Congestion control in TCP

Principles of Congestion control

What causes the congestion ?

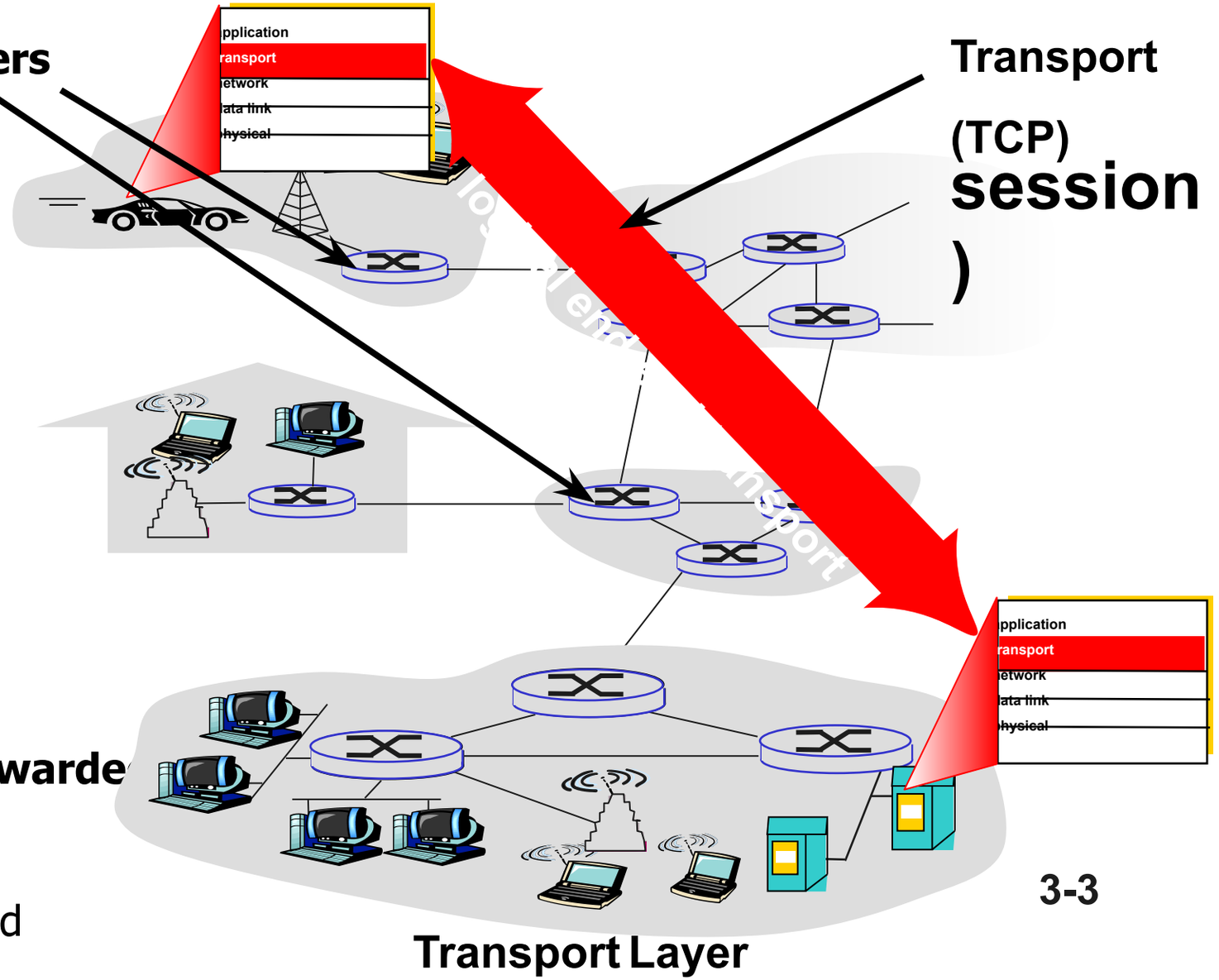
- Congestion is a **phenomena at the routers** of the internet / network

It is a phenomena that happens when the **'load' is more than 'capacity' of the router**

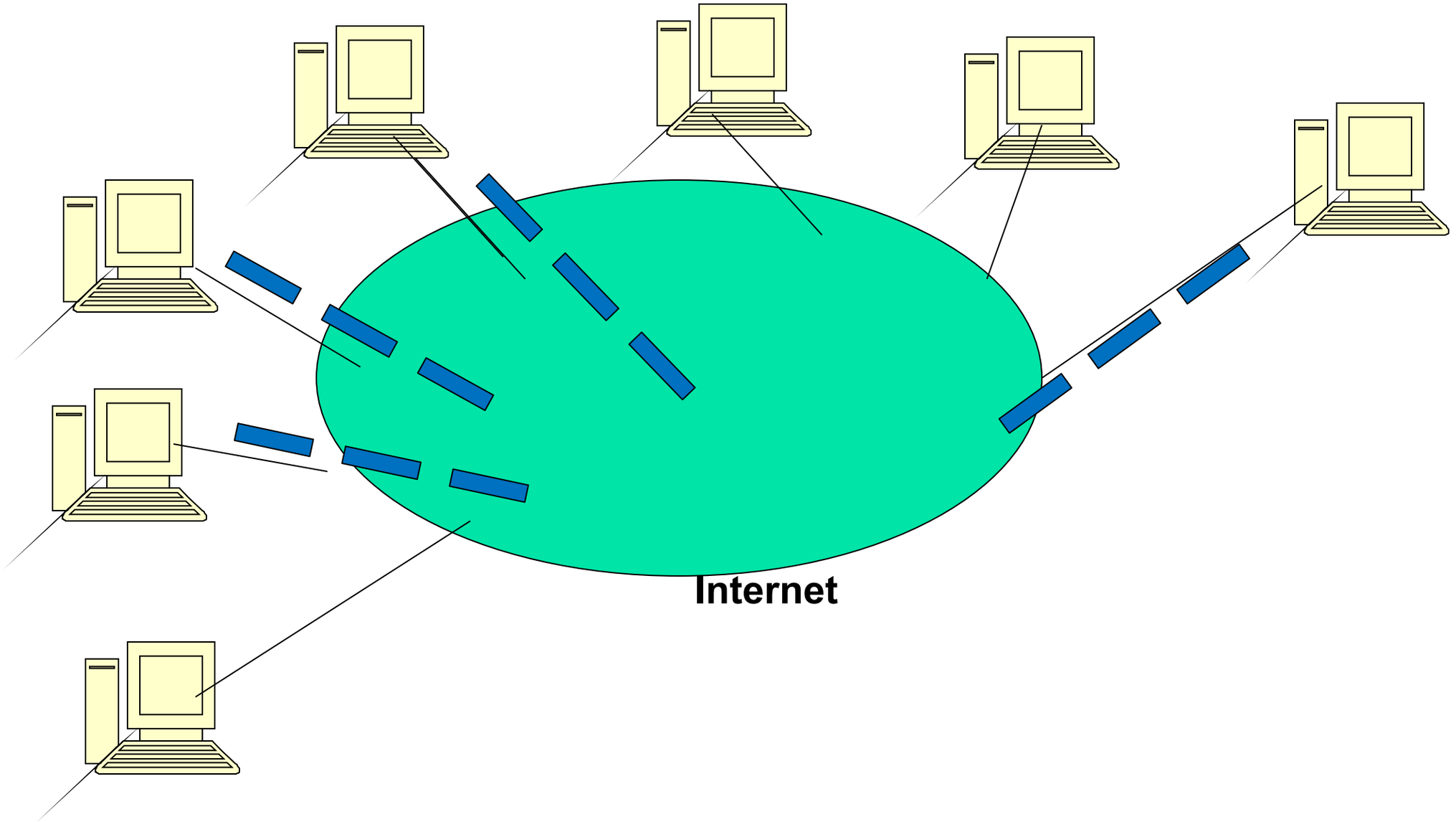
what do we mean by this ?
Router being a packet switch, has to forward the sequence of packets arriving at the input to the appropriate output

Capacity means,
how many packets/second can be forwarde
from the input to the output ?

Packets are stored / queued and forwarded



If too many packets arrive at routers (heavy load), packets may be discarded


Internet

Fundamental question :

If the congestion is the phenomena of in the routers (Network Layer) present in the core network , why should Transport layer , which is at the edge, should bother ?

Here is the reason :

How to ‘control’ congestion ?

Simple; Control the traffic (amount and rate of packets) entering in to the network.

Whose responsibility is this ?

It is the responsibility of every host (edge device) connected to internet.

Controlling the traffic means ……

- Every host , if the congestion is detected , should have the discipline to reduce the

Fundamental question :

Which layer, in every host, has to do this?

Best layer is TRANSPORT LAYER , because :

1. It is there at every end host.

2. It is the ‘department’ that takes the messages from the ‘application layer department’ and sends to the internet through the lower layers.

3. In other words, transport layer is the traffic generating source.

4. 3. It has the responsibility of end to end , process to process communication.

If transport layer at every host follows certain algorithm / mechanism to control the traffic

Yes,

This is the role of congestion control algorithms in transport layer.

Alright,

Traffic generation from Transport layer should depend on the ‘Congestion’ occurring at the network.

But the challenges are :

1. How to detect the congestion ?
2. How to know the ‘degree’ of congestion ?
    - Is it high ? if so how much ?

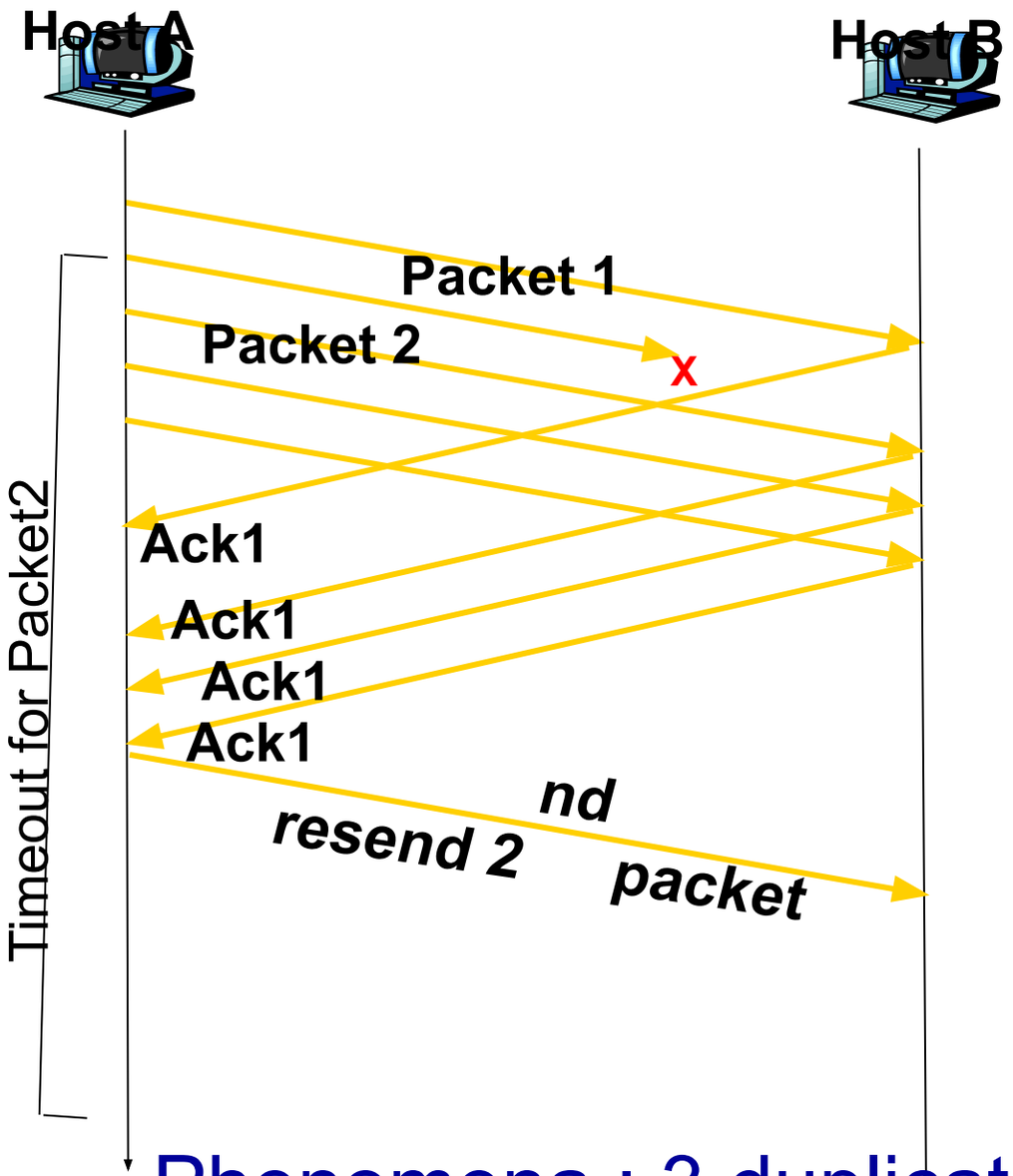
How to detect congestion ?

There is no explicit mechanism to indicate congestion

After sending a segment, not getting the acknowledgement, before the time out is the only way for the sender to detect the loss event (either packet loss or acknowledgement loss)

TCP uses the phenomena, 3 duplicate acknowledgements, to detect the congestion.

What is the phenomena ?
“3 duplicate acknowledgements”



In this example, Packet 2 is lost.

Receiver goes on sending Ack1 to indicate that Packet1 is lost

" Minimum 3 duplicates indicate the Packet loss" before tome out.

If frequency of this phenomena is high, high congestion

Phenomena : 3 duplicate acknowledgements

Yes,

Main philosophy is

reduce the traffic (amount of data) to be sent.

But ,

how much to reduce ?

TCP sender maintains a variable called **Congestion Window** **(CWND)**

23.1 TCP sender does , dynamically, a kind of *'trial and error'* method, to decide the

What is this cwnd ?

In fact, the amount of data to be sent depends on

1. Network capacity
2. Receiver capacity

Each sender maintains two windows :

1) Receiver Window (Window the receiver has granted)
2) Congestion window

Each reflects the number of bytes the sender may transmit

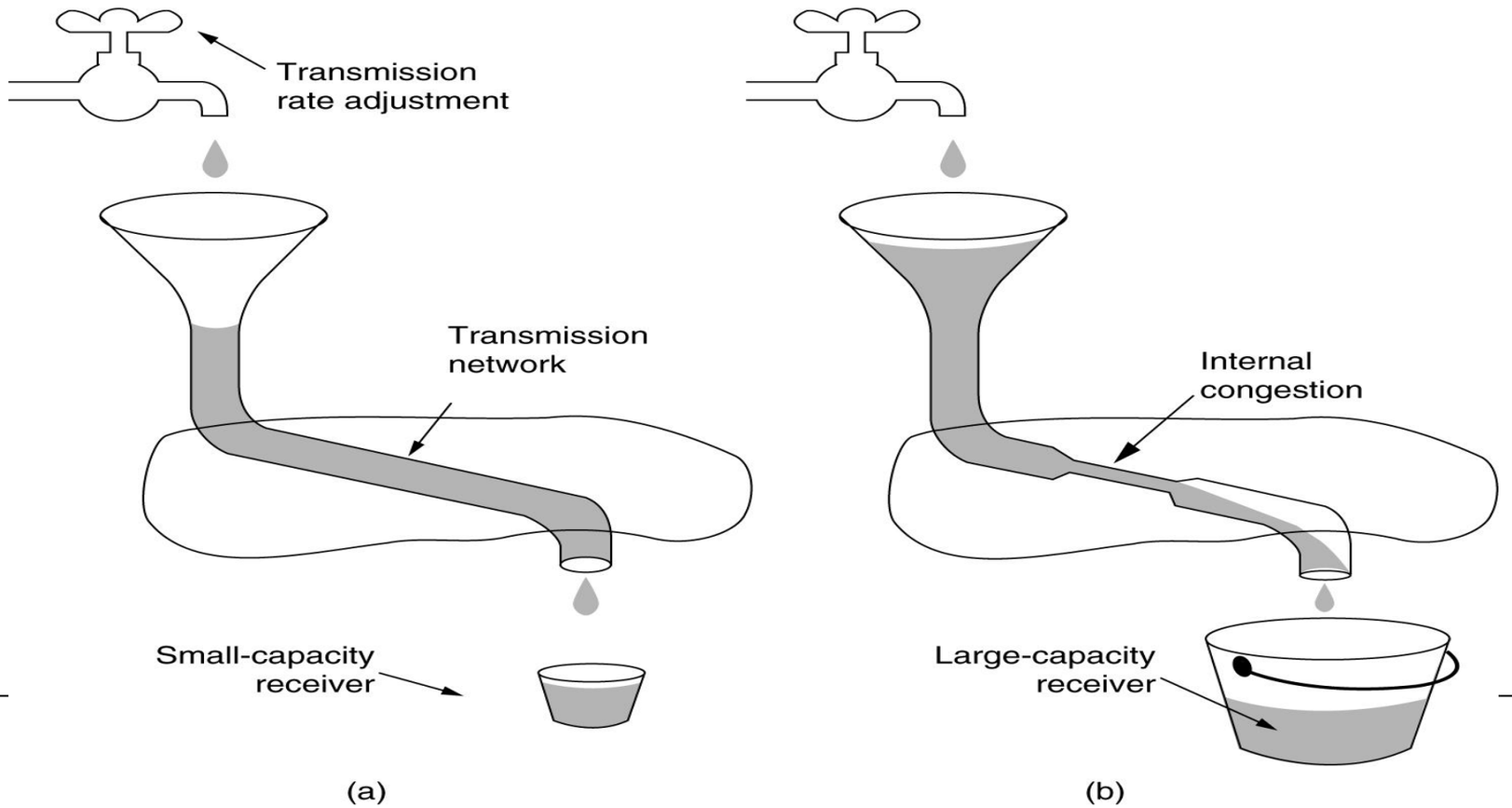
cwnd

[Congestion Window]

is related to congestion in

TCP Congestion Control

(a) A fast network feeding a low capacity receiver.
(b) A slow network feeding a high-capacity receiver.



The objective of TCP congestion control is to have each sender transmit just the right amount of data to keep the network resources utilised, but not overloaded

The number of bytes that may be sent is the minimum of the two windows

In other words,

The number of bytes that may be sent is the minimum of *what sender thinks is alright and what receiver thinks is alright*

#1

receiver says “send 8 KB”

Sender knows that bursts of more than 4 KB congest the network

Hence it sends only 4 KB

#2

receiver says “send 8 KB”

Sender knows that the network can easily forward upto 32 KB , without any congestion.

Hence it sends full 8 KB requested

TCP Congestion Control: details

- sender limits transmission:

  `LastByteSent-LastByteAcked`

  `≤ cwnd`

- roughly,

$$\text{rate} = \frac{\text{cwnd}}{\text{RTT}} \text{ Bytes/sec}$$

- **`cwnd`** is dynamic, function of perceived network congestion

How does sender perceive congestion?

- loss event = timeout *or* 3 duplicate acks
- TCP sender reduces rate (**`cwnd`**) after loss event

TCP congestion control algorithm

3 Phases of TCP congestion control algorithm

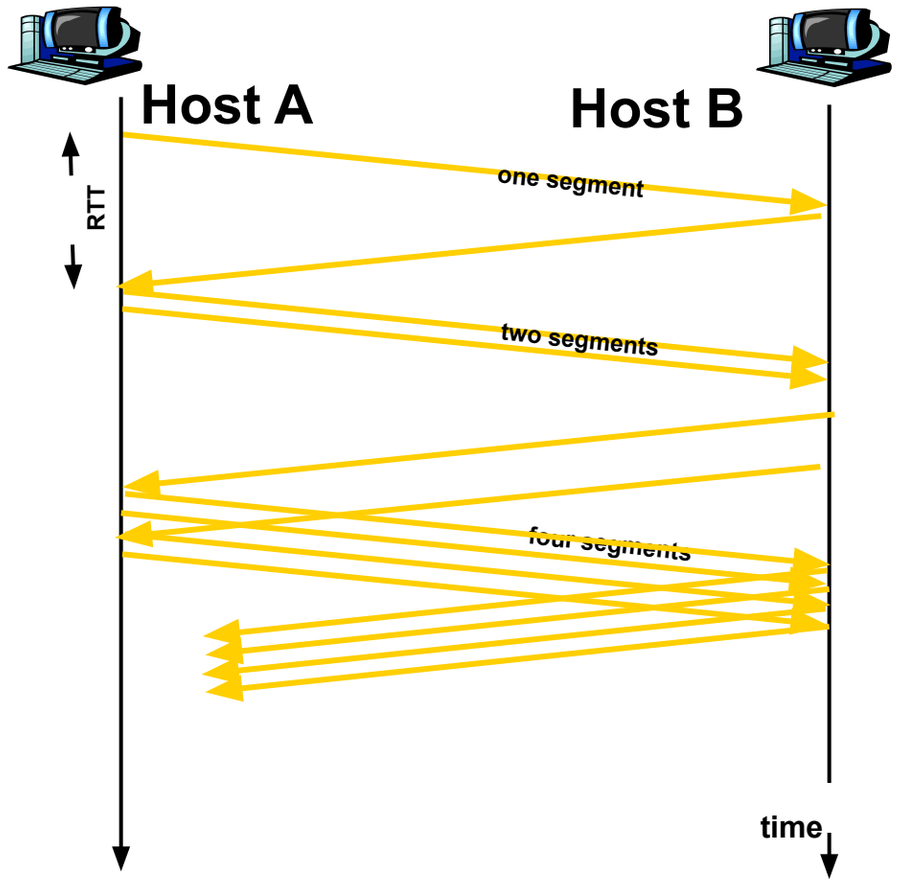
1. SLOW START [EXPONENTIAL INCREASE]

2. ADDITIVE INCREASE

3. MULTIPLICATIVE DECREASE

TCP Slow Start

- when connection begins, increase rate exponentially until first loss event:
  - initially **`cwnd`** = 1 MSS
  - double **`cwnd`** every RTT
  - done by incrementing **`cwnd`** for every ACK received
- summary: initial rate is slow but ramps up exponentially fast



Refinement: inferring loss

- after 3 dup ACKs:
  - **`cwnd`** is cut in half
  - window then grows linearly
- but after timeout event:
  - **`cwnd`** instead set to 1 MSS;
  - window then grows exponentially
  - to a threshold, then grows linearly

Philosophy:

- **3 dup ACKs indicates network capable of delivering some segments**
- **timeout indicates a "more alarming" congestion scenario**

An example of the Internet congestion algorithm.