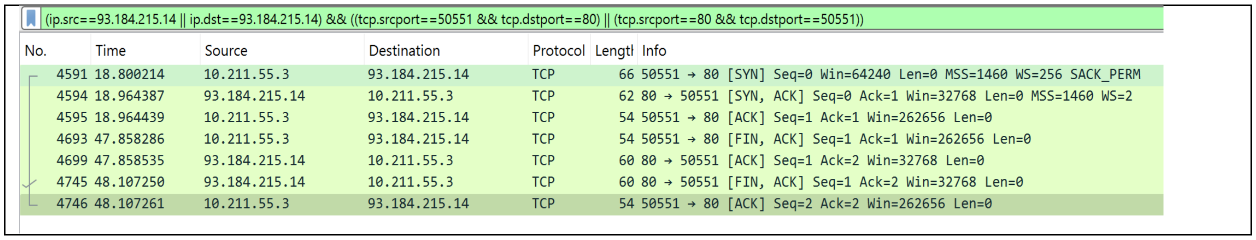
Class Notes

Simulations:

(1) TCP congestion control: https://media.pearsoncmg.com/ph/esm/ecs_kurose_compnetwork_8/cw/content/interactiveanimations/tcp-congestion/index.html

Wrapping up TCP discussion

(2) TCP peers use one sequence number for SYN segment and one last sequence for the FIN segment as well other than usual data. They do so because SYN (TCP connection startup) and FIN (TCP connection teardown) packet are critical, and we want reliability for them as well. Depending on the FIN sequence (which peer initiates it first), and the way FIN's acks are produced, sequence numbers used can vary. Let us see an example. I captured http traffic from http://www.example.com



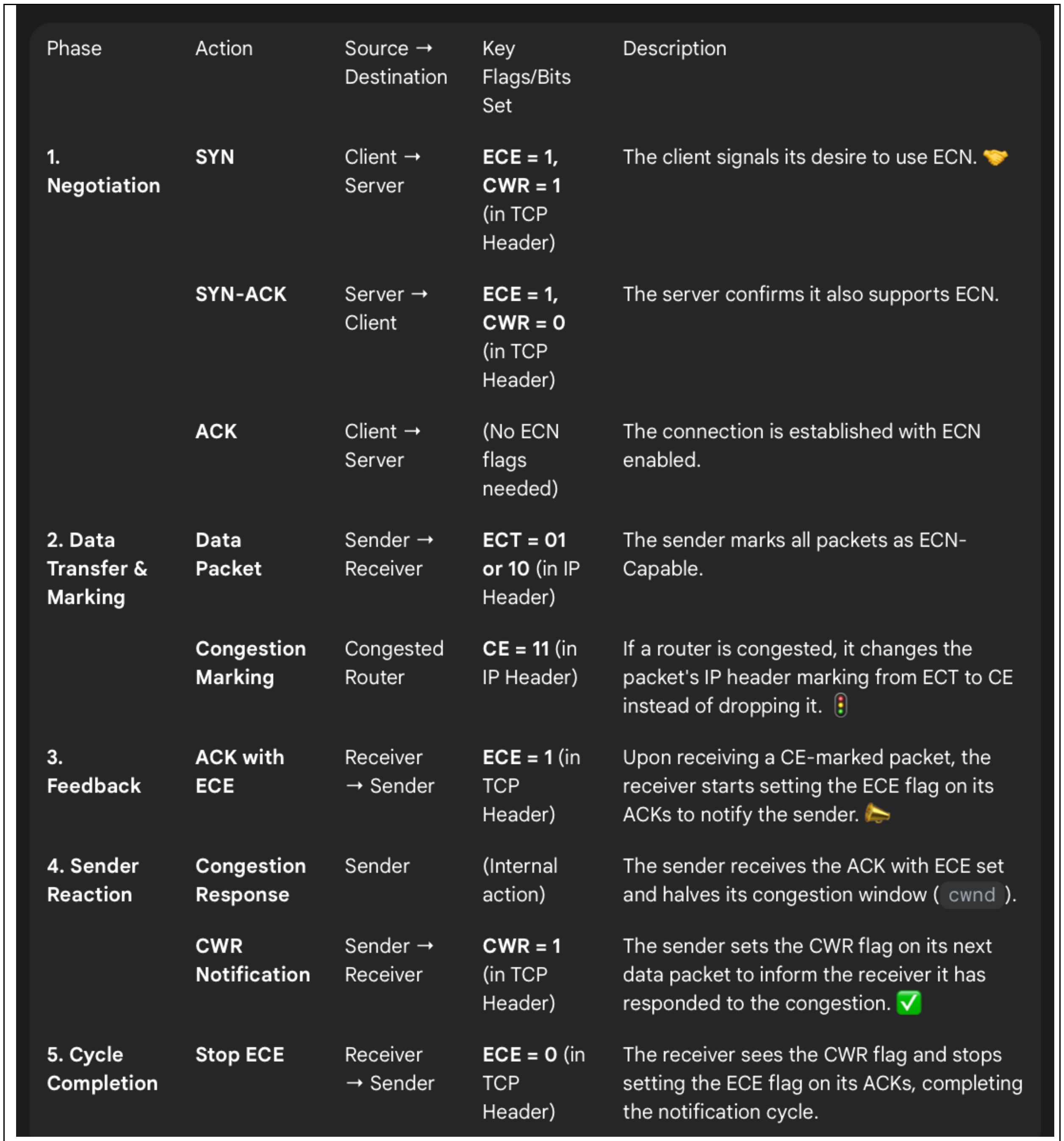
(ip.src==93.184.215.14 || ip.dst==93.184.215.14) && ((tcp.srcport==50551 && tcp.dstport==80) || (tcp.srcport==80 && tcp.dstport==50551))

No.	Time	Source	Destination	Protocol	Lengtl	Info
4591	18.800214	10.211.55.3	93.184.215.14	TCP	66	50551 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
4594	18.964387	93.184.215.14	10.211.55.3	TCP	62	80 → 50551 [SYN, ACK] Seq=0 Ack=1 Win=32768 Len=0 MSS=1460 WS=2
4595	18.964439	10.211.55.3	93.184.215.14	TCP	54	50551 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
4693	47.858286	10.211.55.3	93.184.215.14	TCP	54	50551 → 80 [FIN, ACK] Seq=1 Ack=1 Win=262656 Len=0
4699	47.858535	93.184.215.14	10.211.55.3	TCP	60	80 → 50551 [ACK] Seq=1 Ack=2 Win=32768 Len=0
4745	48.107250	93.184.215.14	10.211.55.3	TCP	60	80 → 50551 [FIN, ACK] Seq=1 Ack=2 Win=32768 Len=0
4746	48.107261	10.211.55.3	93.184.215.14	TCP	54	50551 → 80 [ACK] Seq=2 Ack=2 Win=262656 Len=0

(3) Congestion means: Too many senders sending in too much data in the network such that network queues are consistently building up and packet losses are happening.
   a. Goal is: Keep the "pipe" full but no fuller. We often think path from the sender to the receiver as a pipe through which data can flow.

(4) Two main ways to control congestion (and to potentially avoid it)
   a. Edge based approach (No help from the network layer)
   b. Explicit help from the network layer

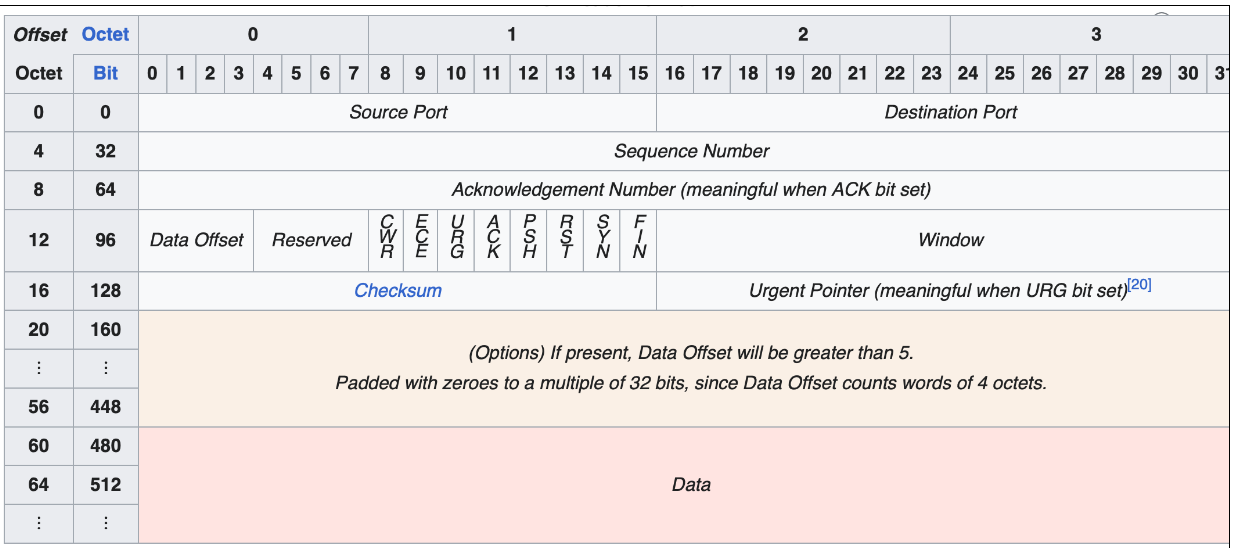
Explicit Congestion Notifications



Phase	Action	Source → Destination	Key Flags/Bits Set	Description
1. Negotiation	**SYN**	Client → Server	**ECE = 1, CWR = 1** (in TCP Header)	The client signals its desire to use ECN. 🤝
	SYN-ACK	Server → Client	**ECE = 1, CWR = 0** (in TCP Header)	The server confirms it also supports ECN.
	ACK	Client → Server	(No ECN flags needed)	The connection is established with ECN enabled.
2. Data Transfer & Marking	**Data Packet**	Sender → Receiver	**ECT = 01 or 10** (in IP Header)	The sender marks all packets as ECN-Capable.
	Congestion Marking	Congested Router	**CE = 11** (in IP Header)	If a router is congested, it changes the packet's IP header marking from ECT to CE instead of dropping it. 🚦
3. Feedback	**ACK with ECE**	Receiver → Sender	**ECE = 1** (in TCP Header)	Upon receiving a CE-marked packet, the receiver starts setting the ECE flag on its ACKs to notify the sender. 📣
4. Sender Reaction	**Congestion Response**	Sender	(Internal action)	The sender receives the ACK with ECE set and halves its congestion window (`cwnd`).
	CWR Notification	Sender → Receiver	**CWR = 1** (in TCP Header)	The sender sets the CWR flag on its next data packet to inform the receiver it has responded to the congestion. ✅
5. Cycle Completion	**Stop ECE**	Receiver → Sender	**ECE = 0** (in TCP Header)	The receiver sees the CWR flag and stops setting the ECE flag on its ACKs, completing the notification cycle.

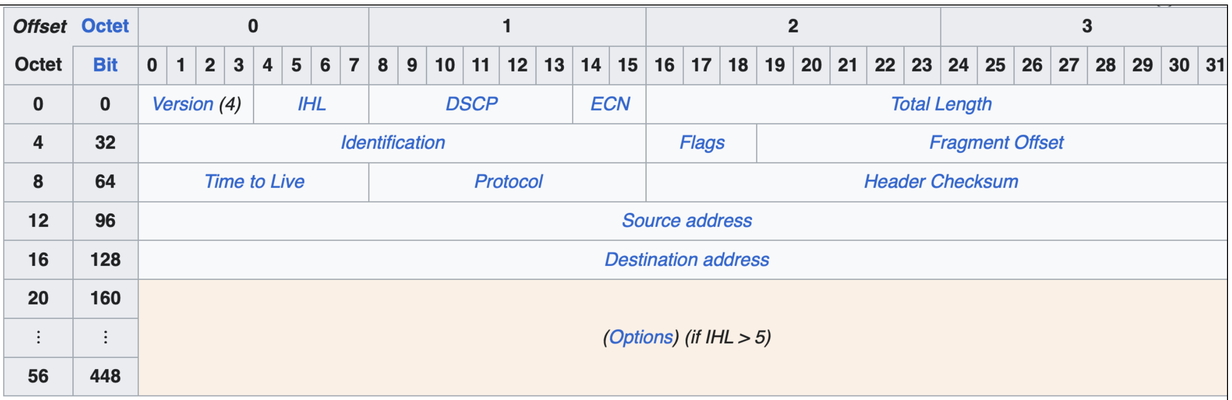
Source: Google Gemini and Wikipedia

TCP header:



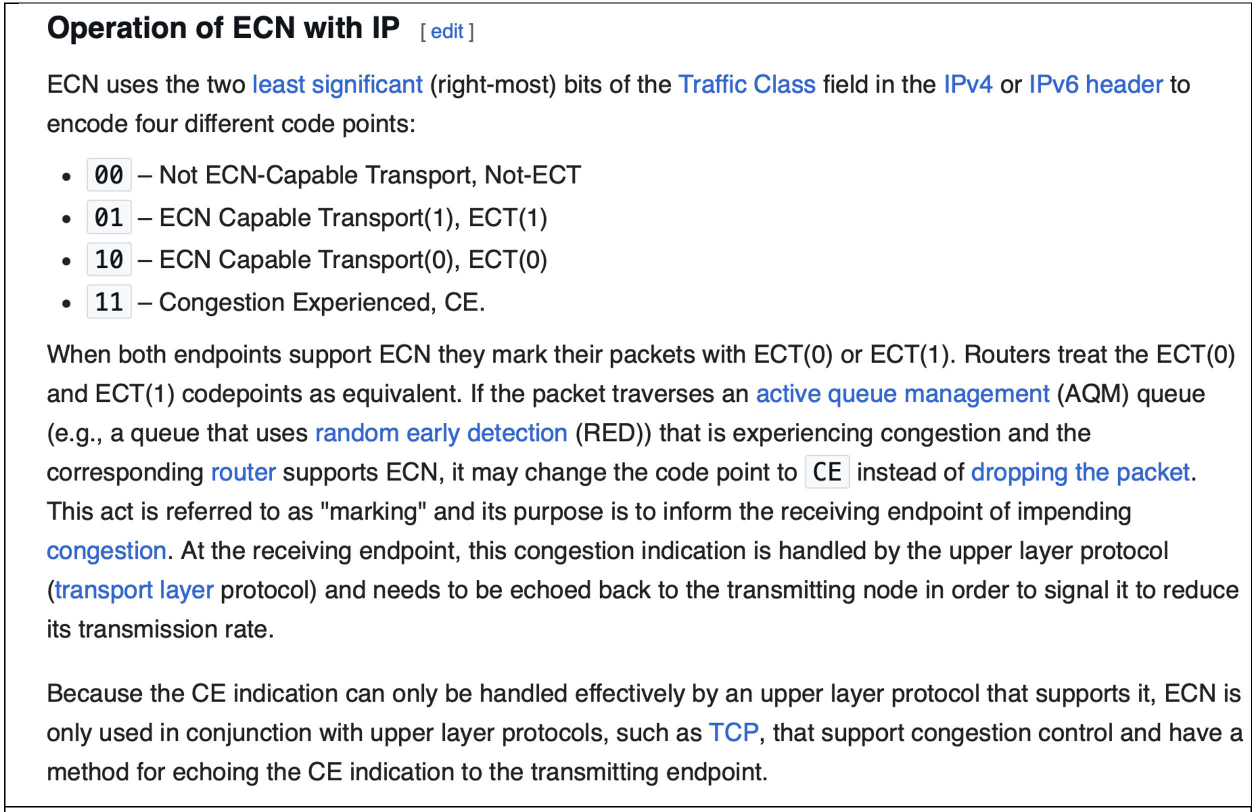
Offset	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	3
0	0	*Source Port*																*Destination Port*															
4	32	*Sequence Number*																															
8	64	*Acknowledgement Number (meaningful 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)*[20]															
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																																
⋮	⋮																																

ECE = ECN-Echo CWR = Congestion Window Reduced



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	*Version (4)*				*IHL*				*DSCP*						*ECN*		*Total Length*															
4	32	*Identification*																*Flags*			*Fragment Offset*												
8	64	*Time to Live*								*Protocol*								*Header Checksum*															
12	96	*Source address*																															
16	128	*Destination address*																															
20	160	*(Options) (if IHL > 5)*																															
⋮	⋮																																
56	448																																

IPv4 packet header



Operation of ECN with IP [edit]

ECN uses the two least significant (right-most) bits of the Traffic Class field in the IPv4 or IPv6 header to encode four different code points:

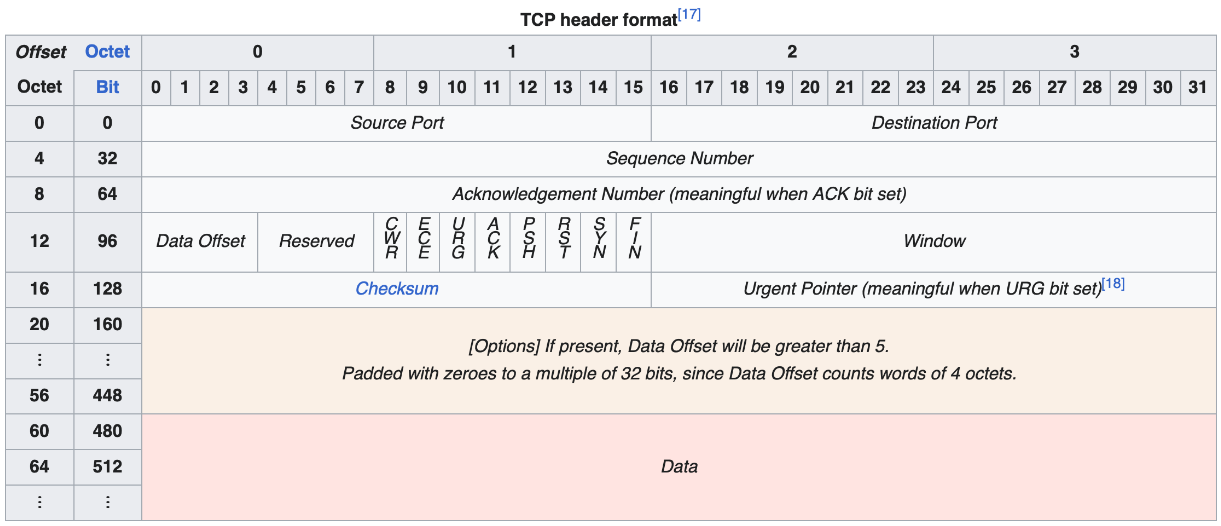
- `00` – Not ECN-Capable Transport, Not-ECT
- `01` – ECN Capable Transport(1), ECT(1)
- `10` – ECN Capable Transport(0), ECT(0)
- `11` – Congestion Experienced, CE.

When both endpoints support ECN they mark their packets with ECT(0) or ECT(1). Routers treat the ECT(0) and ECT(1) codepoints as equivalent. If the packet traverses an active queue management (AQM) queue (e.g., a queue that uses random early detection (RED)) that is experiencing congestion and the corresponding router supports ECN, it may change the code point to `CE` instead of dropping the packet. This act is referred to as "marking" and its purpose is to inform the receiving endpoint of impending congestion. At the receiving endpoint, this congestion indication is handled by the upper layer protocol (transport layer protocol) and needs to be echoed back to the transmitting node in order to signal it to reduce its transmission rate.

Because the CE indication can only be handled effectively by an upper layer protocol that supports it, ECN is only used in conjunction with upper layer protocols, such as TCP, that support congestion control and have a method for echoing the CE indication to the transmitting endpoint.

Source: Wikipedia

TCP Header



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																																
⋮	⋮																																

Source: Wikipedia

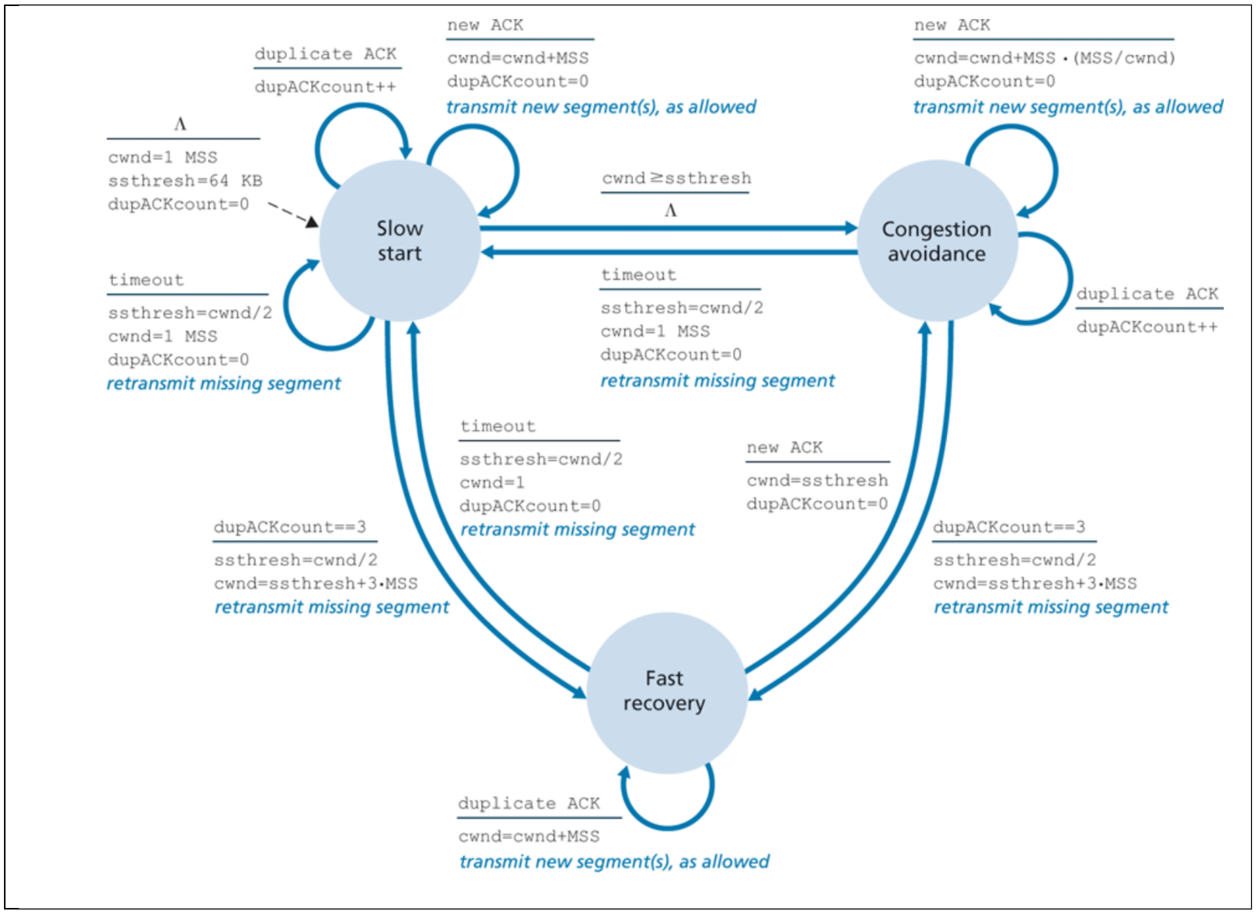
CWR: 1 bit
Congestion window reduced (CWR) flag is set by the sending host to indicate that it received a TCP segment with the ECE flag set and had responded in congestion control mechanism.[22][a]
ECE: 1 bit

ECN-Echo has a dual role, depending on the value of the SYN flag. It indicates:

1. If the SYN flag is set (1), the TCP peer is ECN capable.[23]
2. If the SYN flag is unset (0), a packet with the Congestion Experienced flag set (ECN=11) in its IP header was received during normal transmission.[a] This serves as an indication of network congestion (or impending congestion) to the TCP sender.[24]

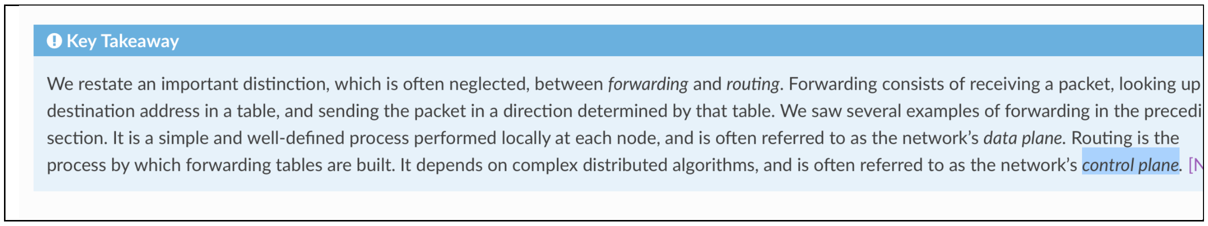
Source: Wikipedia

(5) TCP primarily uses packet losses as an indication of congestion.



The Network Layer (Data Plane) (Chapter 4)

1. Difference between data plane and control plane:



Key Takeaway

We restate an important distinction, which is often neglected, between *forwarding* and *routing*. Forwarding consists of receiving a packet, looking up destination address in a table, and sending the packet in a direction determined by that table. We saw several examples of forwarding in the precedi section. It is a simple and well-defined process performed locally at each node, and is often referred to as the network's *data plane*. Routing is the process by which forwarding tables are built. It depends on complex distributed algorithms, and is often referred to as the network's *control plane*. [N