بسم الله الرحمن الرحیم

تکنولوژی کامپیوتر

جلسه‌ی ششم

شبکه – اچ‌تی‌تی‌پی

جلسه‌ی گذشته

گولنگِ بیشتر

دانلود پکیج و کار کردن با چندتا پکیج

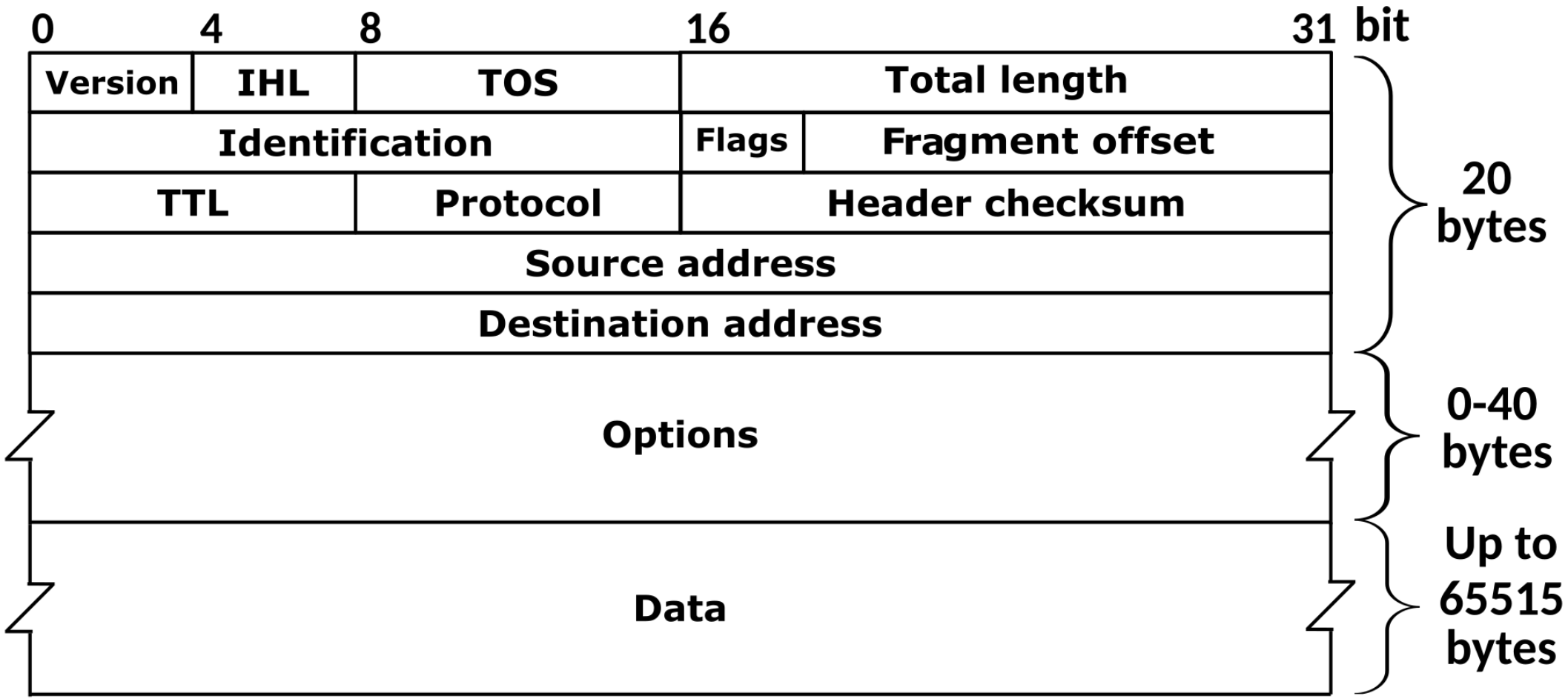
JSON

- معرفی struct tag
- با مثال json

لایه‌های شبکه در مدل TCP/IP

- Link Layer
- Internet Layer
- Transport Layer
- Application Layer

IP


0
4
8
16
31
bit
Version
IHL
TOS
Total length
Identification
Flags
Fragment offset
TTL
Protocol
Header checksum
Source address
Destination address
Options
Data
20 bytes
0-40 bytes
Up to 65515 bytes

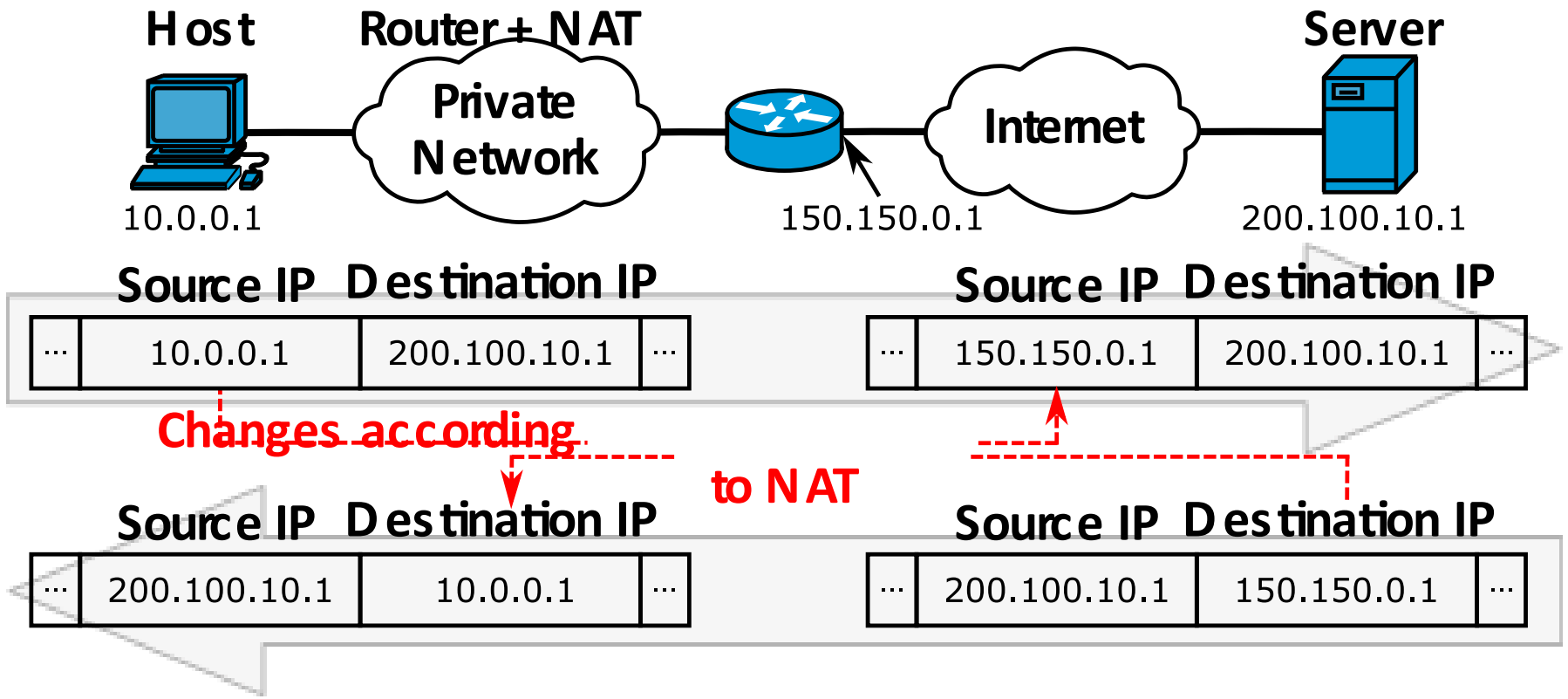
UDP

UDP header format[7]

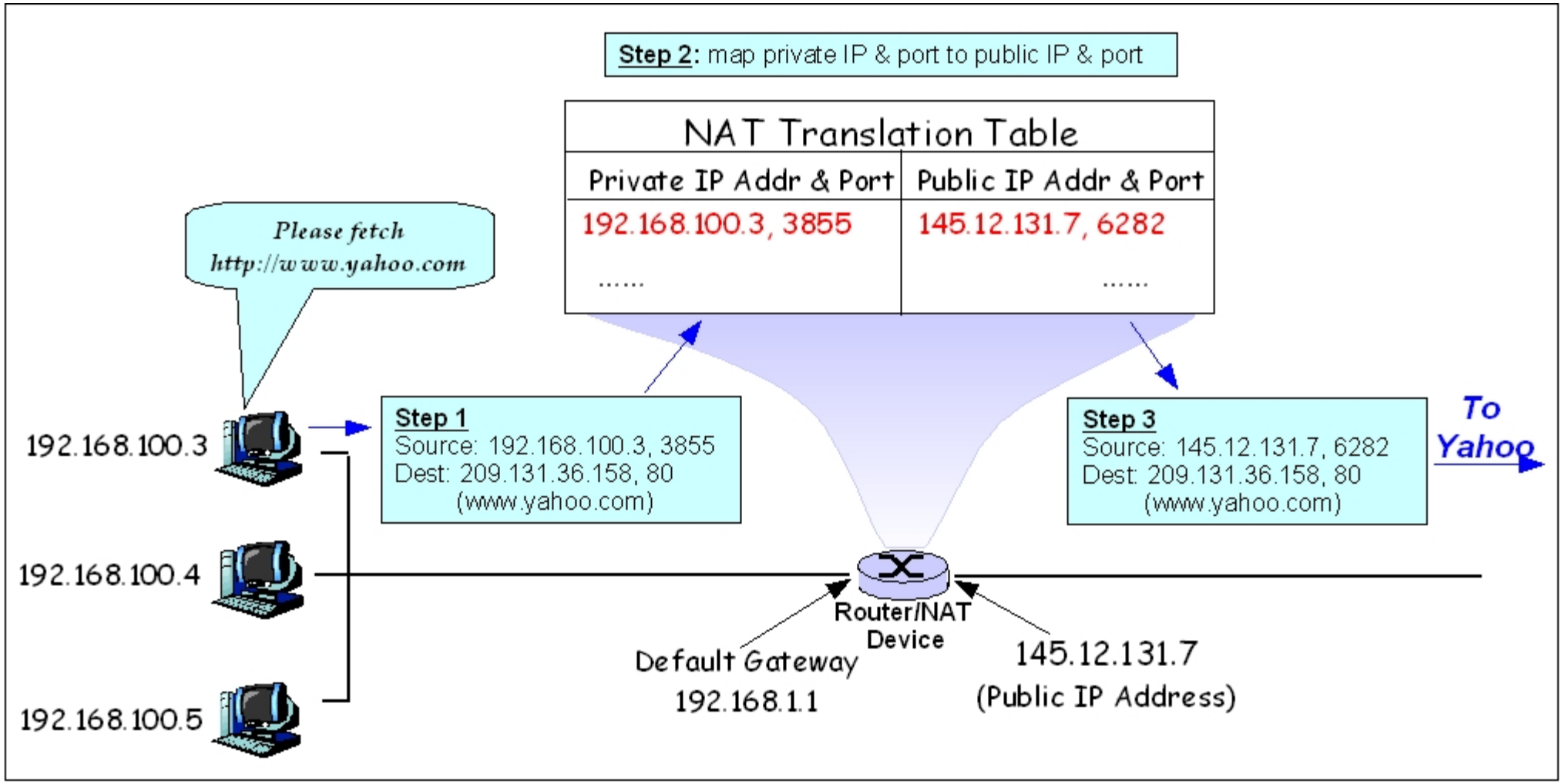
Offset	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	*Source Port*																*Destination Port*															
4	32	*Length*																*Checksum*															
8	64	*Data*																															
12	96																																
⋮	⋮																																

جلسه جدید

NAT (Network address translation)



NAT (Network address translation)



DNS

- How do we efficiently locate resources?
  - *DNS: name → IP address*
- Challenge
  - *How do we scale these to the wide area?*

DNS Goals

- Basically a wide-area distributed database
- Scalability
- Decentralized maintenance
- Robustness
- Global scope
  - *Names mean the same thing everywhere*
- Don’t need
  - *Atomicity*
  - *Strong consistency*

DNS Records

RR format: **(class, name, value, type, ttl)**

- DB contains tuples called resource records (RRs)

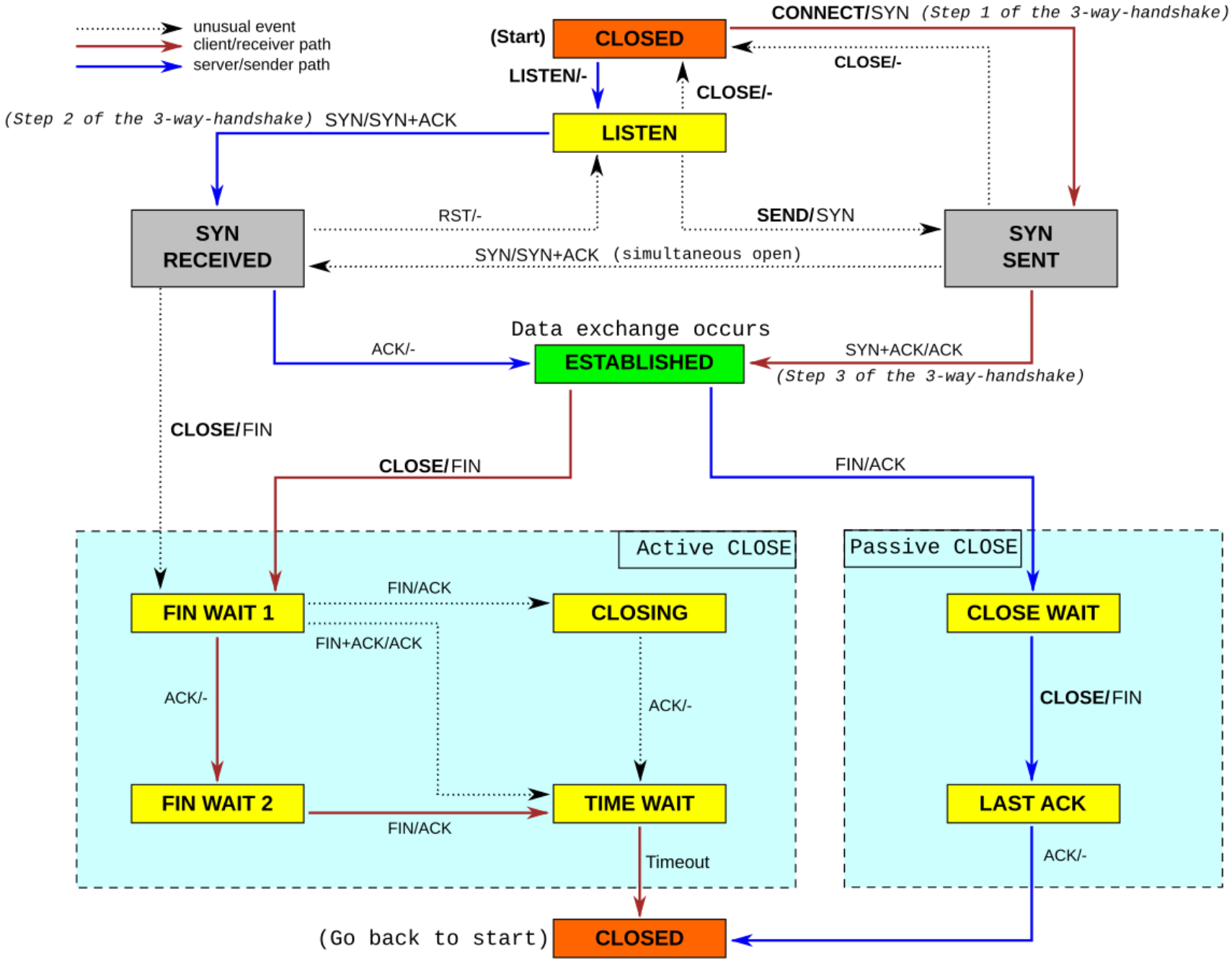
- Type=A
  - ***name*** *is hostname*
  - ***value*** *is IP address*
- Type=NS
  - ***name*** *is domain (e.g. foo.com)*
  - ***value*** *is name of authoritative name server for this domain*

- Type=CNAME
  - **name** is an alias name for some "canonical" (the real) name
  - **value** is canonical name
- Type=MX
  - **value** is hostname of mailserver associated with **name**

Properties of DNS Host Entries

- Different kinds of mappings are possible:
  - *Simple case: 1-1 mapping between domain name and IP addr:*
    - `sharif.edu` maps to 152.89.13.54
  - *Multiple domain names maps to the same IP address:*
    - `ecourse.sharif.edu` and `ocw.sharif.edu` both map to 81.31.170.118
  - *Single domain name maps to multiple IP addresses:*
    - `Balad.ir` map to multiple IP addrs.
  - *Some valid domain names don't map to any IP address:*
    - for example: alaki.alireza.dev

TCP


unusual event
client/receiver path
server/sender path
(Start)
CLOSED
CONNECT/SYN (Step 1 of the 3-way-handshake)
CLOSE/-
LISTEN/-
CLOSE/-
LISTEN
(Step 2 of the 3-way-handshake) SYN/SYN+ACK
SYN RECEIVED
RST/-
SEND/SYN
SYN SENT
SYN/SYN+ACK (simultaneous open)
Data exchange occurs
ACK/-
ESTABLISHED
SYN+ACK/ACK
(Step 3 of the 3-way-handshake)
CLOSE/FIN
CLOSE/FIN
FIN/ACK
Active CLOSE
Passive CLOSE
FIN WAIT 1
FIN/ACK
CLOSING
CLOSE WAIT
FIN+ACK/ACK
ACK/-
ACK/-
CLOSE/FIN
FIN WAIT 2
TIME WAIT
LAST ACK
FIN/ACK
Timeout
ACK/-
(Go back to start)
CLOSED

TCP

TCP header format[17]

Offset	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source Port																Destination Port															
4	32	Sequence Number																															
8	64	Acknowledgement Number (meaningful when ACK bit set)																															
12	96	Data Offset				Reserved				CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window															
16	128	Checksum																Urgent Pointer (meaningful when URG bit set)[18]															
20	160	(Options) If present, Data Offset will be greater than 5. Padded with zeroes to a multiple of 32 bits, since Data Offset counts words of 4 octets.																															
⋮	⋮																																
56	448																																
60	480	Data																															
64	512																																
⋮	⋮																																

TCP

- **Reliable Data Transfer**: Lost or corrupt segments are detected and retransmitted.
- **Ordered Delivery**: Sequence numbers ensure data is reassembled in the correct order.
- **Flow Control**: The receiving end can tell the sender how much data it can handle at once.
- **Congestion Control**: TCP tries to sense network congestion and adjust the sending rate, helping to avoid overwhelming the network.
- C**onnection-Oriented**: The handshake before data transfer ensures both ends agree on parameters (sequence numbers, MSS, etc.).

HTTP (Hypertext Transfer Protocol)

HTTP Request

Methods

- GET: Retrieve a resource (no body typically).
- POST: Send data to the server (e.g., form submissions, file uploads).
- PUT: Update or replace a resource.
- PATCH: Partial update of a resource.
- DELETE: Delete a resource.
- HEAD: Same as GET but returns no body (just headers).
- OPTIONS: Query the server for supported HTTP methods or capabilities.

HTTP Headers

HTTP Request Body

HTTP Response

HTTP Status Code

HTTP Body

Cookies and Sessions

Authentication

HTTP Request Query Params

HTTP Caching