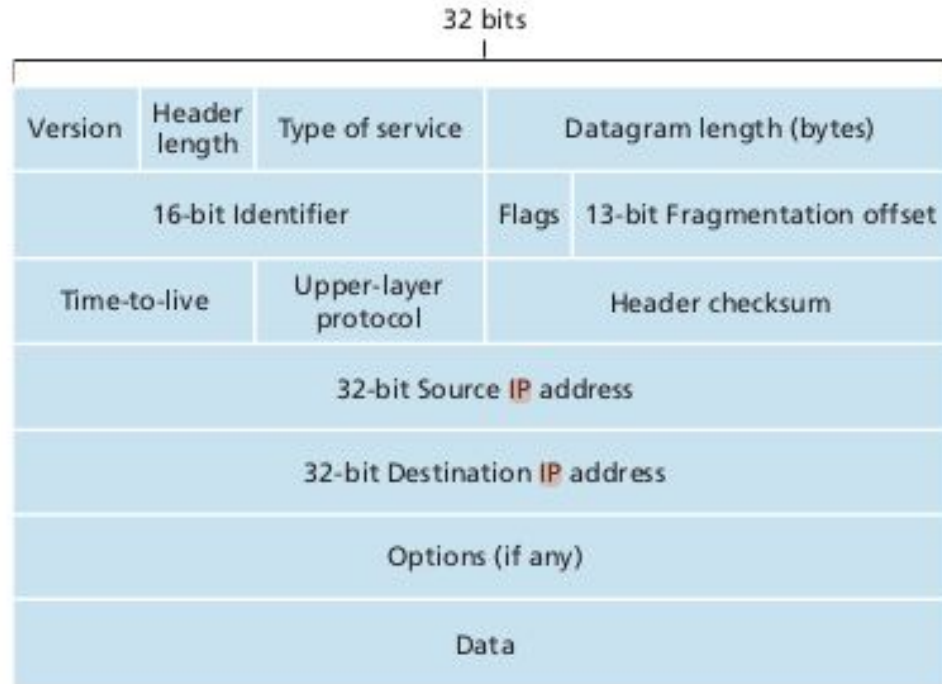
- IP
- TCP
- DNS

TCP/IP Analysis

Introduction

- TCP/IP is a family of communication protocols used to connect computer systems in a network.
- It is named after two of the protocols in the family: Transmission Control Protocol (TCP) and Internet Protocol (IP).
- **Internet Protocol (IP)**
- Each server or client on a TCP/IP internet is identified by a numeric IP (Internet Protocol) address. The two types of IP address are the IPv4 (IP version 4) address and the IPv6 (IP version 6) address.
- TCP is a transport-layer protocol that provides a reliable, full duplex, connection-oriented data transmission service. Most Internet applications use TCP.

IPv4 datagram format



- Version number-These 4 bits specify the IP protocol version of the datagram.
- Header length-typical IP datagram has a 20-byte header.
- Type of service-The type of service (TOS) bits were included in the IPv4 header to allow different types of IP datagrams to be distinguished from each other.
- Datagram length-This is the total length of the IP datagram (header plus data), measured in bytes.
- Identifier, flags, fragmentation offset. These three fields have to do with so-called IP fragmentation.
- Time-to-live - The time-to-live (TTL) field is included to ensure that datagrams do not circulate forever in the network.
- Protocol-This field is used only when an IP datagram reaches its final destination. The value of this field indicates the specific transport-layer protocol to which the data portion of this IP datagram should be passed.
- Header checksum. The header checksum aids a router in detecting bit errors in a received IP datagram.
- Source and destination IP addresses.
- Options-The options fields allow an IP header to be extended.
- Data (payload)- the data field of the IP datagram contains the transport-layer segment (TCP or UDP) to be delivered to the destination. However, the data field can carry other types of data, such as ICMP messages



ip

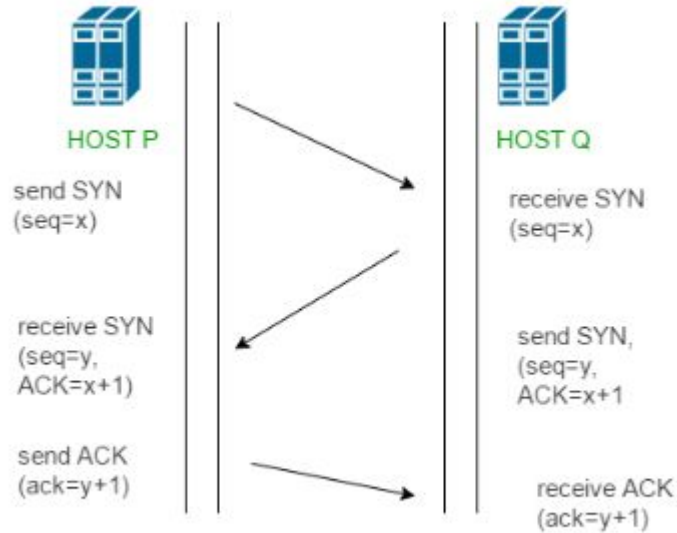
No.	Time	Source	Destination	Protocol	Length	Info
	6456.951.814053679	192.168.220.146	142.250.205.238	UDP	1292	55388 → 443 Len=1250

- Frame 33: 75 bytes on wire (600 bits), 75 bytes captured (600 bits) on interface wlp1s0, id 0
- Ethernet II, Src: IntelCor_31:2e:be (34:f6:4b:31:2e:be), Dst: 22:59:8e:56:64:7b (22:59:8e:56:64:7b)
- Internet Protocol Version 4, Src: 192.168.220.146, Dst: 142.250.205.238
 - 0100 = Version: 4
 - 0101 = Header Length: 20 bytes (5)
 - Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 - Total Length: 61
 - Identification: 0x3013 (12307)
 - Flags: 0x40, Don't fragment
 - ...0 0000 0000 0000 = Fragment Offset: 0
 - Time to Live: 64
 - Protocol: UDP (17)
 - Header Checksum: 0x1079 [validation disabled]
 - [Header checksum status: Unverified]
 - Source Address: 192.168.220.146
 - Destination Address: 142.250.205.238
- User Datagram Protocol, Src Port: 55388, Dst Port: 443
 - Source Port: 55388
 - Destination Port: 443
 - Length: 41
 - Checksum: 0xe022 [unverified]
 - [Checksum Status: Unverified]
 - [Stream index: 1]
 - [Timestamps]
 - UDP payload (33 bytes)
- Data (33 bytes)
 - Data: 4cf9d543d487b0d527d18b4640f35b834b7a0f6a206b3b0b2d264ce5b8247b88f0
 - [Length: 33]

The TCP Connection

- TCP is said to be connection-oriented because before one application process can begin to send data to another, the two processes must first “handshake” with each other
- The client first sends a special TCP segment; the server responds with a second special TCP segment; and finally the client responds again with a third special segment.
- The first two segments carry no payload, that is, no application-layer data; the third of these segments may carry a payload.
- Because three segments are sent between the two hosts, this connection-establishment procedure is often referred to as a three-way handshake.
- full-duplex service
- Point-to-point

3-way handshaking




- **Step 1 (SYN):** In the first step, the client wants to establish a connection with a server, so it sends a segment with SYN(Synchronize Sequence Number) which informs the server that the client is likely to start communication and with what sequence number it starts segments with
- **Step 2 (SYN + ACK):** Server responds to the client request with SYN-ACK signal bits set. Acknowledgement(ACK) signifies the response of the segment it received and SYN signifies with what sequence number it is likely to start the segments with
- **Step 3 (ACK):** In the final part client acknowledges the response of the server and they both establish a reliable connection with which they will start the actual data transfer

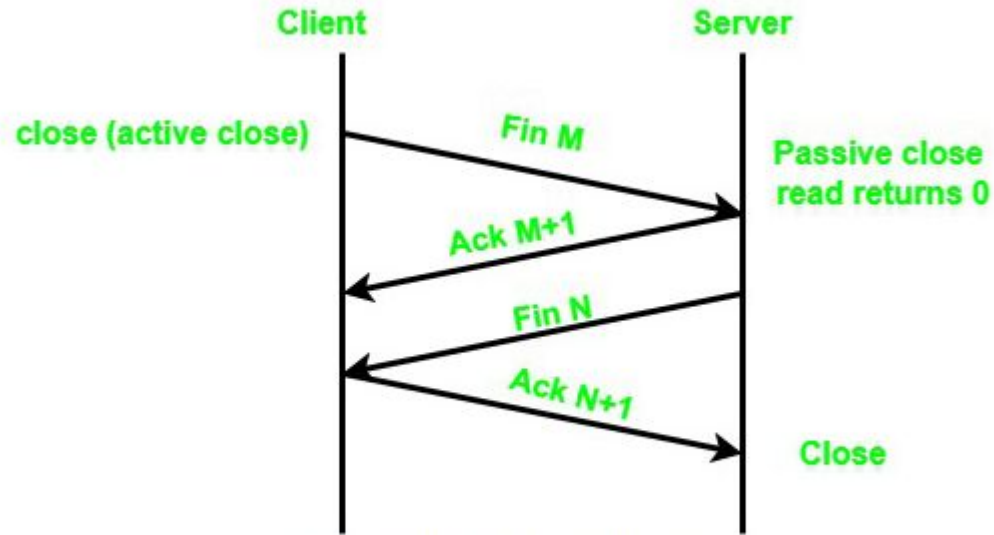
No.	Time	Source	Destination	Protocol	Length	Info
4	1.513342	192.168.0.103	192.168.0.130	TCP	66	54770→80 [SYN] Seq=0 Win=8192 Len=0 MSS=1260 WS=256 SACK_PERM=1
8	1.513924	192.168.0.130	192.168.0.103	TCP	66	80→54770 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
11	1.514021	192.168.0.103	192.168.0.130	TCP	54	54770→80 [ACK] Seq=1 Ack=1 Win=534016 Len=0

Connection Closing

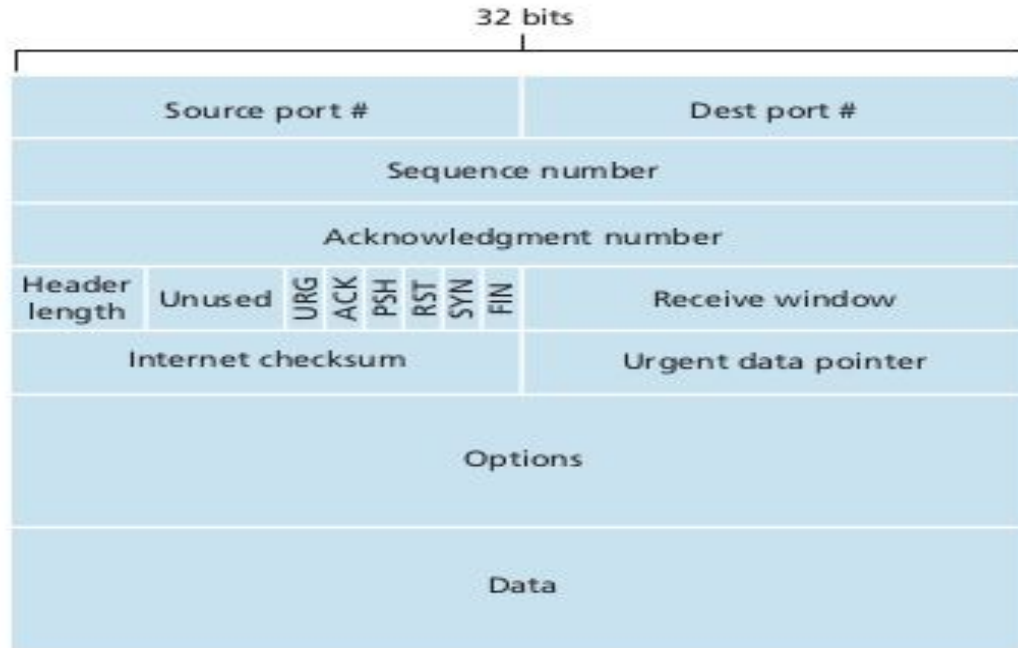
- To terminate an established TCP connection, the following 4 TCP packets are needed to be exchanged.
 - Host A → Host B: FIN flag set.
 - Host B → Host A: ACK flag set.
 - Host B → Host A: FIN flag set.
 - Host A → Host B: ACK flag set.
- Firstly, from one side of the connection, either from the client or the server the FIN flag will be sent as the request for the termination of the connection.
- In the second step, whoever receives the FIN flag will then be sending an ACK flag as the acknowledgment for the closing request to the other side.
- And, at the Later step, the server will also send a FIN flag as the closing signal to the other side.
- In the final step, the TCP, who received the final FIN flag, will be sending an ACK flag as the final Acknowledgement for the suggested connection closing.

	No.	Time	Source	Destination	Protocol	Length	Info
	1	0.000000	192.168.10.226	192.168.11.12	TCP	54	19707 → 23 [FIN, ACK] Seq=1 Ack=1 Win=16583 Len=0
	2	0.004330	192.168.11.12	192.168.10.226	TCP	60	23 → 19707 [ACK] Seq=1 Ack=2 Win=479 Len=0
	3	0.015354	192.168.11.12	192.168.10.226	TCP	60	23 → 19707 [FIN, ACK] Seq=1 Ack=2 Win=479 Len=0
	4	0.015461	192.168.10.226	192.168.11.12	TCP	54	19707 → 23 [ACK] Seq=2 Ack=2 Win=16583 Len=0

4-way handshake



TCP segment structure



- The TCP segment consists of header fields and a data field.
- The 32-bit sequence number field and the 32-bit acknowledgment number field are used by the TCP sender and receiver in implementing a reliable data transfer service
- The 16-bit receive window field is used for flow control.
- The 4-bit header length field specifies the length of the TCP header
- The TCP receive window size is the amount of receive data (in bytes) that can be buffered during a connection.
- The flag field contains 6 bits.
 - The ACK bit is used to indicate that the value carried in the acknowledgment field is valid.
 - The RST, SYN, and FIN bits are used for connection setup and teardown,
 - Setting the PSH bit indicates that the receiver should pass the data to the upper layer immediately.
 - Finally, the URG bit is used to indicate that there is data in this segment that the sending-side upper-layer entity has marked as “urgent.”
- The location of the last byte of this urgent data is indicated by the 16-bit urgent data pointer field.

DNS

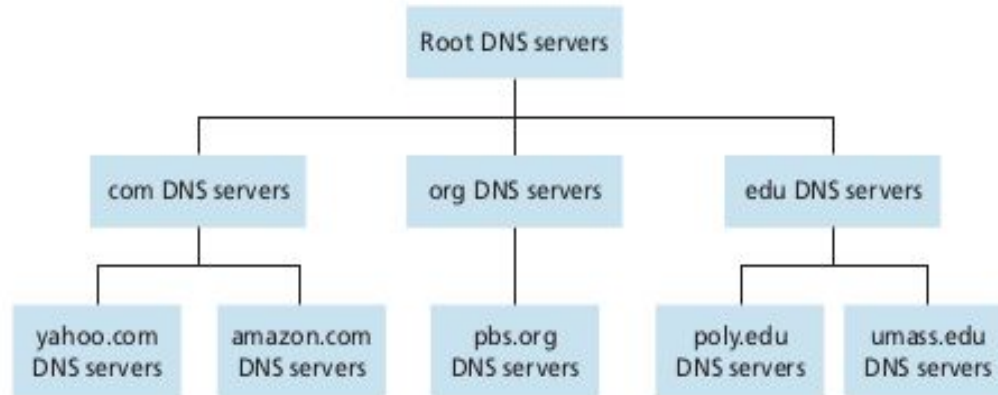
Introduction

- Just as humans can be identified in many ways, so too can Internet hosts.
- One identifier for a host is its hostname (cnn.com, www.yahoo.com)
- However, hostnames provide little, if any, information about the location within the Internet of the host.
- Furthermore, because hostnames can consist of variable-length alphanumeric characters, they would be difficult to process by routers.
- For these reasons, hosts are also identified by so-called IP addresses.
- An IP address is hierarchical because as we scan the address from left to right, we obtain more and more specific information about where the host is located in the Internet.
- People prefer the more mnemonic hostname identifier, while routers prefer fixed-length, hierarchically structured IP addresses.
- In order to reconcile these preferences, we need a directory service that translates hostnames to IP addresses. This is the main task of the Internet's domain name system (DNS).
- The DNS protocol runs over UDP and uses port 53.

- DNS is a distributed database implemented in a hierarchy of DNS servers.
- Why not centralize DNS?
 - single point of failure
 - traffic volume
 - distant centralized database
 - maintenance

A Distributed, Hierarchical Database

- The DNS uses a large number of servers, organized in a hierarchical fashion and distributed around the world.
- Client wants IP for `www.amazon.com`
 - client queries a root server to find `com` DNS server
 - client queries `com` DNS server to get `amazon.com` DNS server
 - client queries `amazon.com` DNS server to get IP address for `www.amazon.com`



Root Servers

The authoritative name servers that serve the DNS root zone, commonly known as the “root servers”, are a network of hundreds of servers in many countries around the world. They are configured in the DNS root zone as 13 named authorities, as follows.

List of Root Servers

HOSTNAME	IP ADDRESSES	OPERATOR
a.root-servers.net	198.41.0.4, 2001:503:ba3e::2:30	Verisign, Inc.
b.root-servers.net	199.9.14.201, 2001:500:200::b	University of Southern California, Information Sciences Institute
c.root-servers.net	192.33.4.12, 2001:500:2::c	Cogent Communications
d.root-servers.net	199.7.91.13, 2001:500:2d::d	University of Maryland
e.root-servers.net	192.203.230.10, 2001:500:a8::e	NASA (Ames Research Center)
f.root-servers.net	192.5.5.241, 2001:500:2f::f	Internet Systems Consortium, Inc.
g.root-servers.net	192.112.36.4, 2001:500:12::d0d	US Department of Defense (NIC)
h.root-servers.net	198.97.190.53, 2001:500:1::53	US Army (Research Lab)
i.root-servers.net	192.36.148.17, 2001:7fe::53	Netnod
j.root-servers.net	192.58.128.30, 2001:503:c27::2:30	Verisign, Inc.
k.root-servers.net	193.0.14.129, 2001:7fd::1	RIPE NCC
l.root-servers.net	199.7.83.42, 2001:500:9f::42	ICANN
m.root-servers.net	202.12.27.33, 2001:dc3::35	WIDE Project

Local Name Server

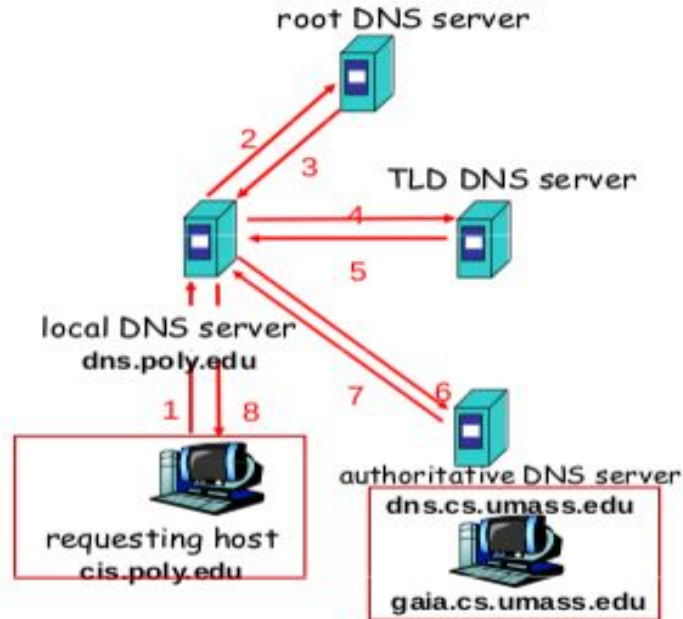
- Does not strictly belong to hierarchy
- Each ISP (residential ISP, company, university) has one, also called “default name server”
- When host makes DNS query, query is sent to its local DNS server– acts as proxy, forwards query into hierarchy

DNS name resolution example

- Host at cis.poly.edu wants IP address for gaia.cs.umass.edu

iterated query:

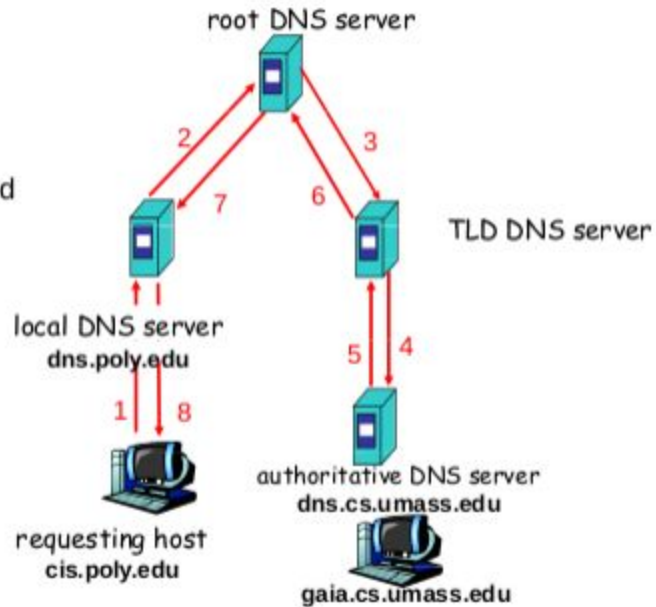
- contacted server replies with name of server to contact
- "I don't know this name, but ask this server"



DNS name resolution example

recursive query:

- puts burden of name resolution on contacted name server
- heavy load?



DNS Caching

- The idea behind DNS caching is very simple. In a query chain, when a DNS server receives a DNS reply (containing, for example, a mapping from a hostname to an IP address), it can cache the mapping in its local memory.
- Because hosts and mappings between hostnames and IP addresses are by no means permanent, DNS servers discard cached information after a period of time
- Because of caching, the local DNS server will be able to immediately return the IP address of cnn.com to this second requesting host without having to query any other DNS servers.
- A local DNS server can also cache the IP addresses of TLD servers, thereby allowing the local DNS server to bypass the root DNS servers in a query chain

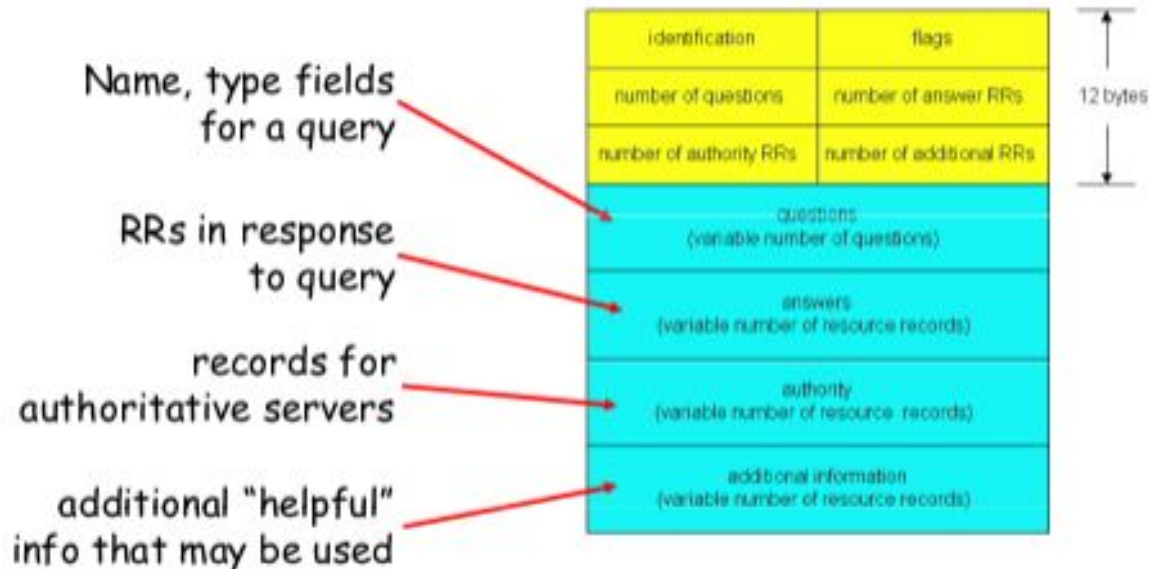
DNS Records and Messages

- Each DNS reply message carries one or more resource records.
- A resource record is a four-tuple that contains the following fields:

(Name, Value, Type, TTL)

- Type=A
 - **name** is hostname
 - **value** is IP address
- Type=NS
 - **name** is domain (e.g. foo.com)
 - **value** is hostname of authoritative name server for this domain
- Type=CNAME
 - **name** is alias name for some "canonical" (the real) name
www.ibm.com is really servereast.backup2.ibm.com
 - **value** is canonical name
- Type=MX
 - **value** is name of mailserver associated with **name**

DNS protocol, messages





003

	Source	Destination	Protocol	Length	Info
64624	200.200.200.55	200.200.200.15	DNS	71	Standard query 0x0003 AAAA www.mit.edu
23434	200.200.200.15	200.200.200.55	DNS	200	Standard query response 0x0003 AAAA www.mit.edu CNAME www.mit...

ion Port: 50951

L66

: 0x7e22 [unverified]

n Status: Unverified]

Index: 63]

pps]

oad (158 bytes)

e System (response)

ion ID: 0x0003

x8180 Standard query response, No error

```

00 35 c7 07 00 a6 7e 22 00 03 81 80 00 01  .7.5... ~"...
00 00 00 00 03 77 77 77 03 6d 69 74 03 65  ....w ww.mit.e
00 00 1c 00 01 c0 0c 00 05 00 01 00 00 05  du.....

```

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help



dns.id == 0x0003

No.	Time	Source	Destination	Protocol	Length	Info
562	11.364624	200.200.200.55	200.200.200.15	DNS	71	Standard query 0x0003 AAAA www.mit.edu
565	11.423434	200.200.200.15	200.200.200.55	DNS	200	Standard query response 0x0003 AAAA www.mit.edu CNAME www.mit...

[Checksum Status: Unverified]
[Stream index: 63]
▸ [Timestamps]
UDP payload (29 bytes)
▼ Domain Name System (query)
Transaction ID: 0x0003
▼ Flags: 0x0100 Standard query
0... .. = Response: Message is a query
.000 0... .. = Opcode: Standard query (0)
.... ..0. = Truncated: Message is not truncated
.... ..1 = Recursion desired: Do query recursively
....0... .. = Z: reserved (0)
....0 = Non-authenticated data: Unacceptable
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 0
▼ Queries
▸ www.mit.edu: type AAAA, class IN
[Response In: 565]



dns.id == 0x0003

No.	Time	Source	Destination	Protocol	Length	Info
562	11.364624	200.200.200.55	200.200.200.15	DNS	71	Standard query 0x0003 AAAA www.mit.edu
565	11.423434	200.200.200.15	200.200.200.55	DNS	200	Standard query response 0x0003 AAAA www.mit.edu CNAME www.mit...

UDP payload (158 bytes)

Domain Name System (response)

Transaction ID: 0x0003

Flags: 0x8180 Standard query response, No error

1... .. = Response: Message is a response
.000 0... .. = Opcode: Standard query (0)
... .0.. = Authoritative: Server is not an authority for domain
... ..0. = Truncated: Message is not truncated
... ..1 = Recursion desired: Do query recursively
...1... = Recursion available: Server can do recursive queries
...0.. ... = Z: reserved (0)
...0. = Answer authenticated: Answer/authority portion was not authenticated by the server
...0 = Non-authenticated data: Unacceptable
...0000 = Reply code: No error (0)

Questions: 1

Answer RRs: 4

Authority RRs: 0

Additional RRs: 0

Queries

www.mit.edu: type AAAA, class IN

Answers

www.mit.edu: type CNAME, class IN, cname www.mit.edu.edgekey.net
www.mit.edu.edgekey.net: type CNAME, class IN, cname e9566.dscb.akamaiedge.net
e9566.dscb.akamaiedge.net: type AAAA, class IN, addr 2600:140f:2c00:181::255e
e9566.dscb.akamaiedge.net: type AAAA, class IN, addr 2600:140f:2c00:1b7::255e

[\[Request In: 562\]](#)

[Time: 0.058810000 seconds]

A basic **DNS response** has:

- **Transaction Id**-for identification of the communication done.
- **Flags**-for verification of response whether it is valid or not.
- **Questions**-default is 1 for any request sent or received. It mainly denotes whether you have queried for something or not.
- **Answers**-default is 0 if the response is sent, and it's 1 if received. If the received packet is viewed then the Answers section has the IP address of the desired domain name along with **Time to Live** which is basically a counter which expires after its allotted time.

Queries section which gives the subjective details of the communication. The queries section has the following:

- **Name:** Domain name of the destination or web address to be reached or reached by in case of the received packet.
- **Type:** which is 'A' for IPv4(32 bits) and is 'AAAA' for IPv6(128 bits).
- **Class:** which is 'IN' by default, which means an internet IP address has been asked for.

THANK YOU