Computer Network Architecture

计算机网络体系结构

沈航

南京工业大学

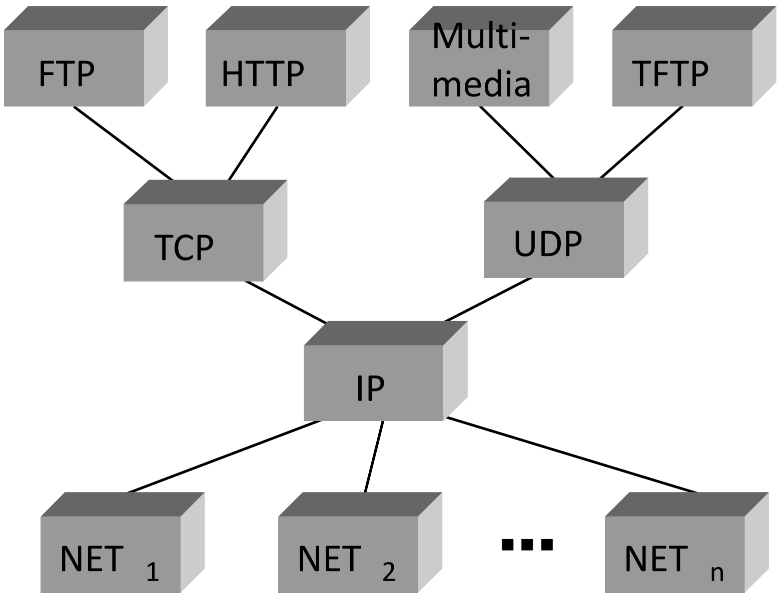
Email: hshen@njtech.edu.cn

The TCP/IP Architecture

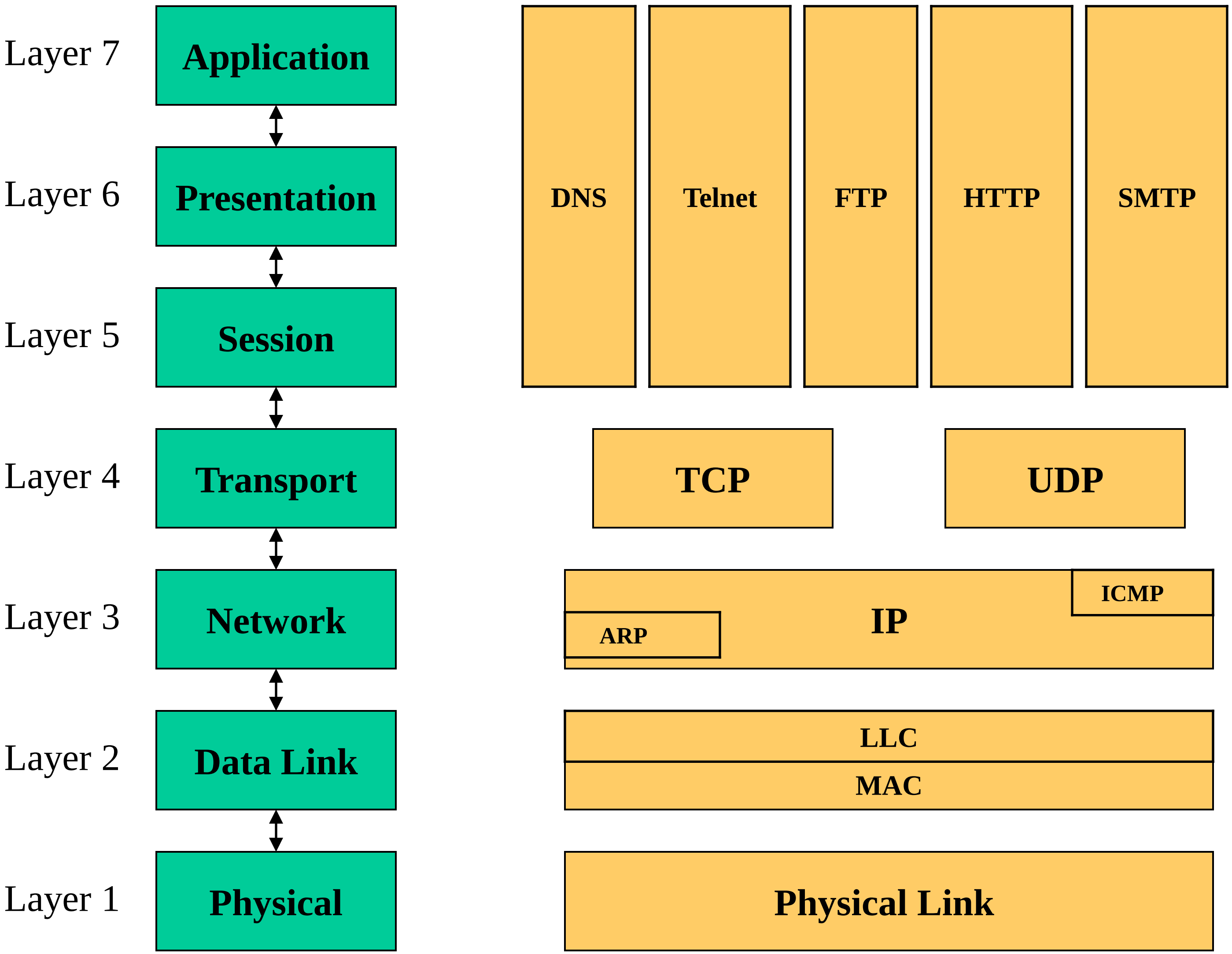
(continued)

TCP/IP Architecture

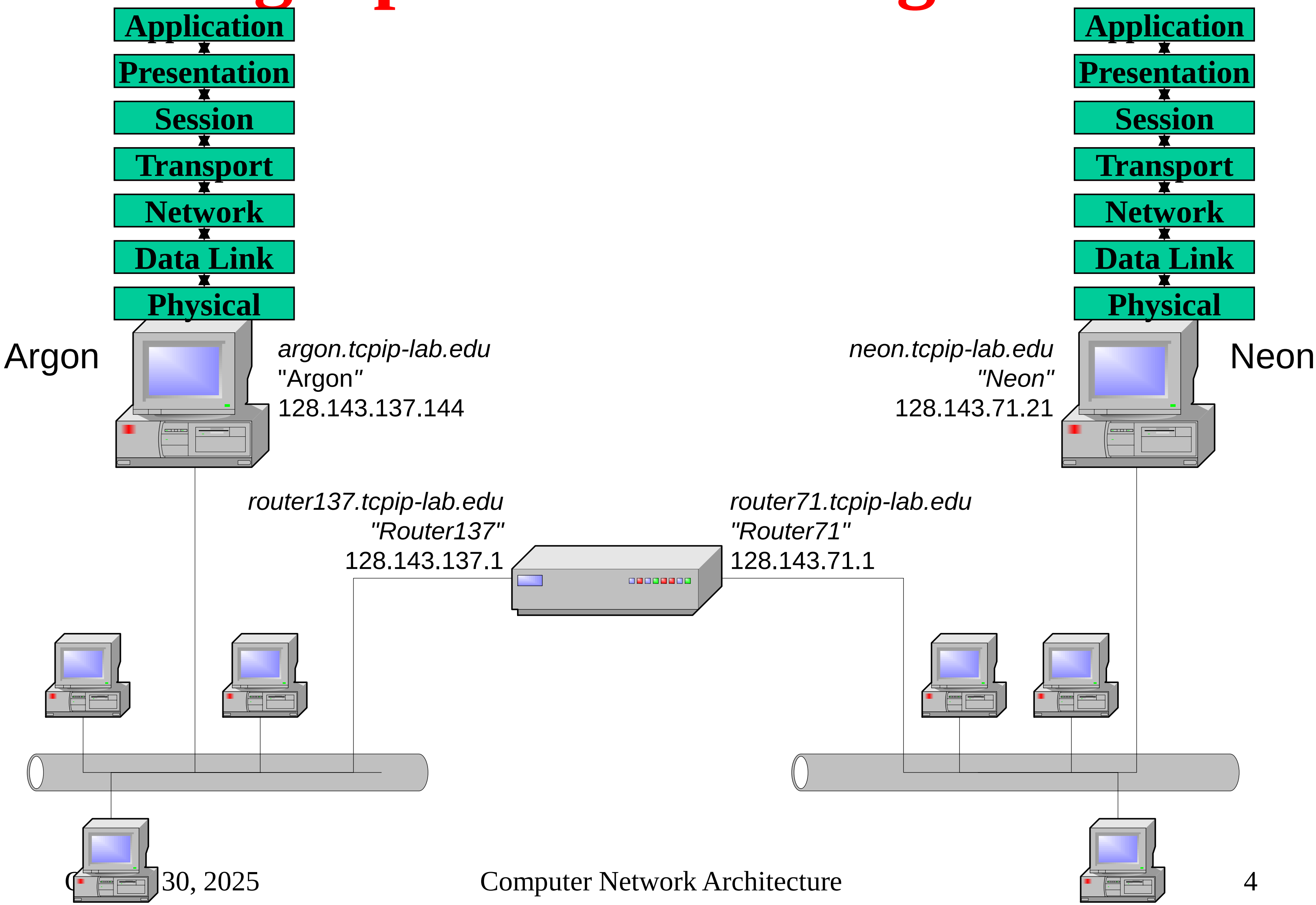
- **The TCP/IP Architecture defined by IETF**
- **Transparent Design**
  - ♠ **Everything over IP**
  - ♠ **IP over Everything**
  - ♠ **Best-effort**



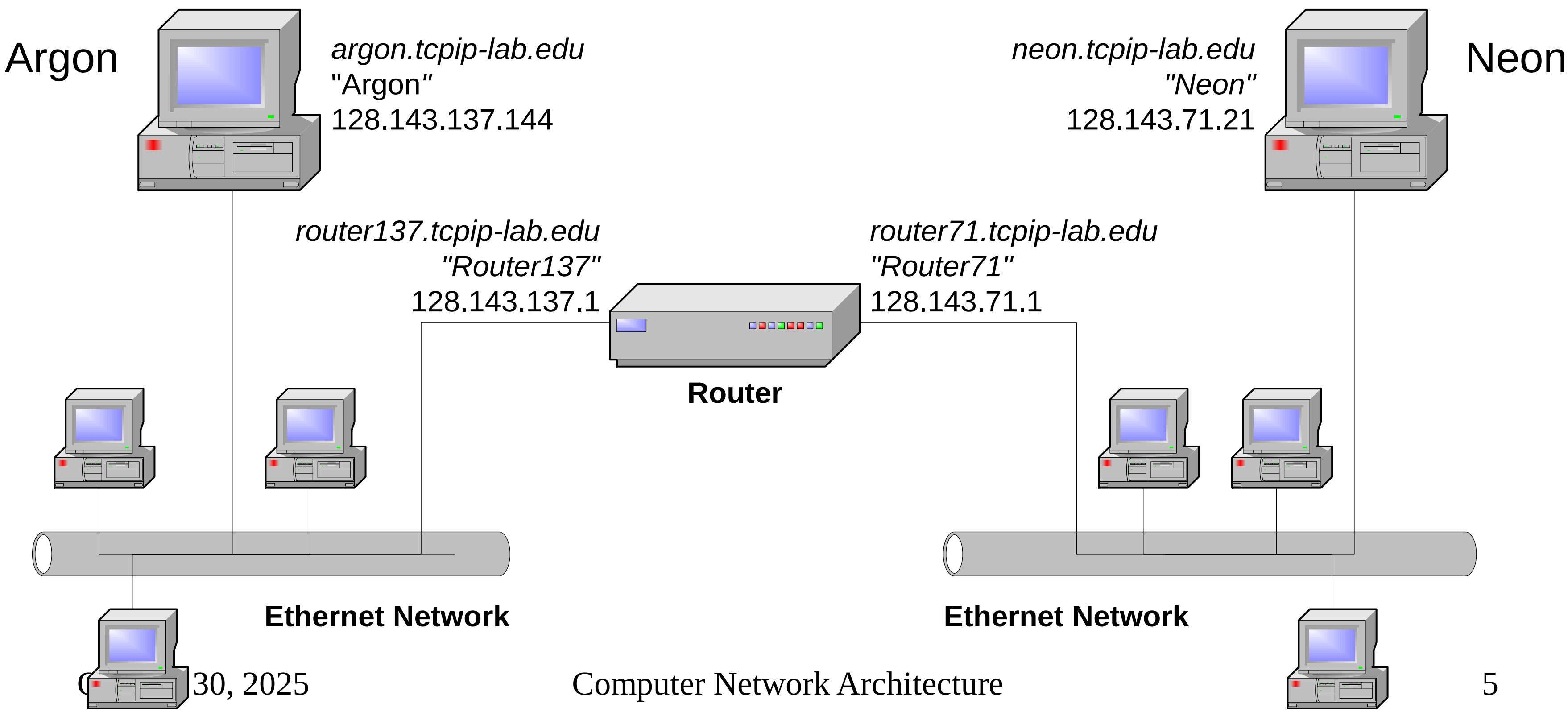
OSI Model versus TCP/IP



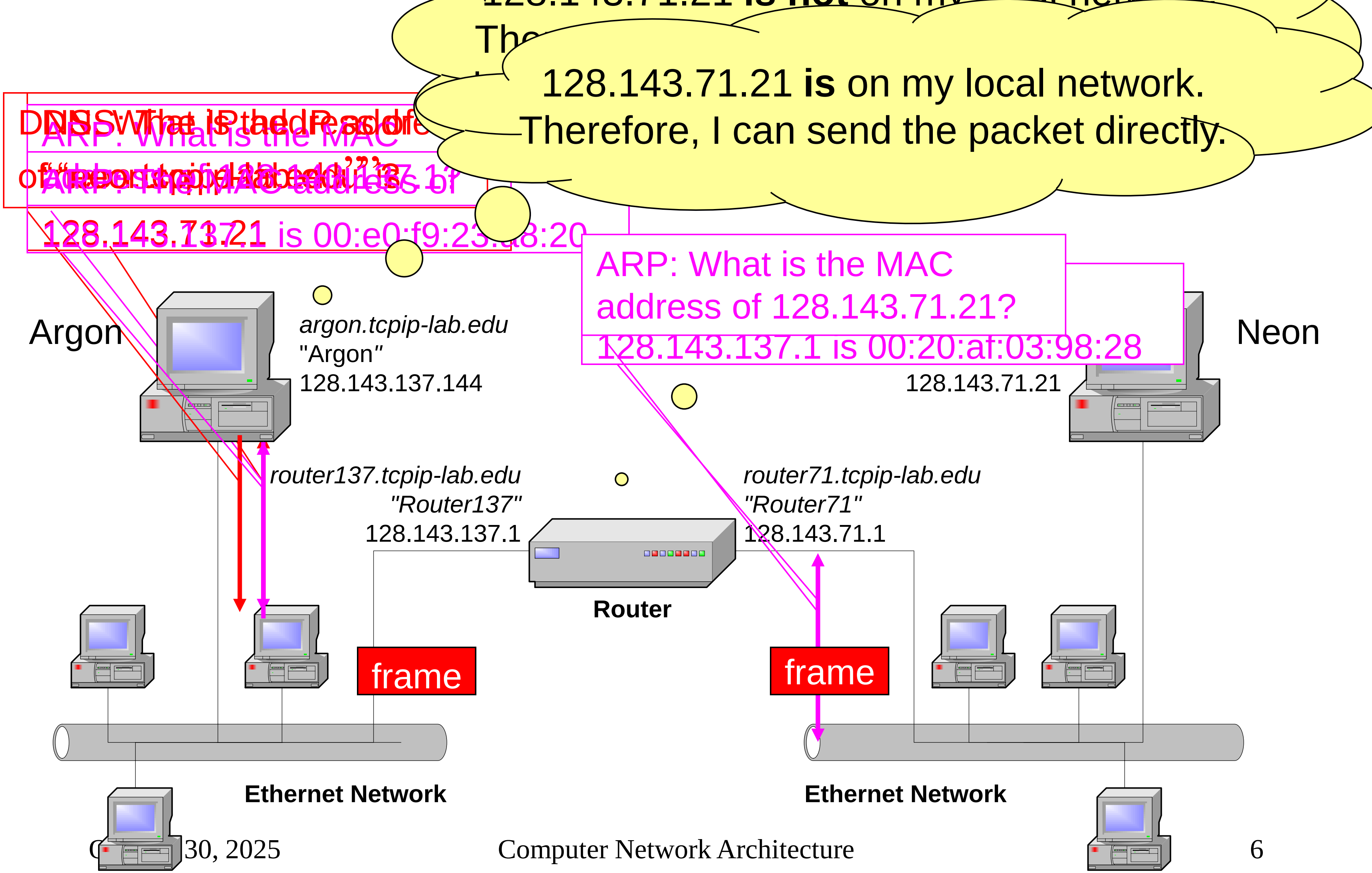
Sending a packet from Argon to Neon



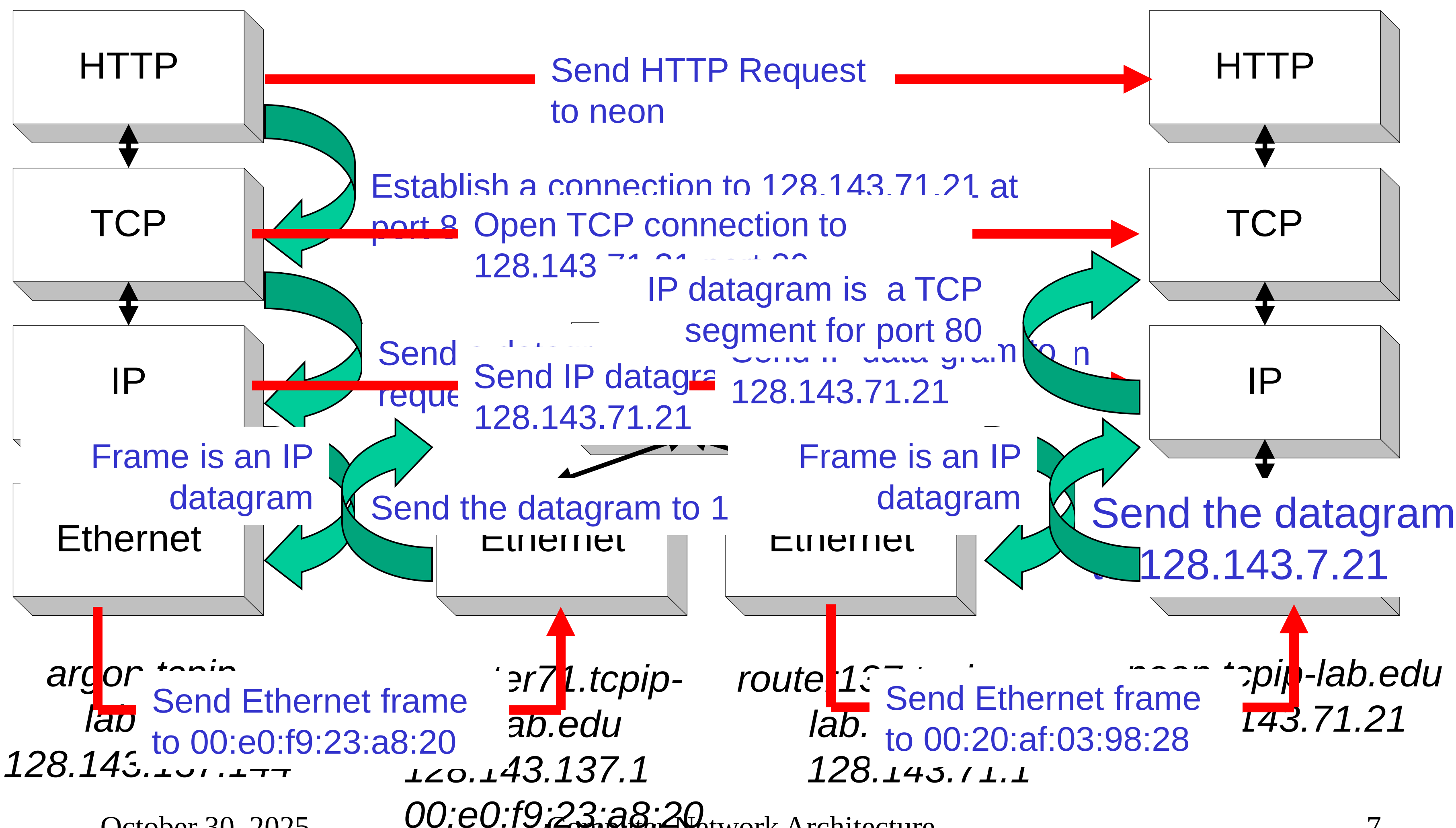
Sending a packet from Argon to Neon



Sending a pac

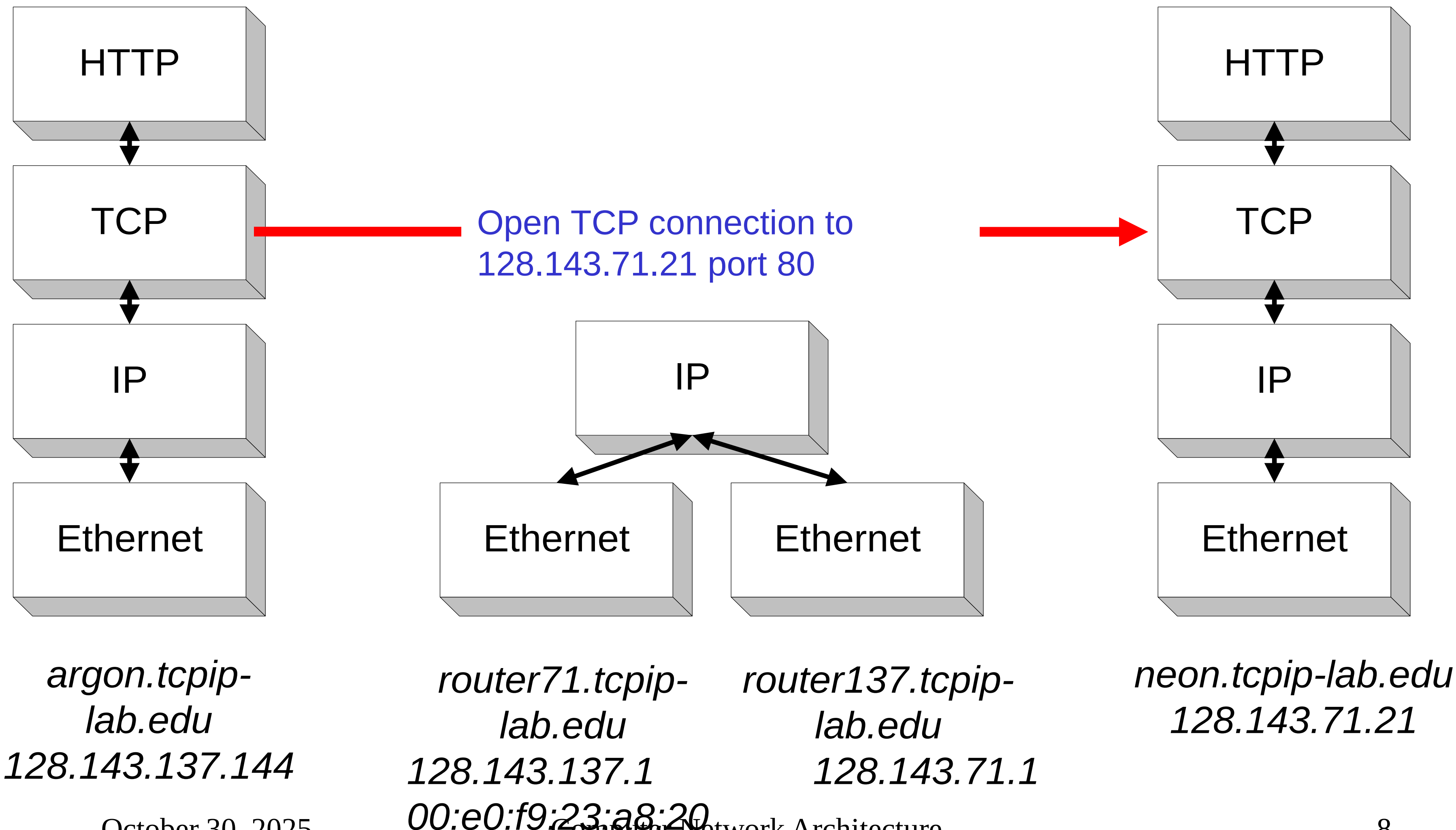


Layers in the Example



TCP 传输控制协议

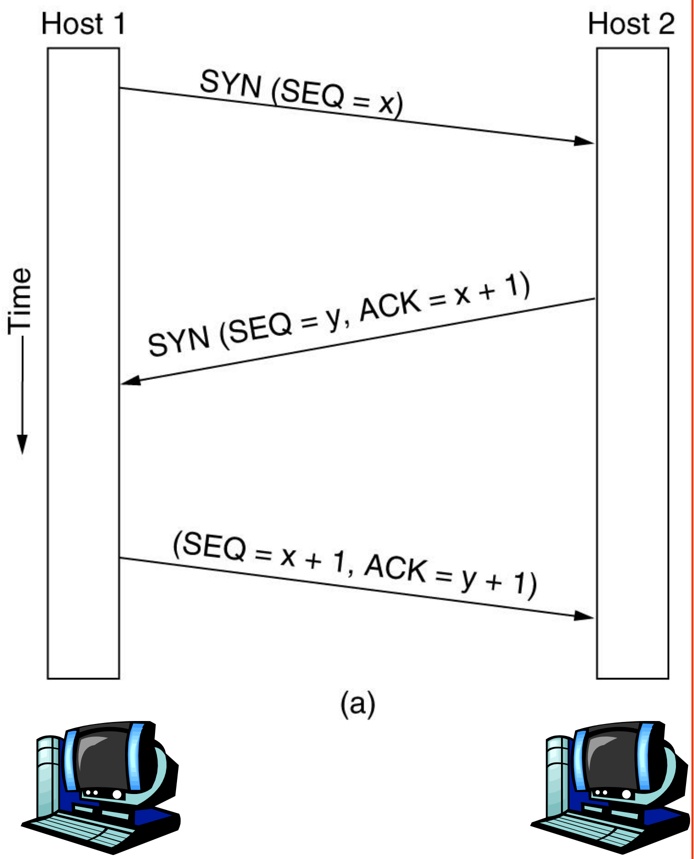
TCP is a connection-oriented protocol



TCP (Transmission Control Protocol) TCP 传输控制协议

- **TCP was specifically designed to provide a reliable end-to-end byte stream over an unreliable IP networks;**
- **TCP provides full duplex communication, with <u>Flow control</u> and <u>Congestion control</u>**

TCP Connection Establishment



Three way handshake:

Step 1: client host sends TCP SYN segment to server with initial seq number, but no data

Step 2: server host receives SYN, replies with SYN/ACK segment

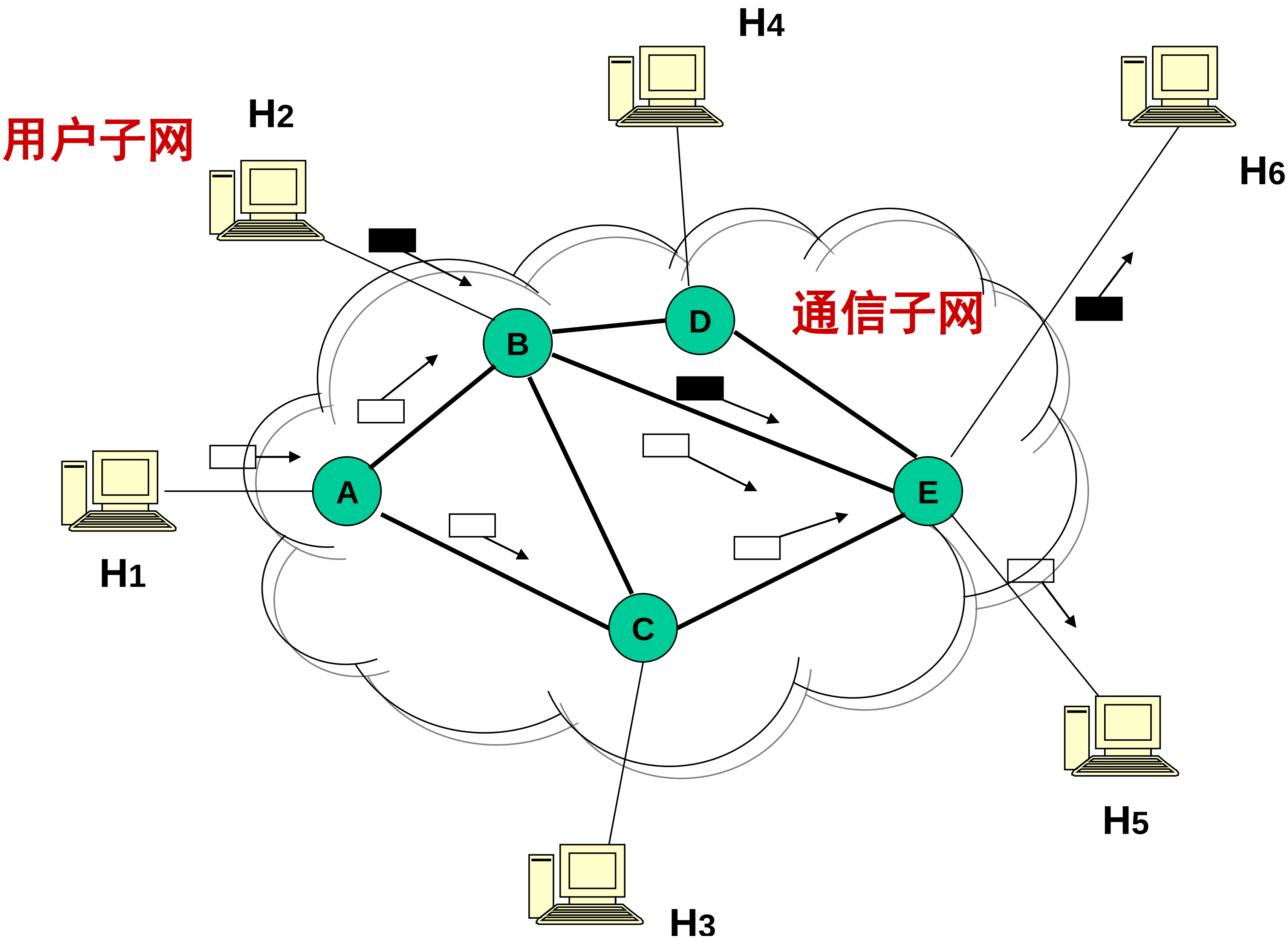
Step 3: client receives SYN/ACK, replies with ACK segment, which may contain data

Introduction to TCP

- **How to convert an unreliable connection into a reliable connection:**
  - **TCP numbers each segment and uses an ARQ protocol to recover lost segments.**
  - **The sliding window protocol滑动窗口协议 is used for flow control.**
  - **Some versions of TCP implement *Go Back N* and other versions implement *Selective Repeat*.**

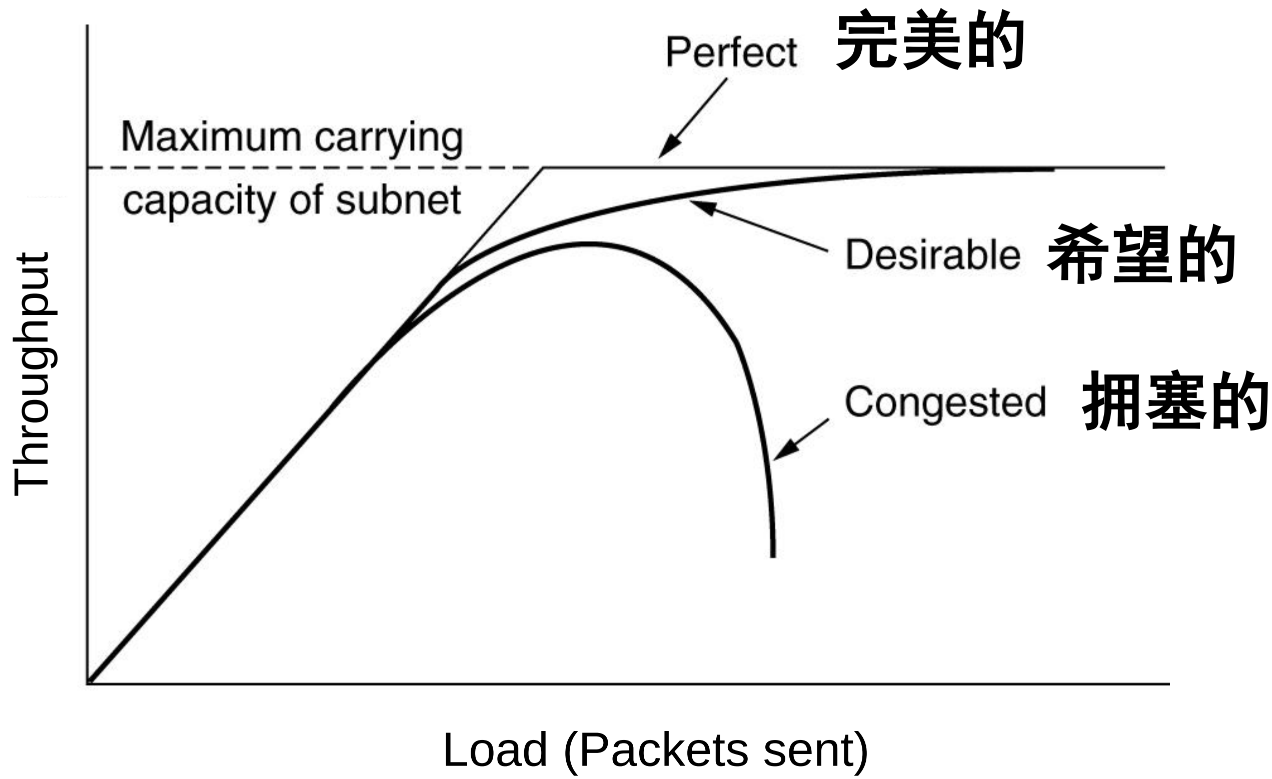
The TCP Protocol

- **The basic protocol used by TCP entities is the Sliding Window Protocol滑动窗口协议.**
- **To provide reliable transmission, the TCP must be robust enough in the face of many kinds of failures, and deal with many unexpected events.**



分组交换网示意图

Congestion

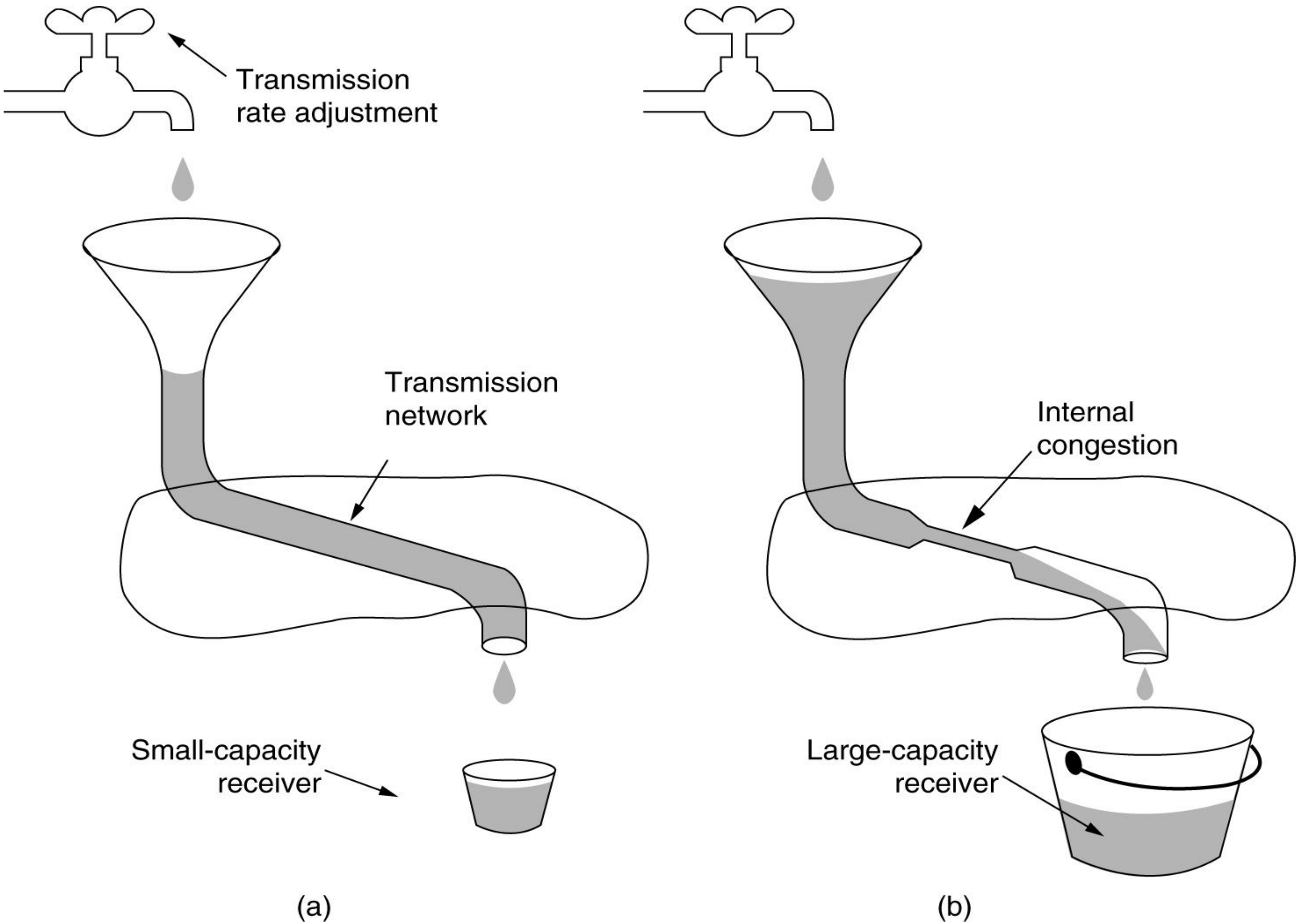


When too much traffic is offered, congestion sets in and performance degrades sharply.

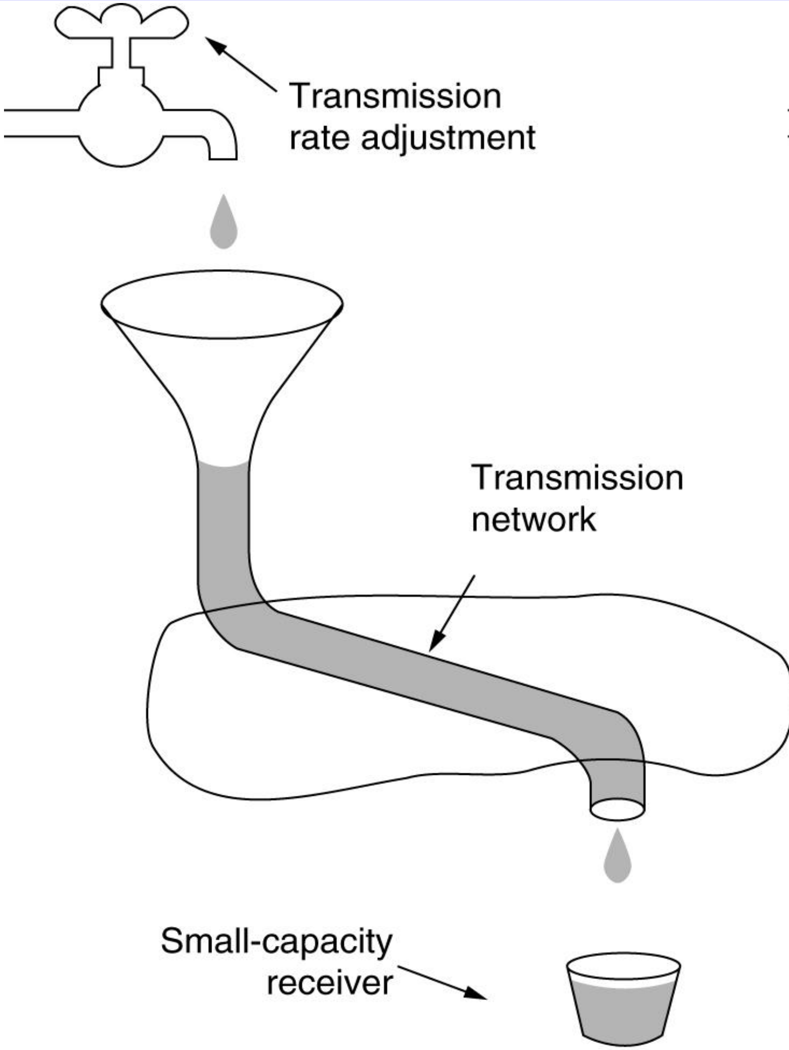
TCP Congestion Control 拥塞控制

- **Congestion:** **informally: “too many sources sending too much data too fast for *network* to handle”**
- **Goal of Congestion Control : limit senders as needed to ensure load on the network is “reasonable”**

Congestion Control拥塞控制 & Flow Control流量控制



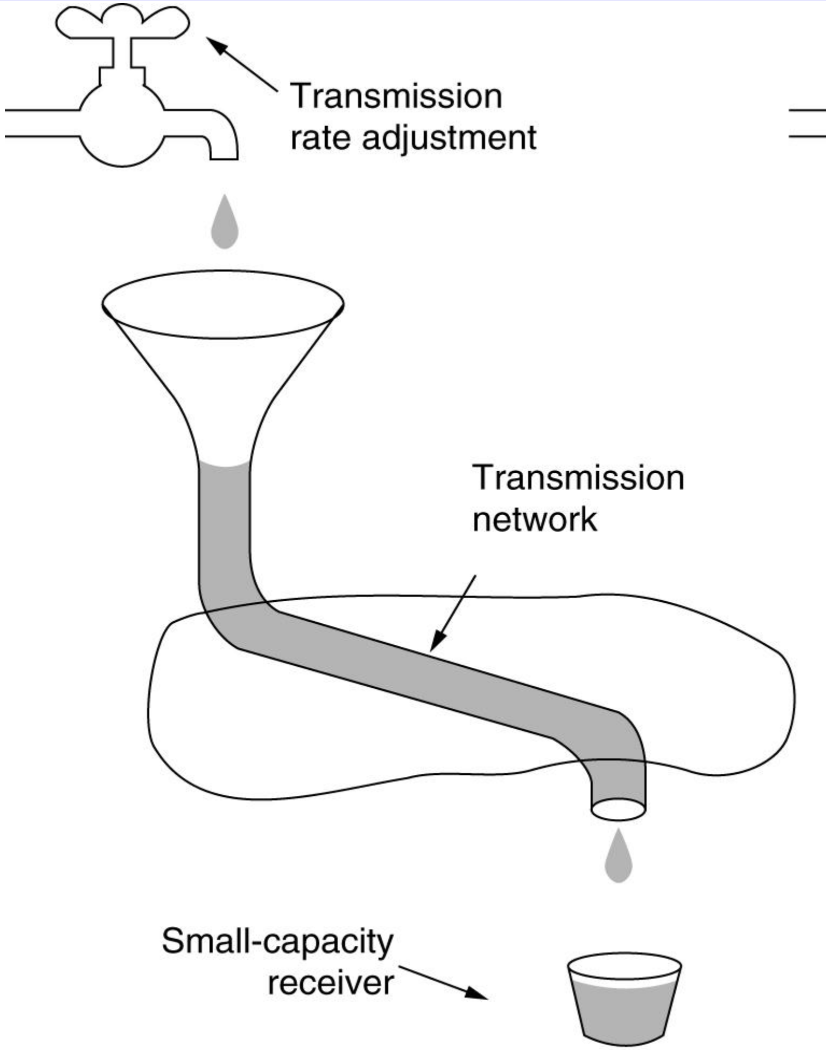
Congestion Control拥塞控制 & Flow Control流量控制



(a)

Flow control relates to the point-to-point traffic between a given sender and a given receiver. Its job is to make sure that a fast sender cannot continually transmit data faster than the receiver is able to absorb it.

Congestion Control拥塞控制 & Flow Control流量控制

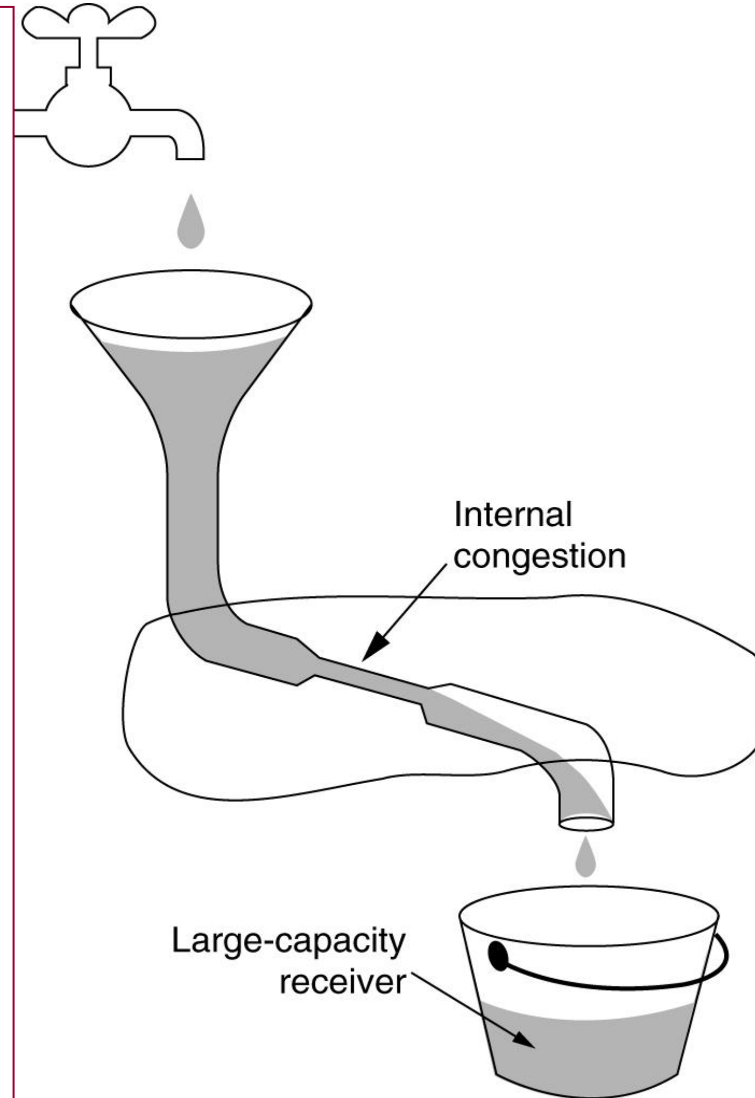


(a)

Flow control frequently involves some direct feedback from the receiver to the sender to tell the sender how things are doing at the other end.

Congestion Control拥塞控制 & Flow Control流量控制

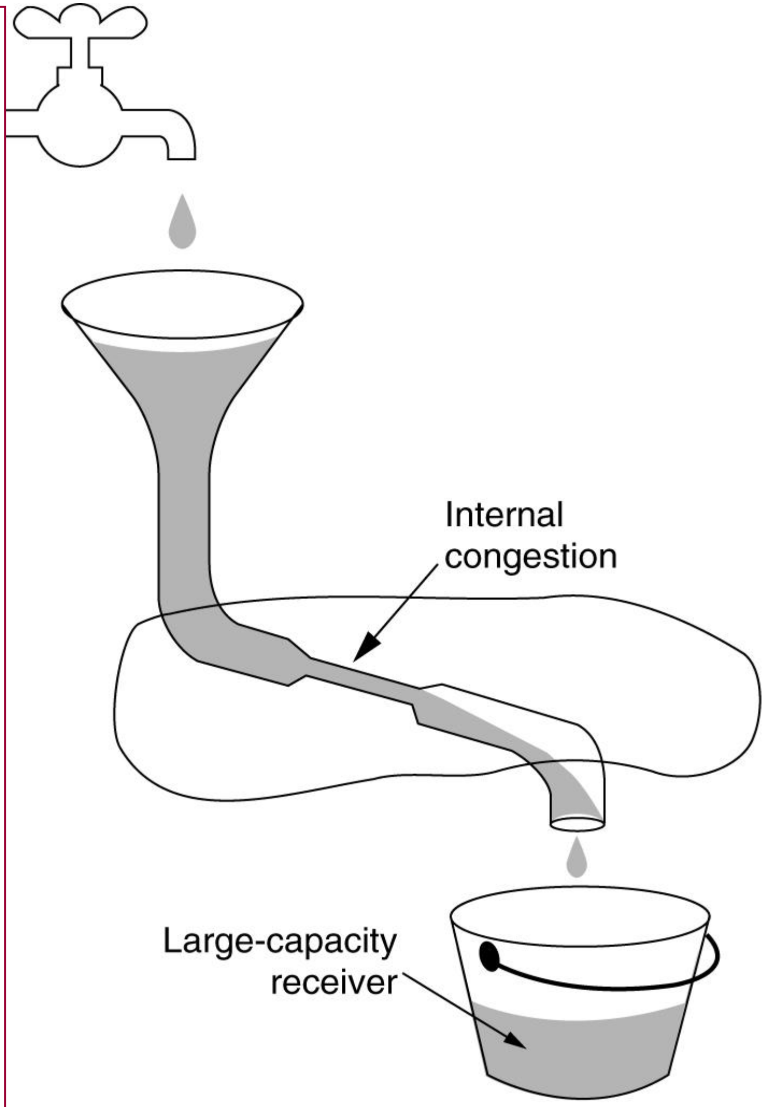
Congestion control has to do with making sure the subnet is able to carry the offered traffic.



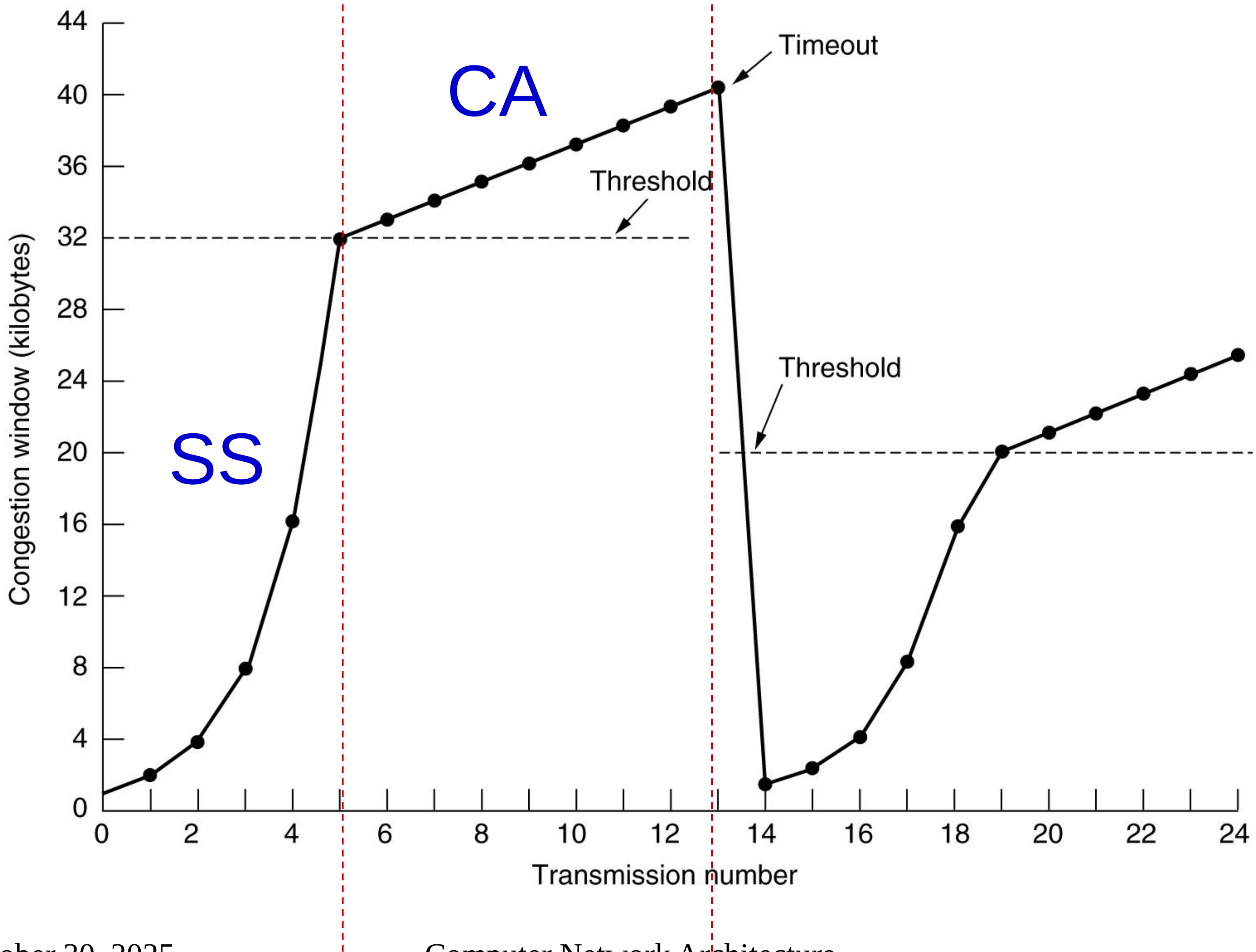
(b)

Congestion Control拥塞控制 & Flow Control流量控制

It is a global issue, involving the behavior of all the hosts, all the routers, the store-and-forwarding processing within the routers, and all the other factors that tend to diminish the carrying capacity of the subnet.



(b)



CA
SS
Timeout
Threshold
Threshold
Congestion window (kilobytes)
Transmission number
0
4
8
12
16
20
24
28
32
36
40
44
0
2
4
6
8
10
12
14
16
18
20
22
24

Background: Congestion control

- **In 1988, Van JACOBSON proposed first congestion control algorithm***
- **Since then, many new versions: Tahoe, Reno, New-Reno, SACK, Vegas, …**

***Van JACOBSON, "*Congestion Avoidance and Control*", Proceedings of ACM SIGCOMM, pp. 314-329, Stanford, CA, USA, 1988.**

General Principles of Congestion Control

Two types of solutions:

- **Open loop开环: try to solve the problem by good design, to make sure it never occurs.**
- **Closed loop闭环: is based on the concept of a feedback loop反馈环路.**

Closed loop Congestion Control

1. **Monitor the system.**
   - **Detect when and where congestion occurs.探测什么时间, 什么地方发生拥塞。**
2. **Pass information to where action can be taken.**
3. **Adjust调整 system operation to correct the problem.**

Indication of Network Congestion

- **Packet drops due to lack of buffer space**
- **average queue lengths** ↗
- **the number of packets that time out and are retransmitted** ↗
- **the average packet delay** ↗
- ……

Random Early Detection (RED)*
随机早期预测

- **Early random drop随机早期丢弃**
  - **rather than wait for queue to become full, drop each arriving packet with some *drop probability* whenever the queue length exceeds some *drop level***

S.Floyd, V.Jacobson, “Random early detection gateways for congestion avoidance”, IEEE/ACM Transactions on Networking, pp. 397-413. August 1993.

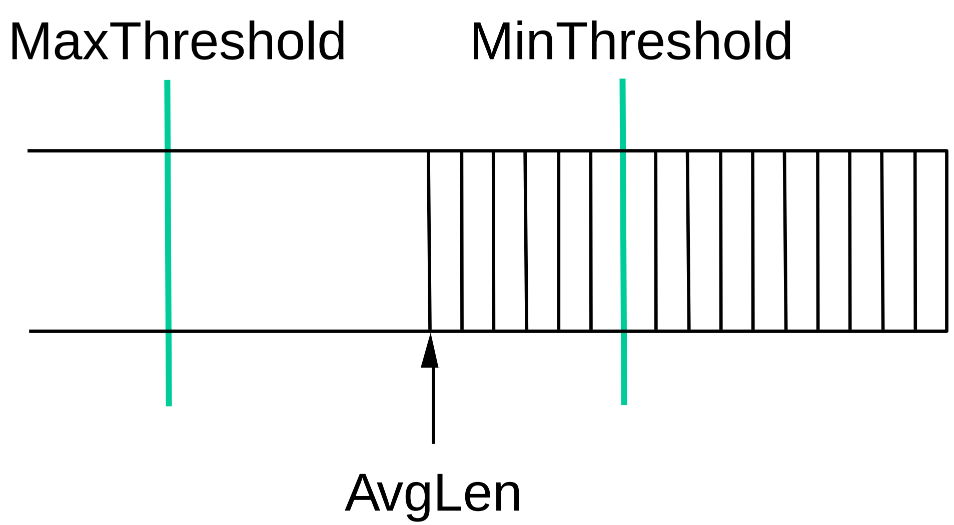
RED Details

- **Compute average queue length**

```
AvgLen = (1-Weight) * AvgLen +
                Weight * SampleLen
```

0 < `Weight` < 1 (usually 0.002)

`SampleLen` is queue length each time a packet arrives

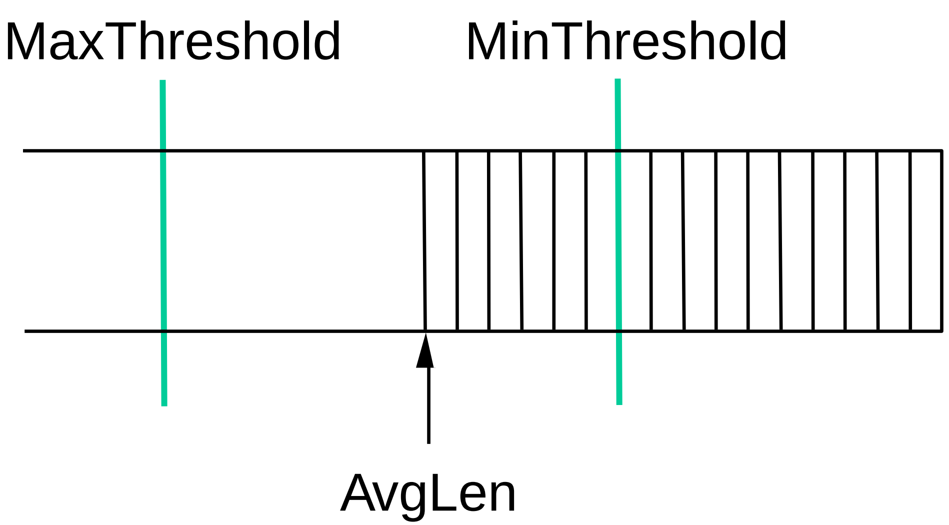


RED Details

- **Two queue length thresholds**

  if AvgLen <= MinThreshold then

  enqueue the packet



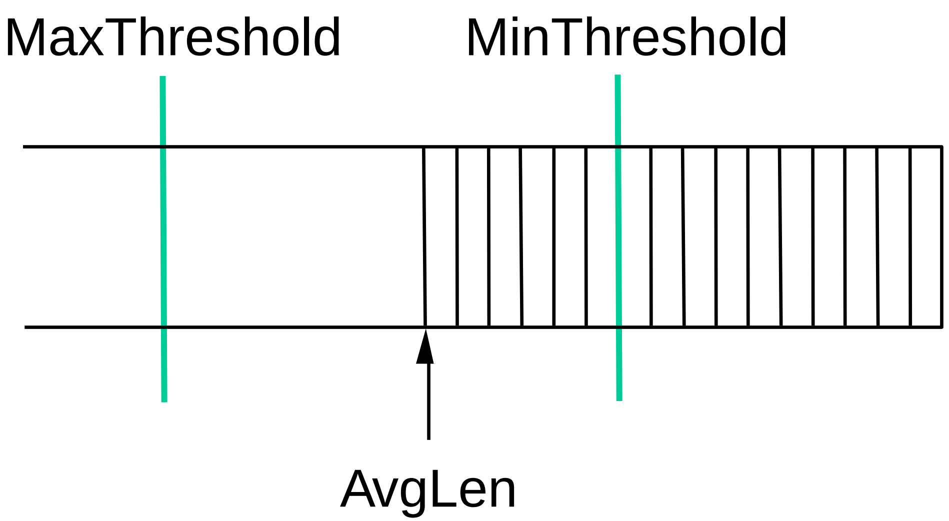
RED Details

- **Two queue length thresholds**

  if MinThreshold < AvgLen < MaxThreshold then

  calculate probability P

  drop arriving packet with probability P

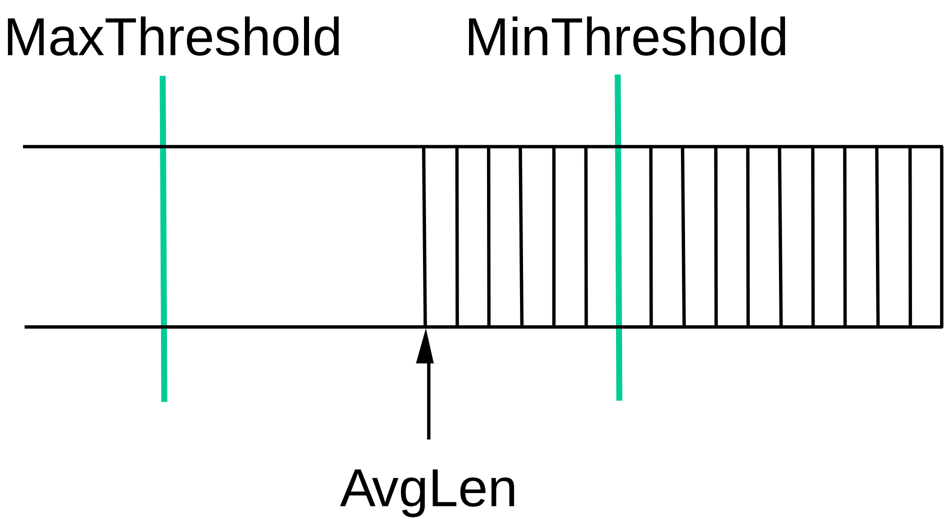


RED Details

- **Two queue length thresholds**

  if MaxThreshold <= AvgLen then

  drop arriving packet



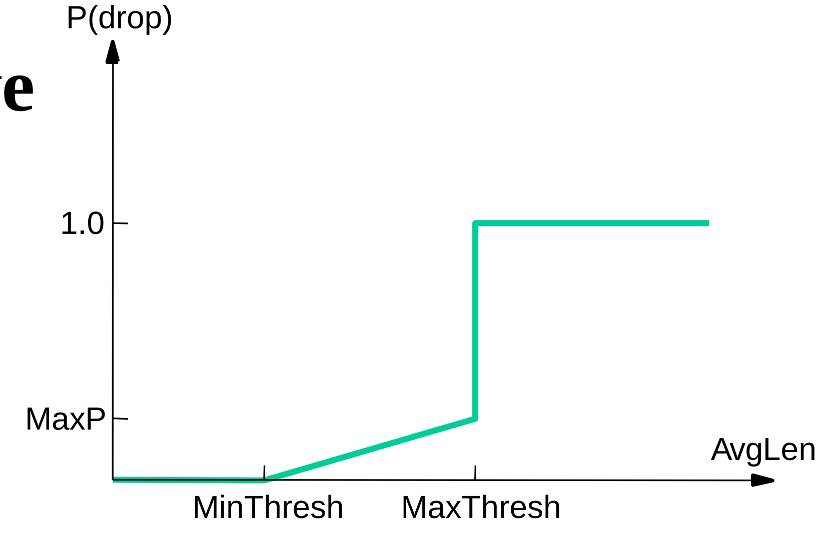
RED Details (cont)

a) Computing probability P

TempP = MaxP * (AvgLen - MinThreshold)/ (MaxThreshold - MinThreshold)

P = TempP/(1 - count * TempP)

b) Drop Probability Curve

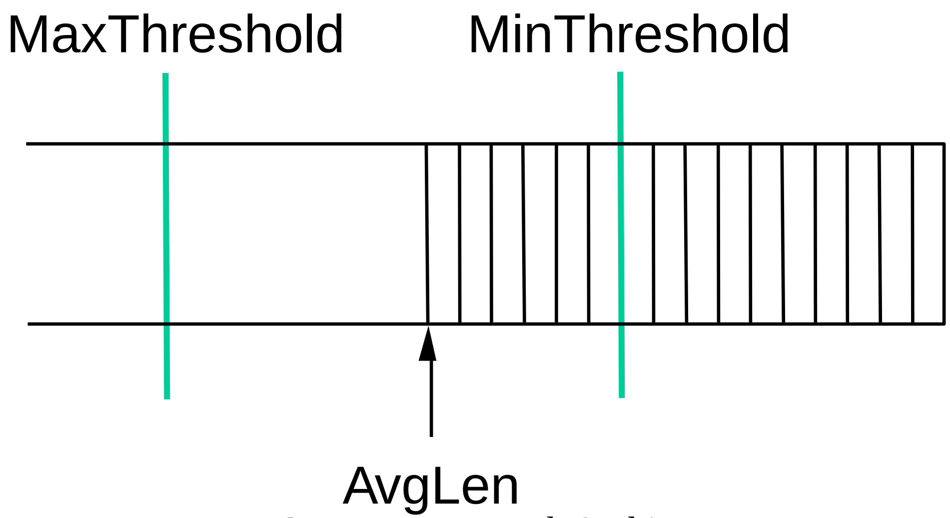


Tuning RED

- **Probability of dropping a particular flow's packet(s) is roughly proportional to the share of the bandwidth that flow is currently getting**
- `MaxP` **is typically set to 0.02, meaning that when the average queue size is halfway between the two thresholds, the router drops roughly one out of 50 packets.**

Tuning RED

- **If traffic is bursty, then `MinThreshold` should be sufficiently large to allow link utilization to be maintained at an acceptably high level.**

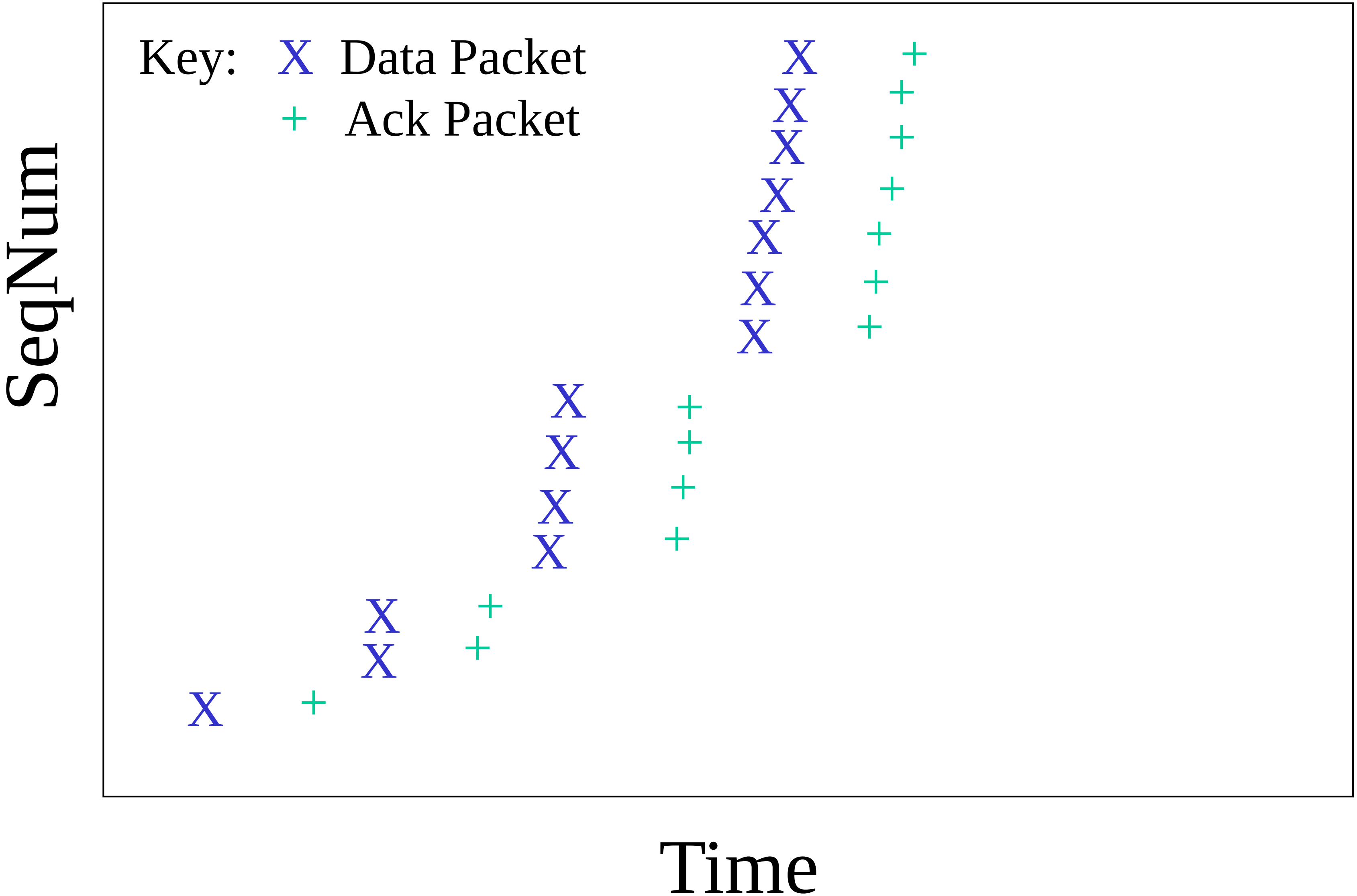


Tuning RED

- **Difference between two thresholds should be larger than the typical increase in the calculated average queue length in one RTT;**
- **setting `MaxThreshold` to twice `MinThreshold` is reasonable for traffic on today's Internet.**

TCP Sequence Number Plot

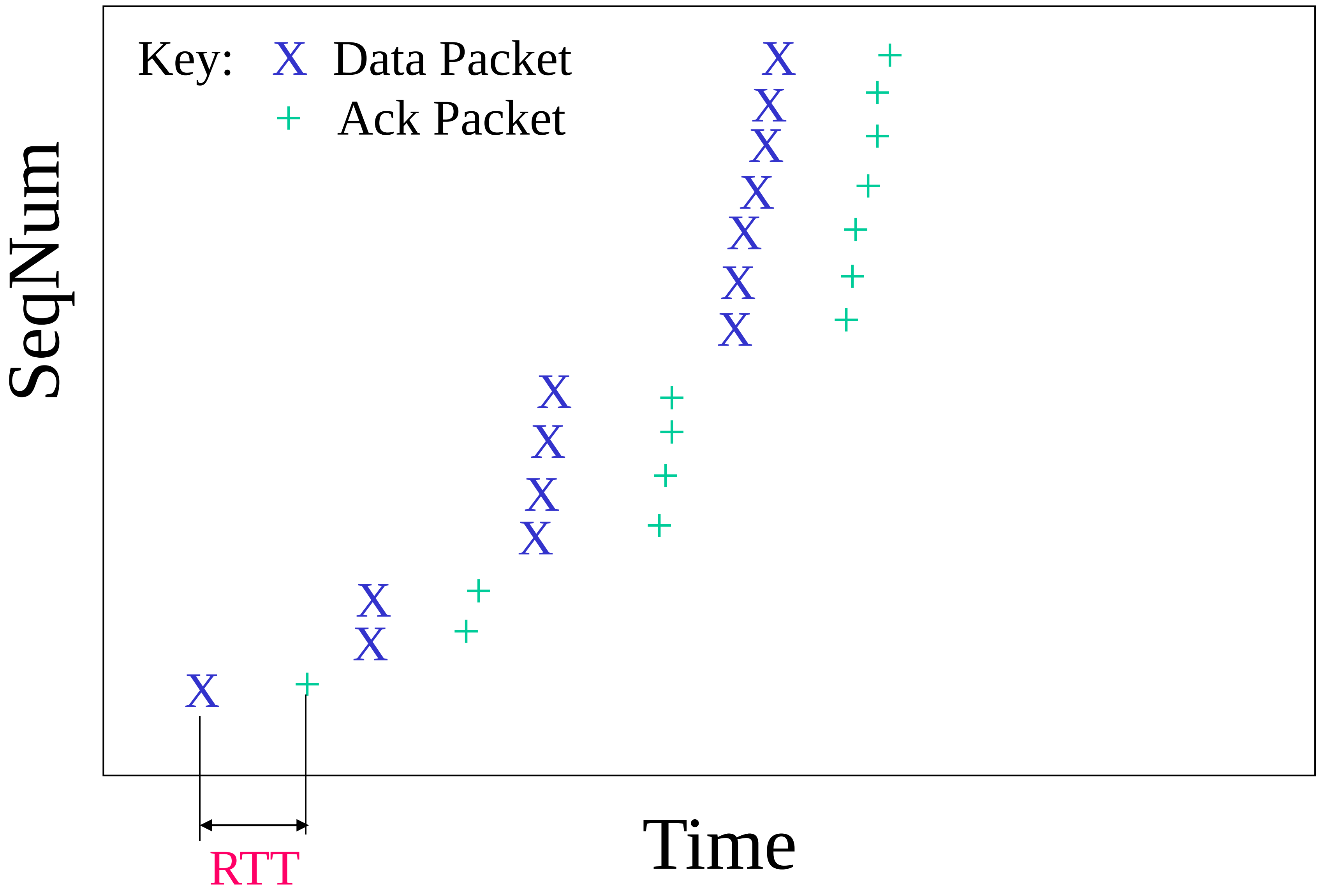
- There is a beautiful way to plot and visualize the dynamics of TCP behaviour
- Plot packet events (data and acks) as points in 2-D space, with time on the horizontal axis, and sequence number on the vertical axis
- Example: Consider a 14-packet transfer

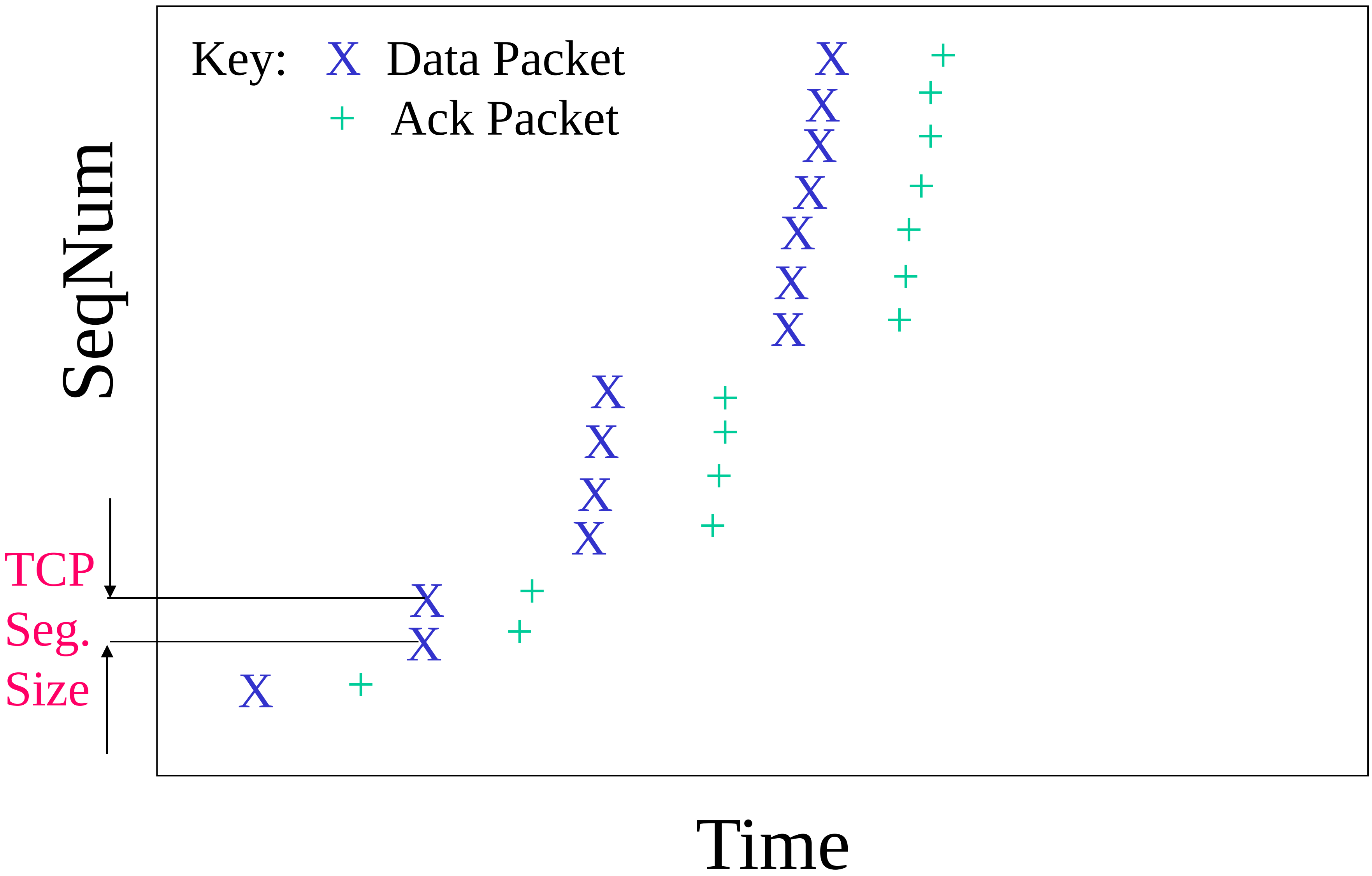

Key: X Data Packet
+ Ack Packet
SeqNum
Time

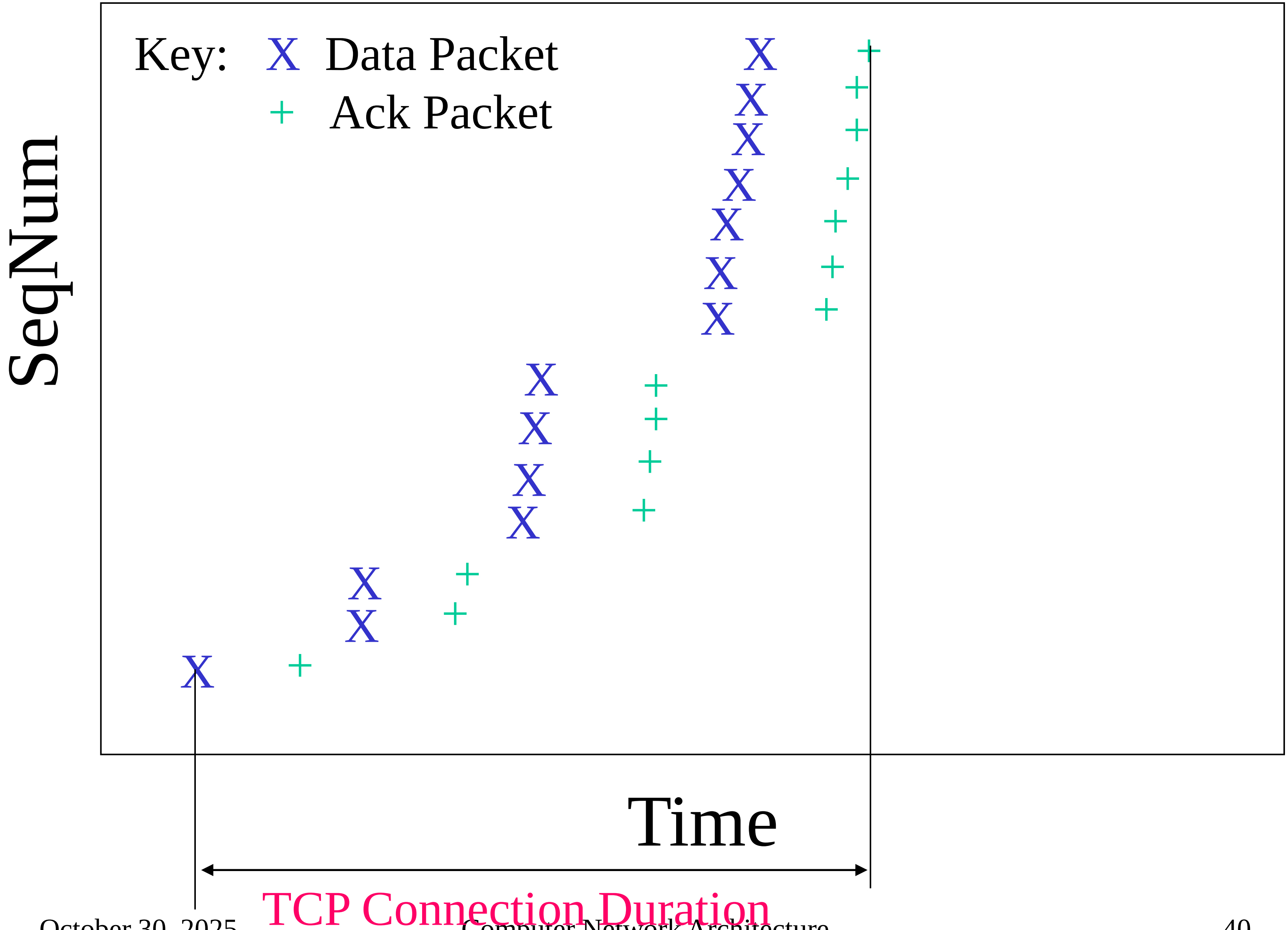
So What?

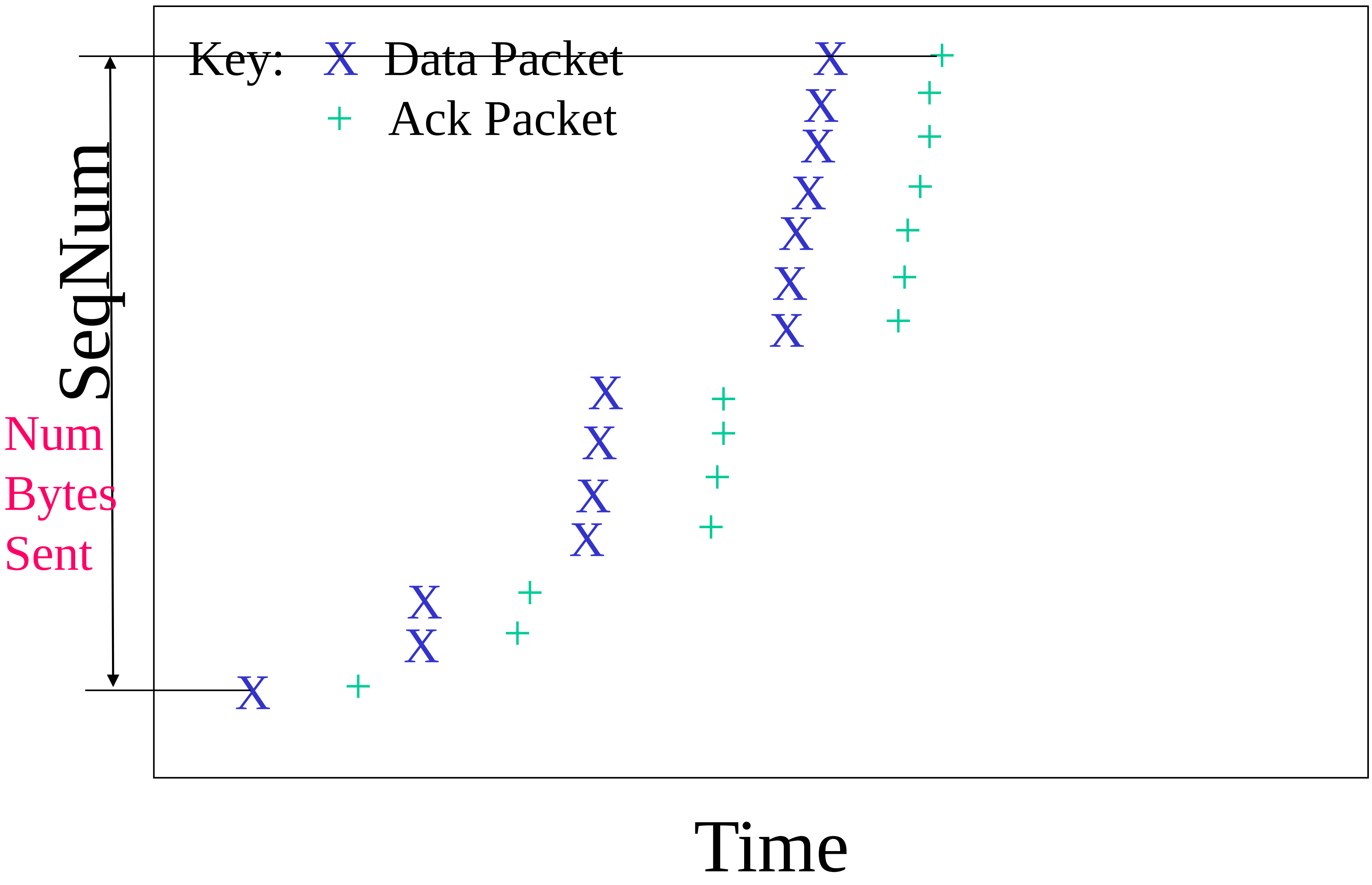
- **What can it tell you?**

- **Everything!!!☺**

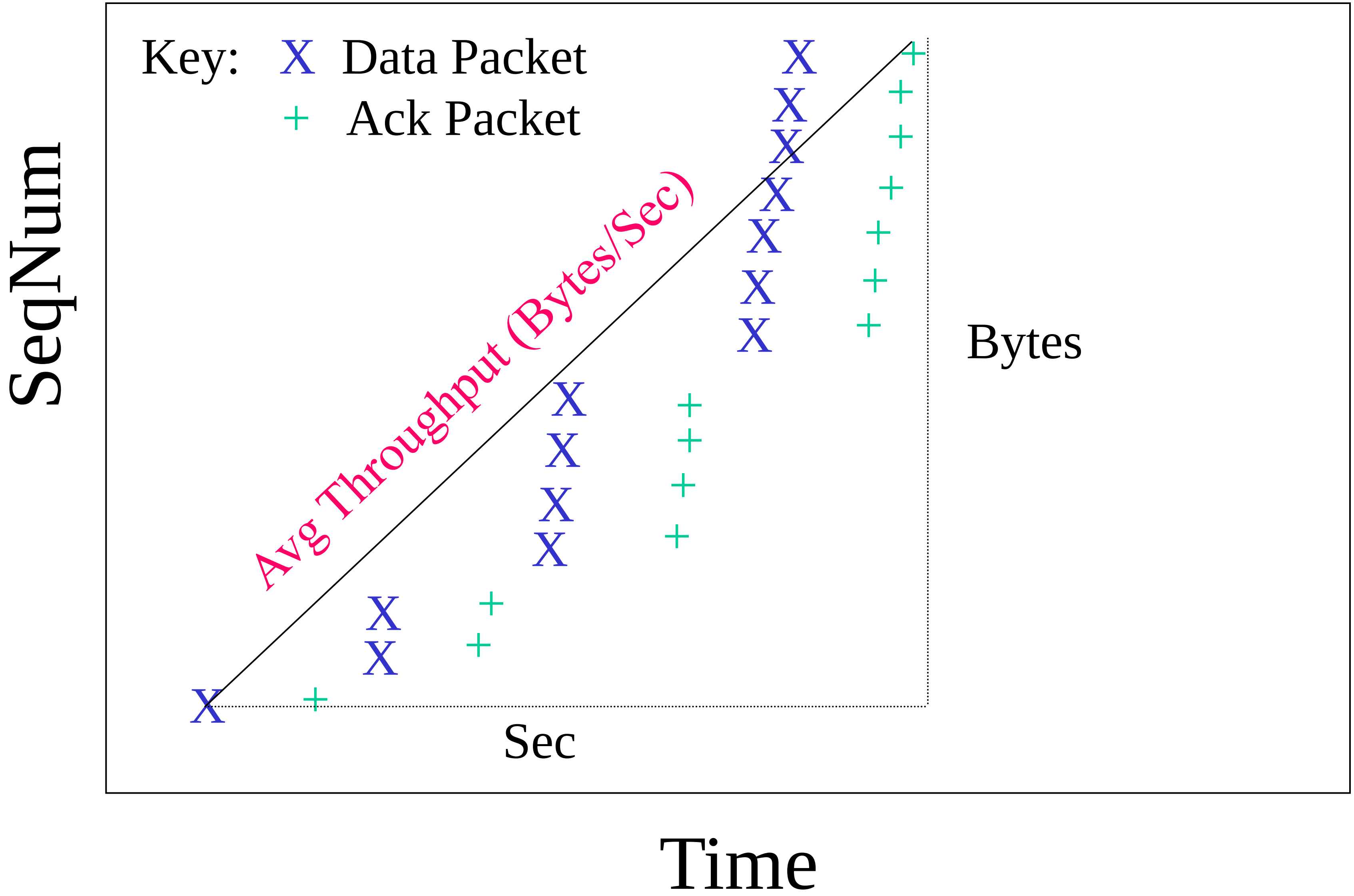

Key:
X Data Packet
+ Ack Packet
SeqNum
Time
RTT

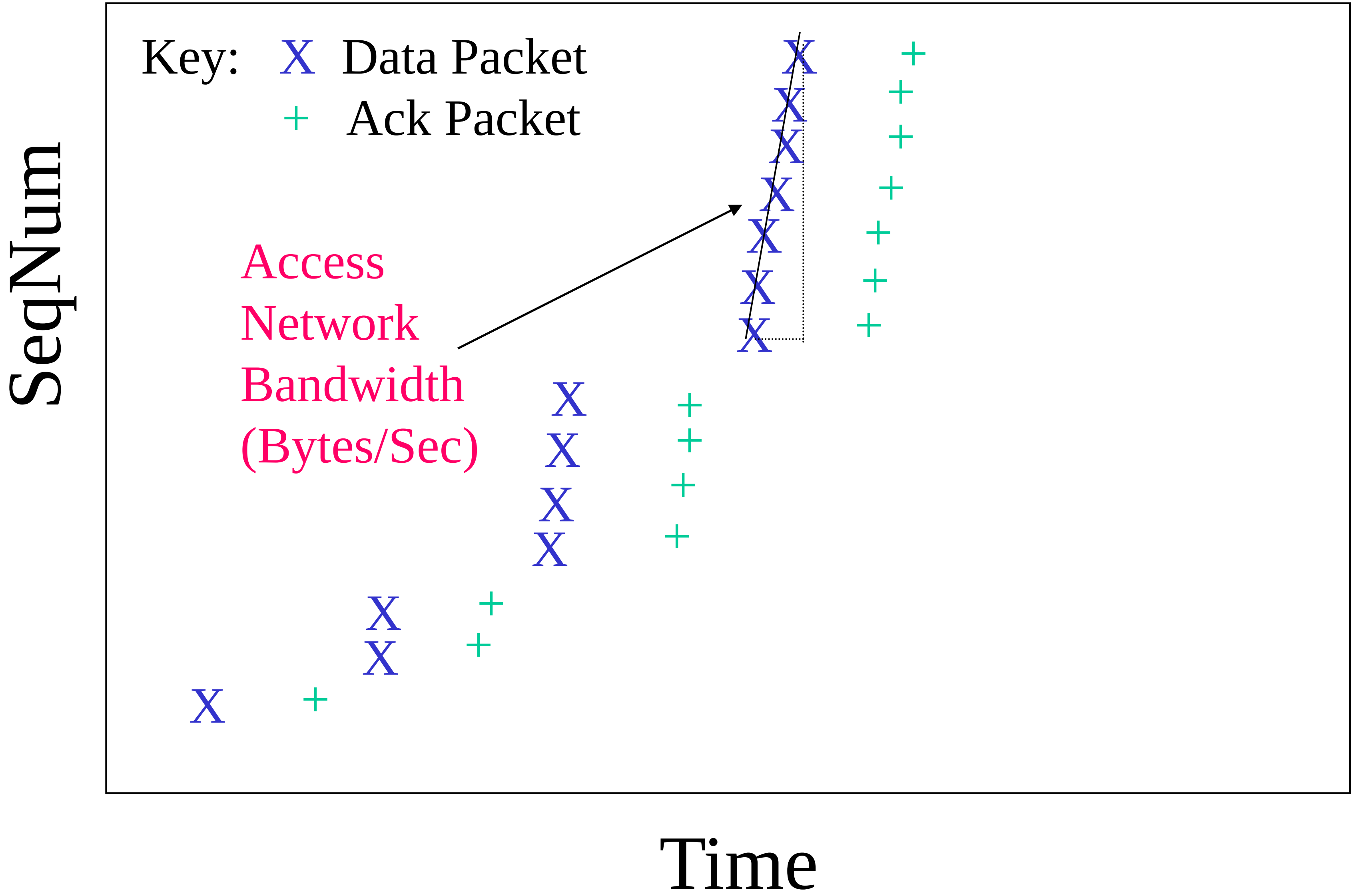

Key:
X Data Packet
+ Ack Packet
SeqNum
TCP
Seg.
Size
Time

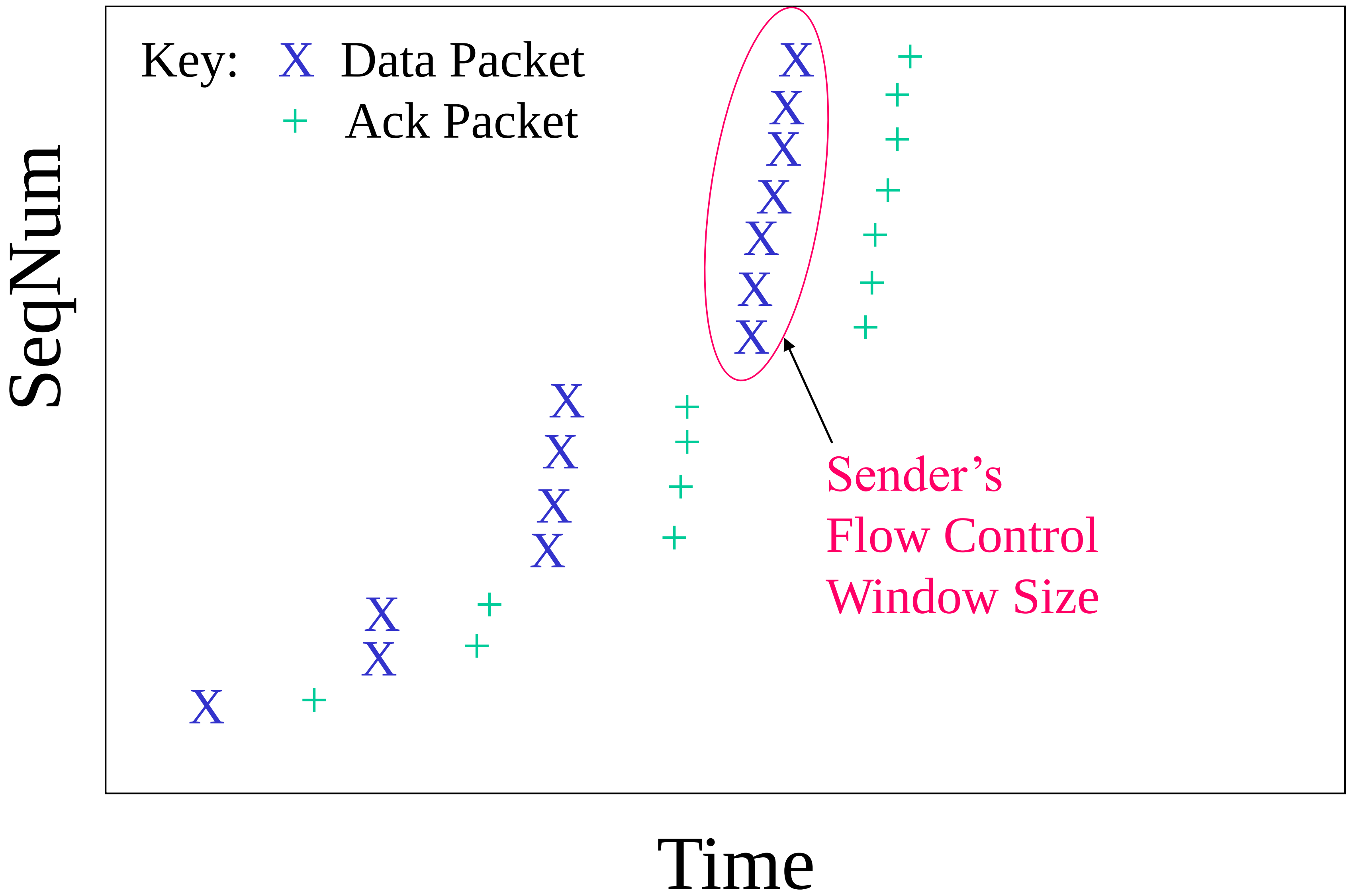

Key:
X Data Packet
+ Ack Packet
SeqNum
Time
TCP Connection Duration

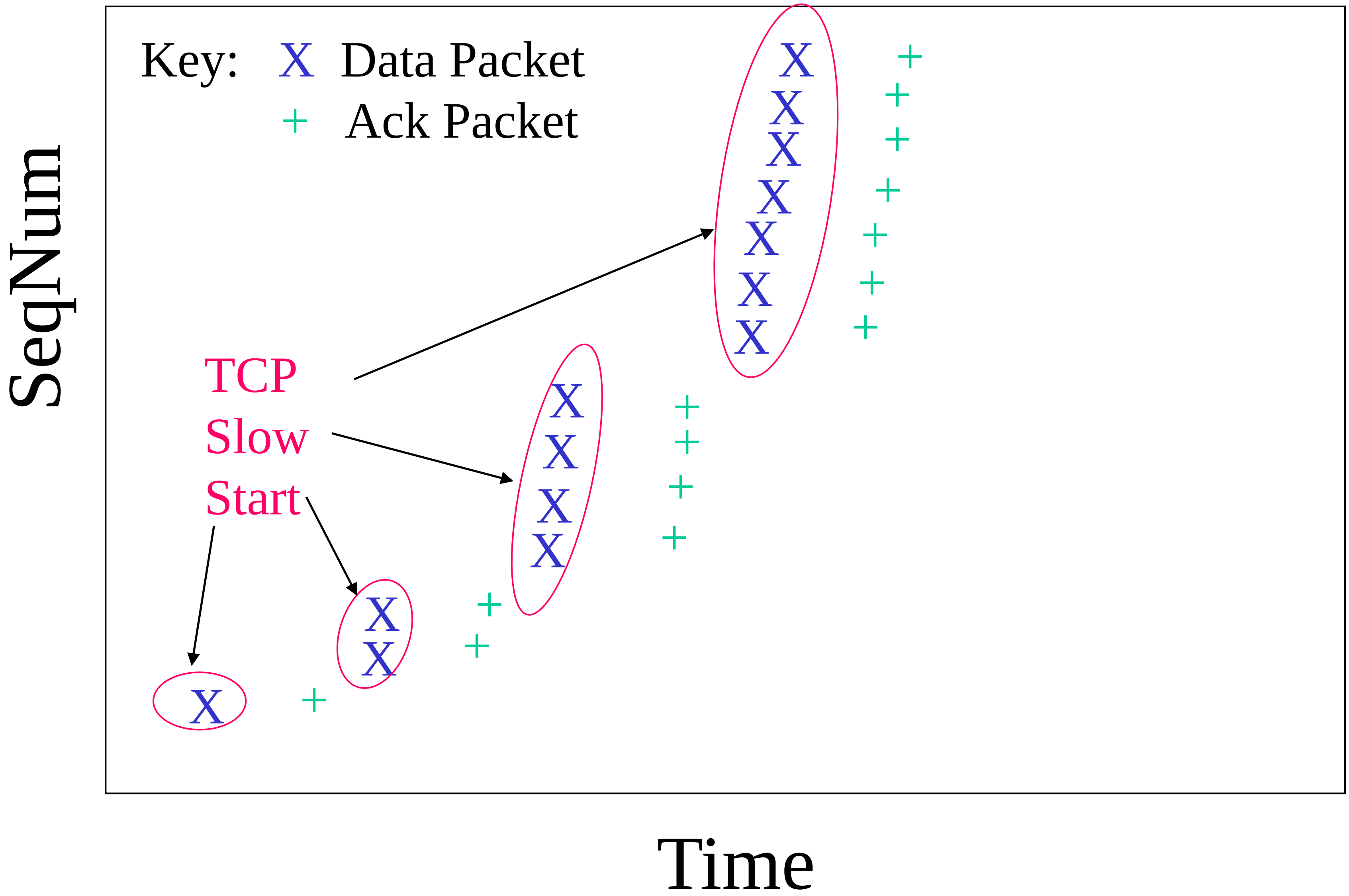


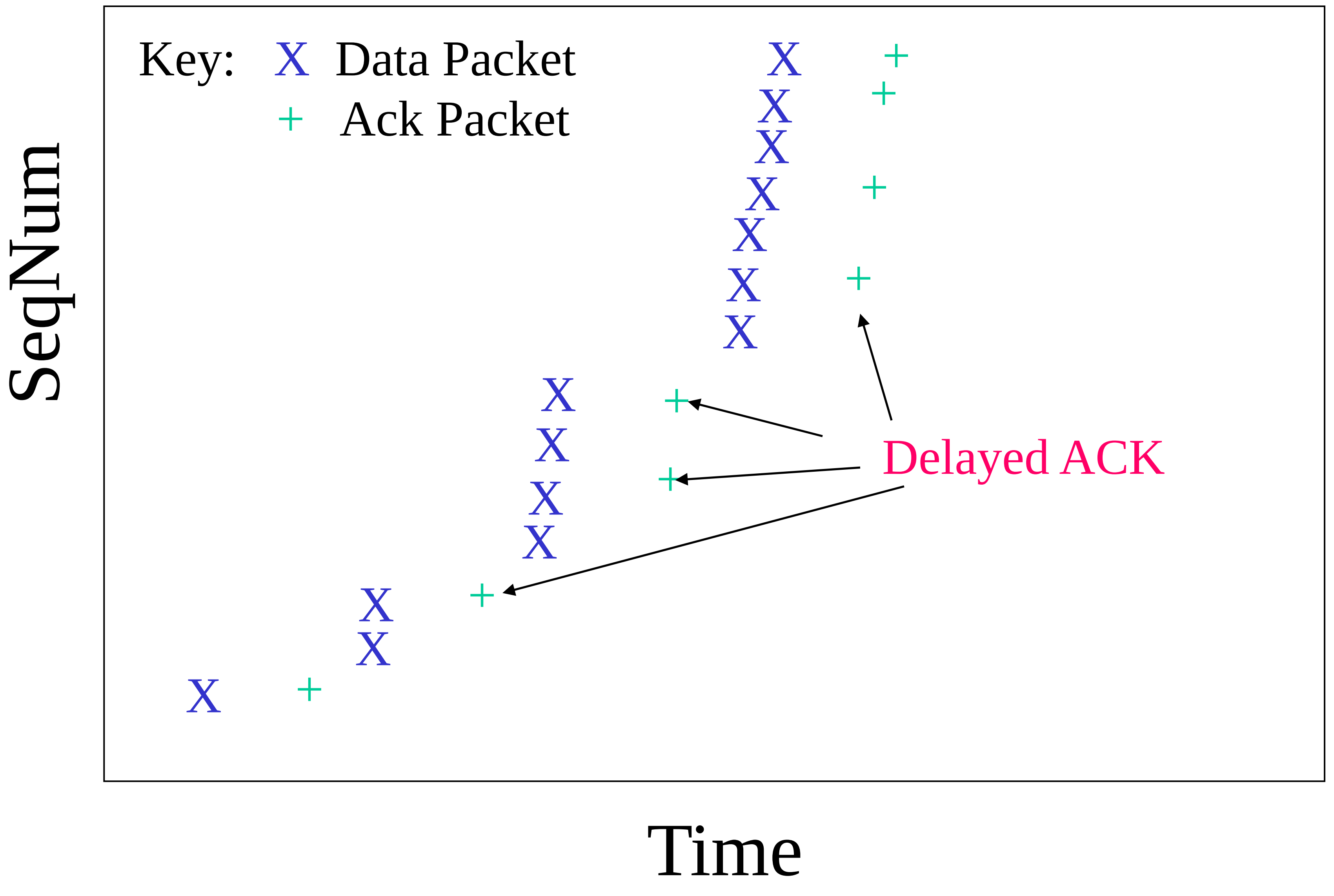
Key:
X Data Packet
+ Ack Packet
SeqNum
Num
Bytes
Sent
Time

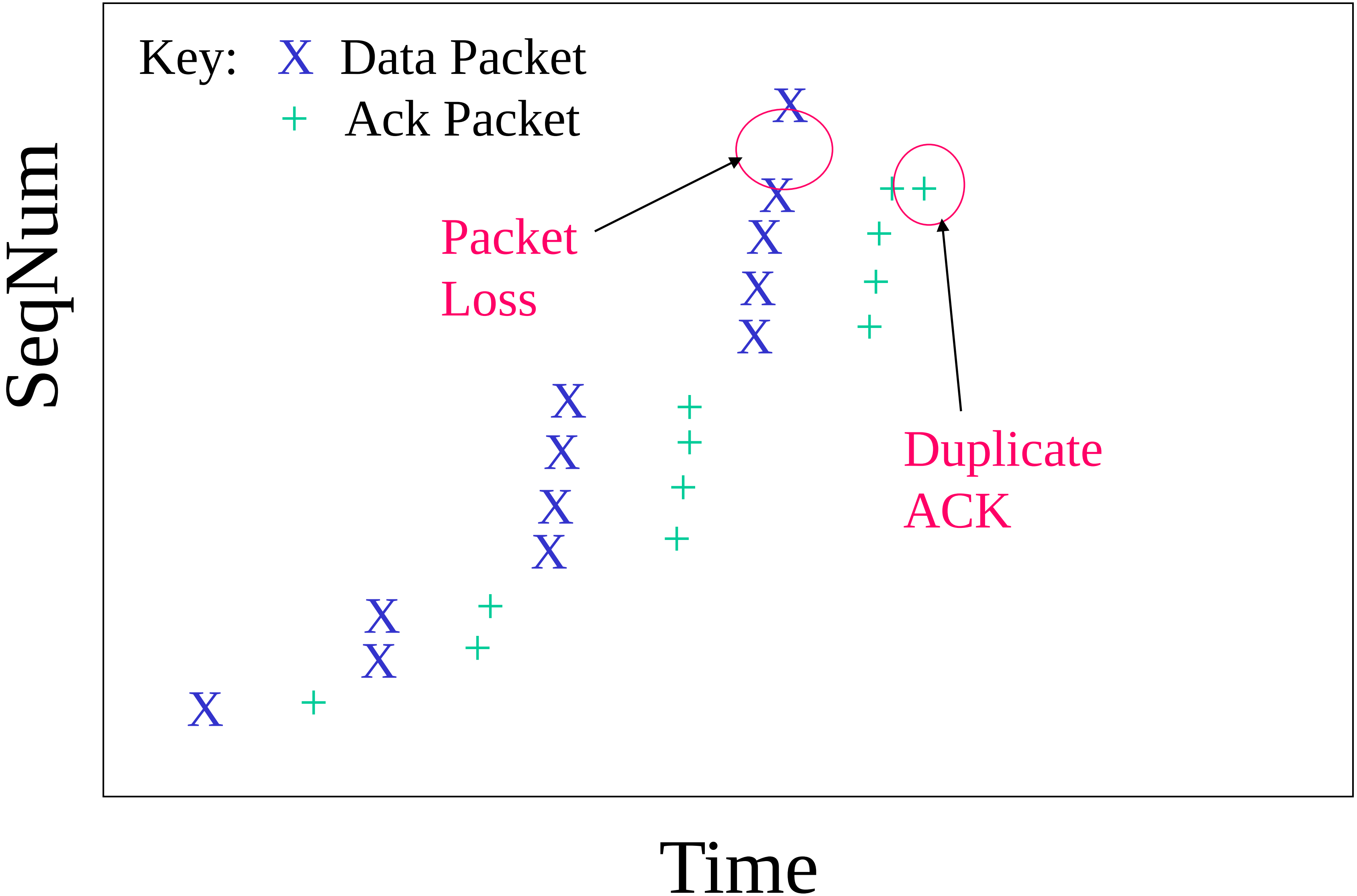

SeqNum
Key: X Data Packet
+ Ack Packet
Avg Throughput (Bytes/Sec)
Bytes
Sec
Time

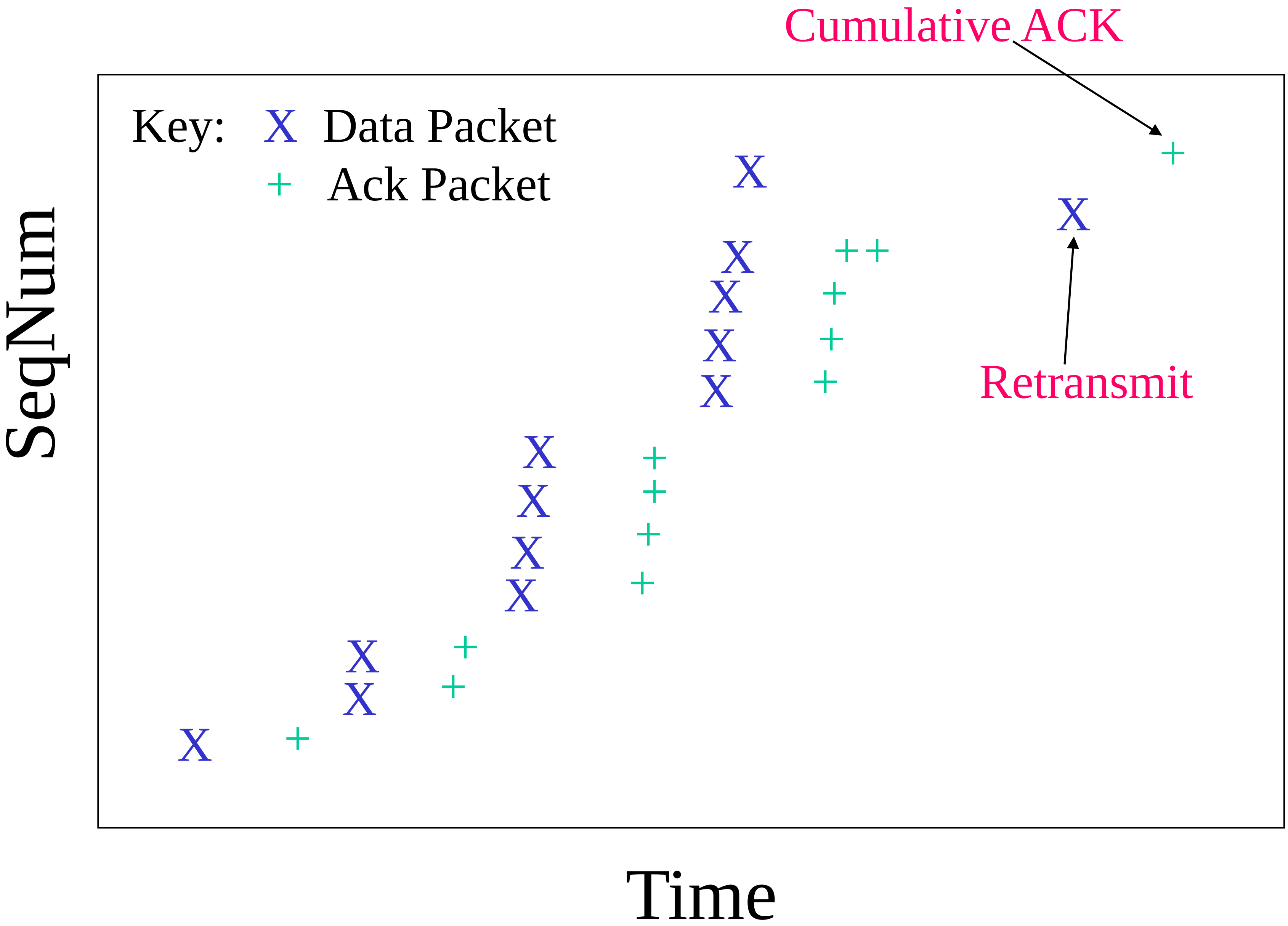

Key: X Data Packet
+ Ack Packet
Access
Network
Bandwidth
(Bytes/Sec)
SeqNum
Time

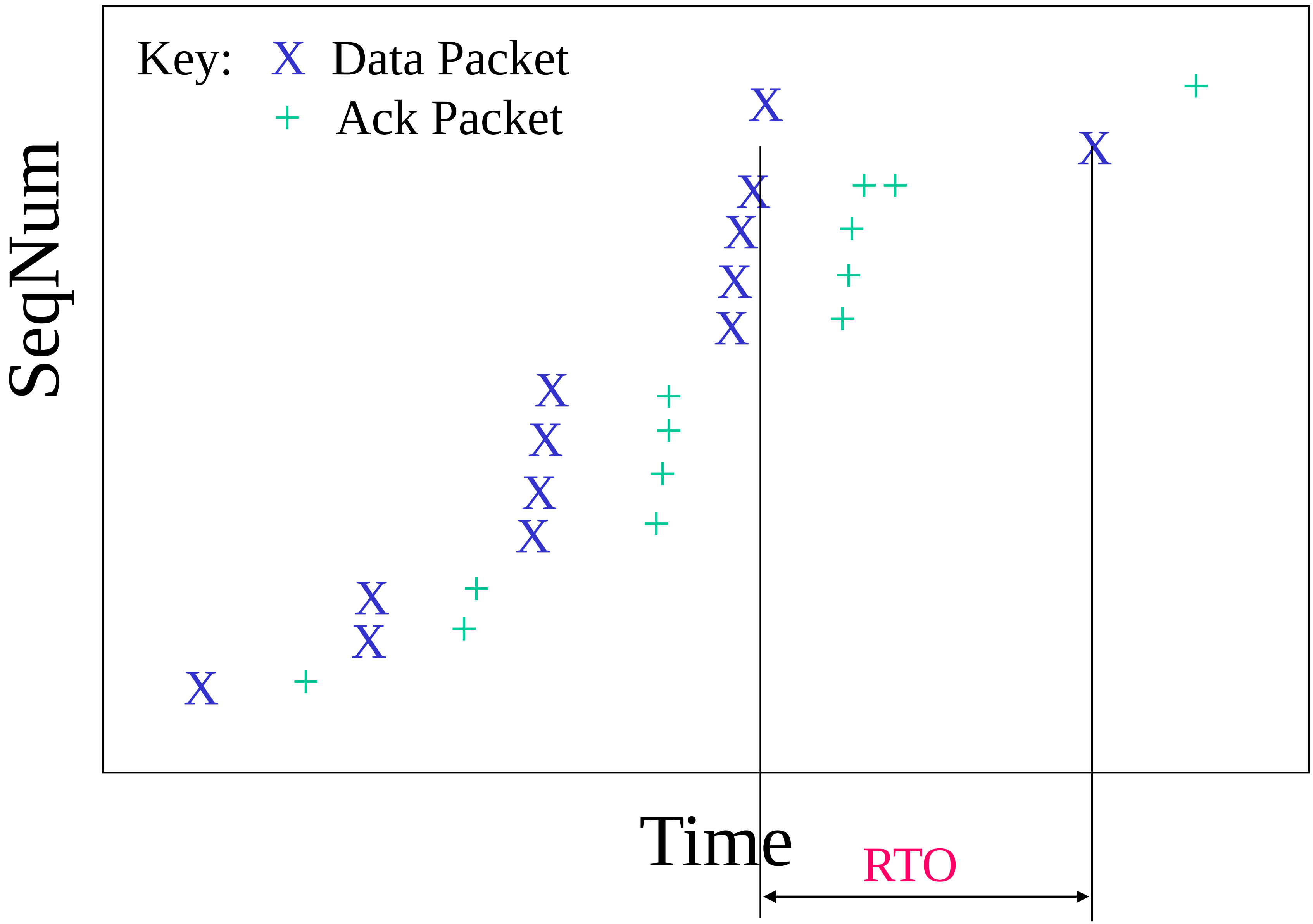

Key: X Data Packet
+ Ack Packet
Sender's
Flow Control
Window Size
SeqNum
Time


Key: X Data Packet
+ Ack Packet
SeqNum
TCP
Slow
Start
Time


Key:
X Data Packet
+ Ack Packet
SeqNum
Time
Delayed ACK

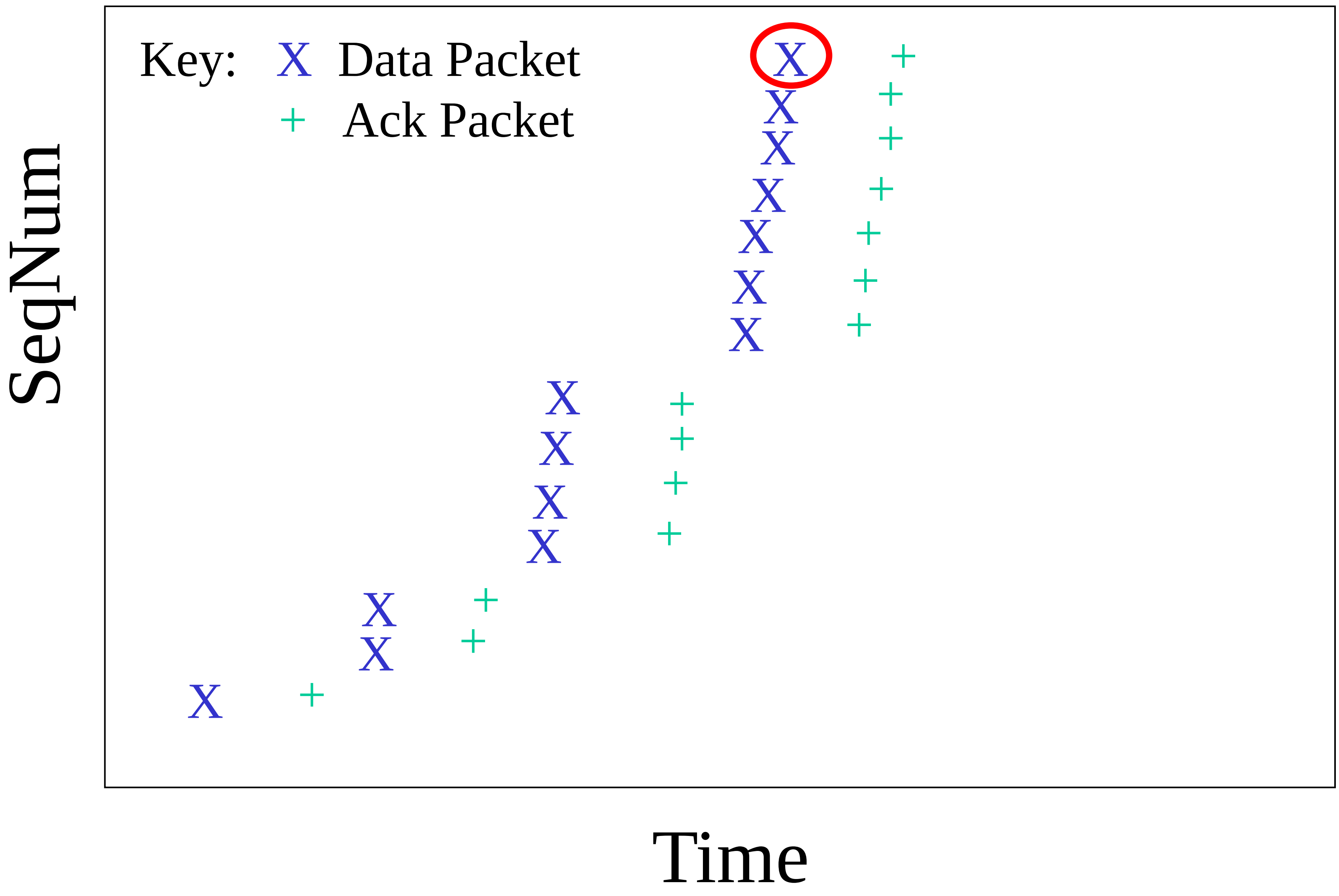

Key: X Data Packet
+ Ack Packet
Packet Loss
Duplicate ACK
SeqNum
Time

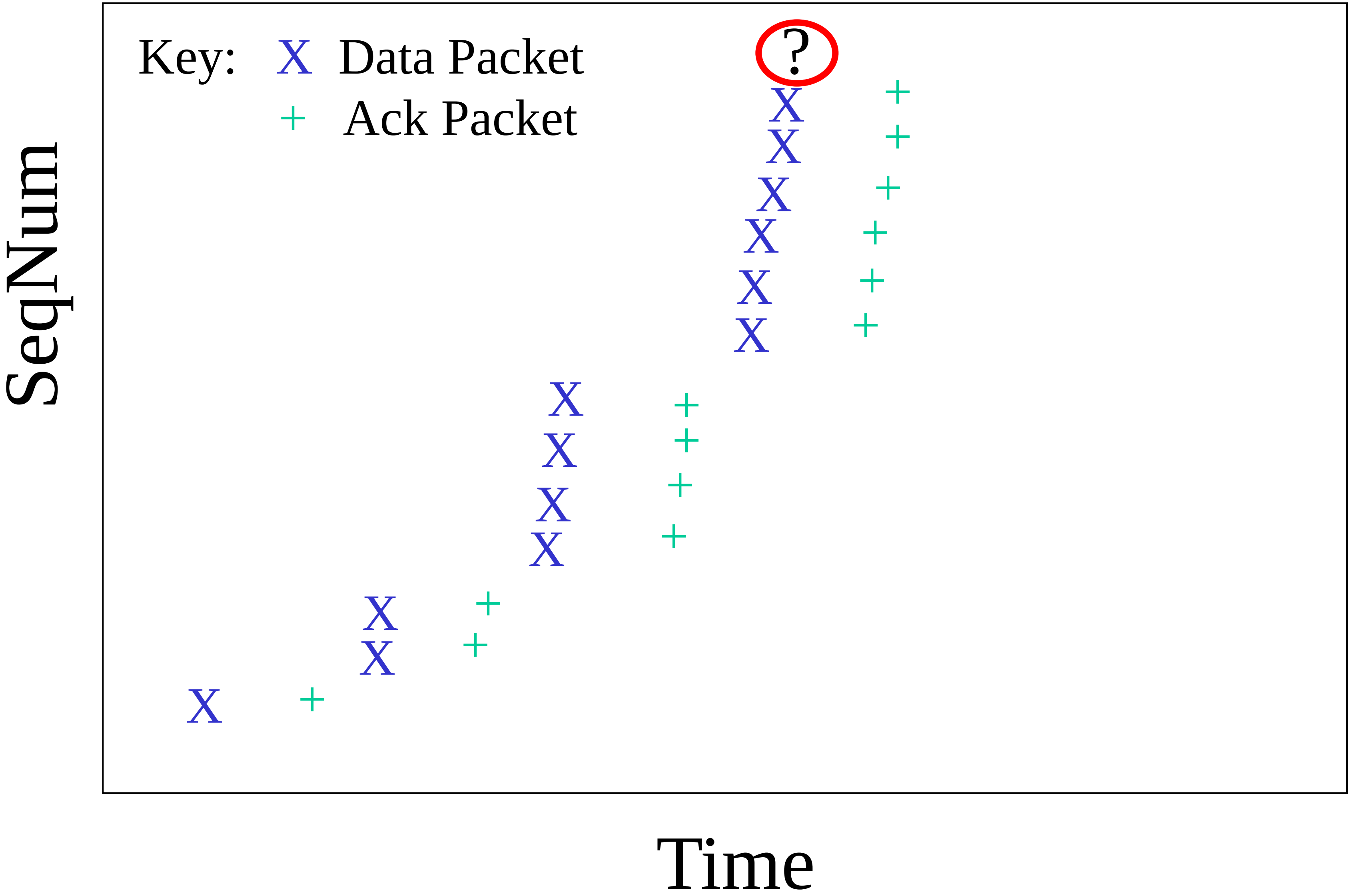

Cumulative ACK
Key:
X Data Packet
+ Ack Packet
Retransmit
SeqNum
Time

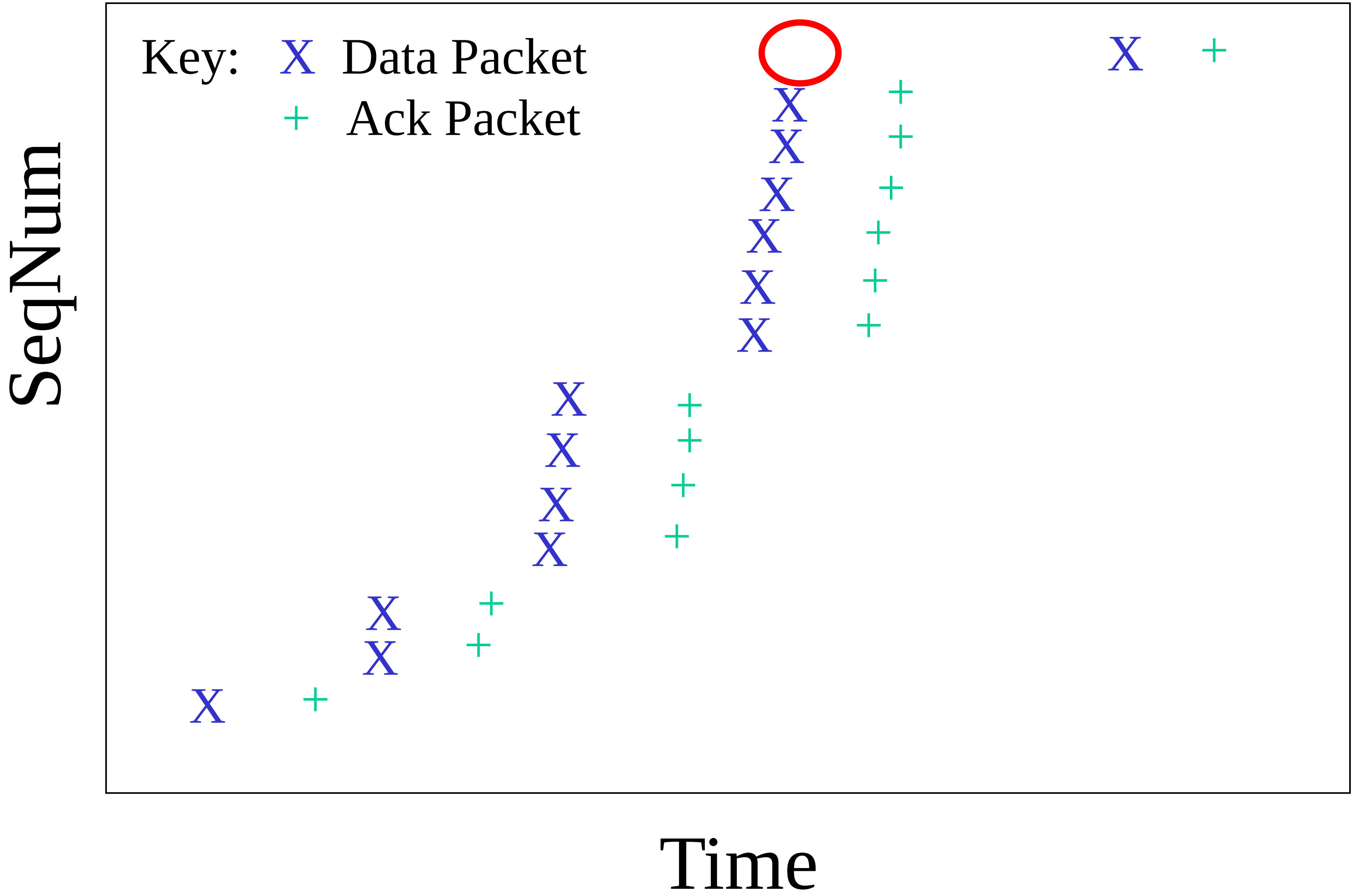

Key:
X Data Packet
+ Ack Packet
SeqNum
Time
RTO

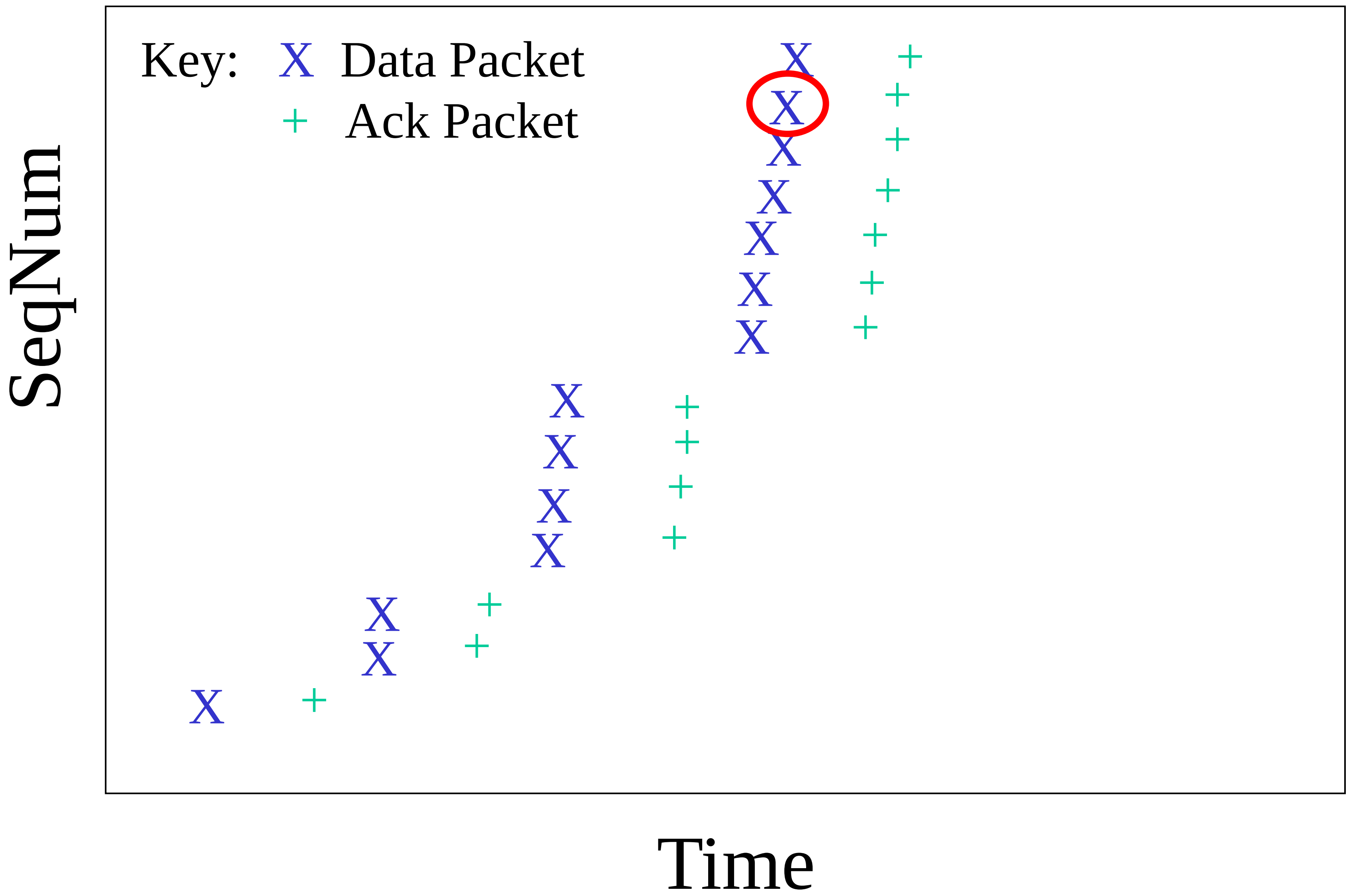
TCP 101 (Cont'd)

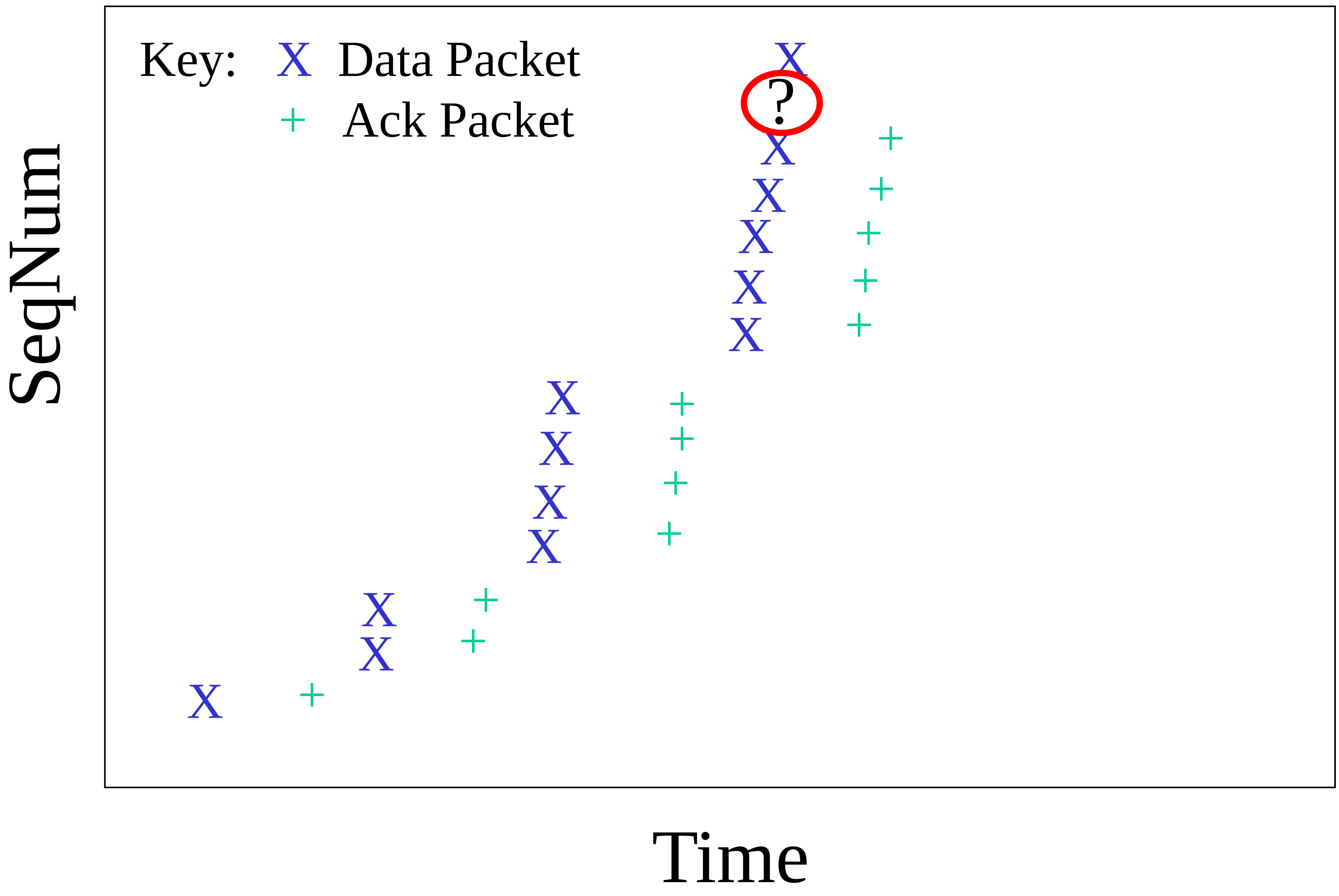
- **What happens when a packet loss occurs?**
- **Quiz Time...**
  - **Consider a 14-packet Web document**
  - **For simplicity, consider only a single packet loss**

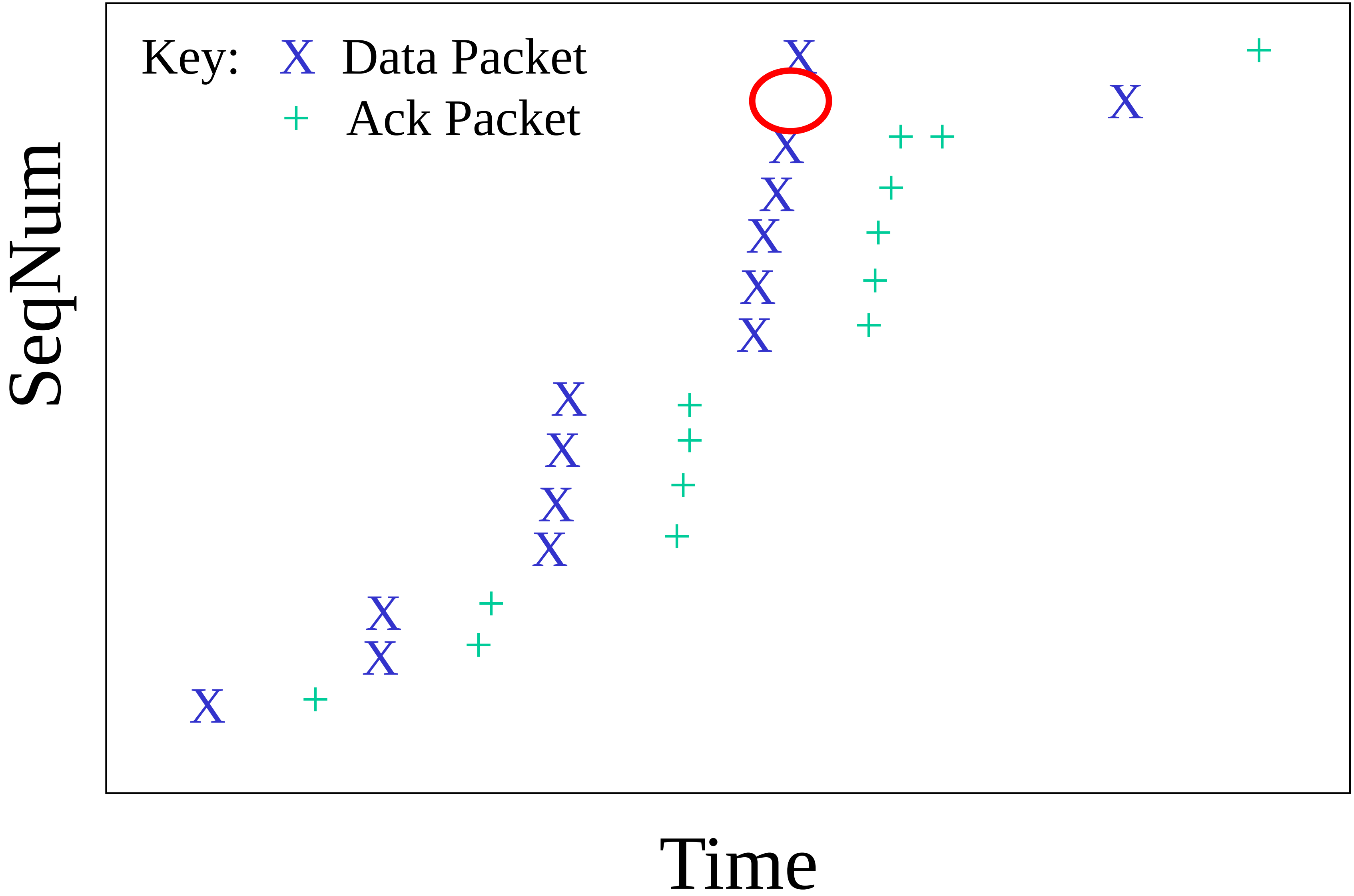

Key: X Data Packet
+ Ack Packet
SeqNum
Time

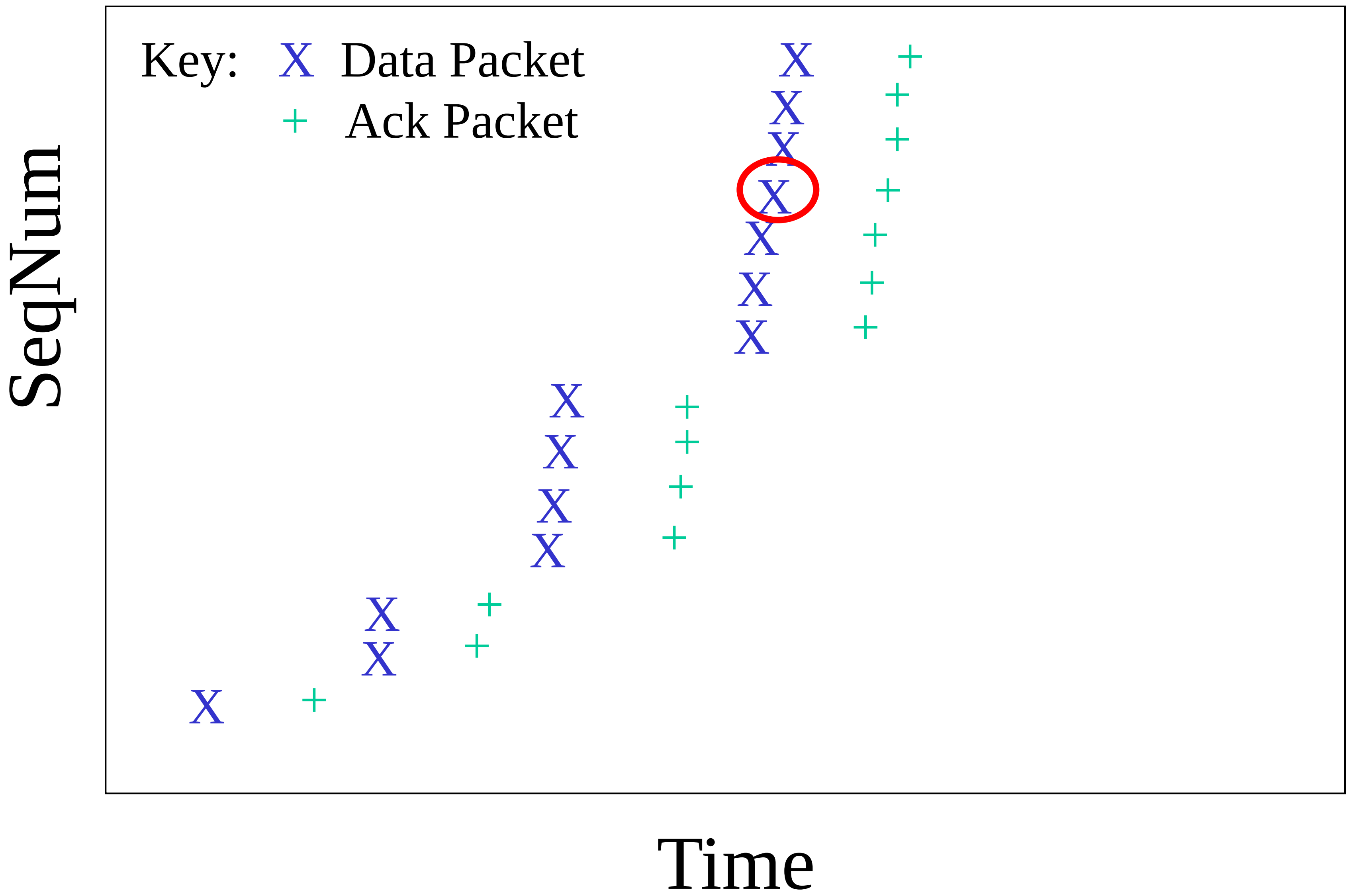

Key:
X Data Packet
+ Ack Packet
?
SeqNum
Time

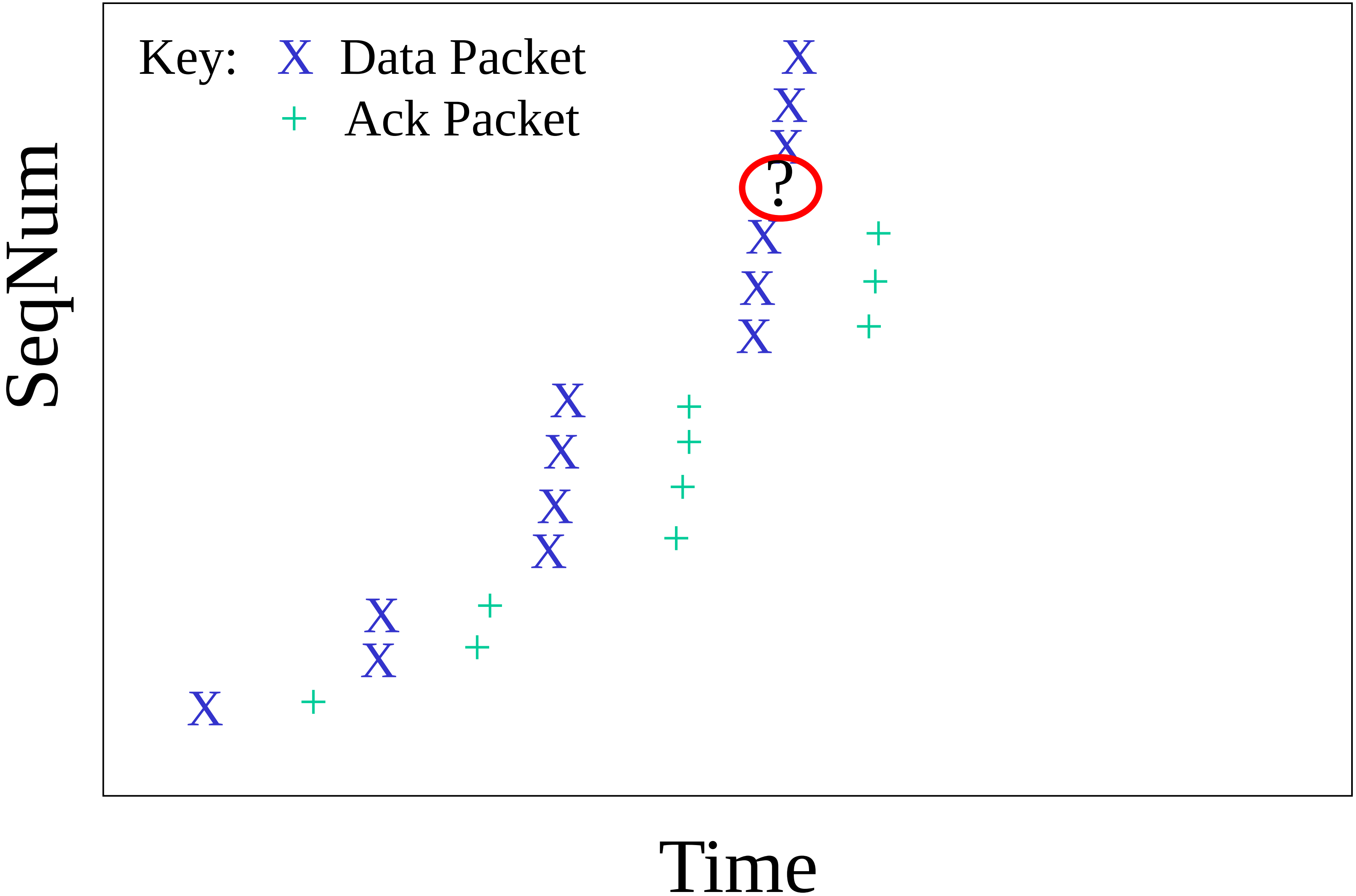

Key:
X Data Packet
+ Ack Packet
SeqNum
Time

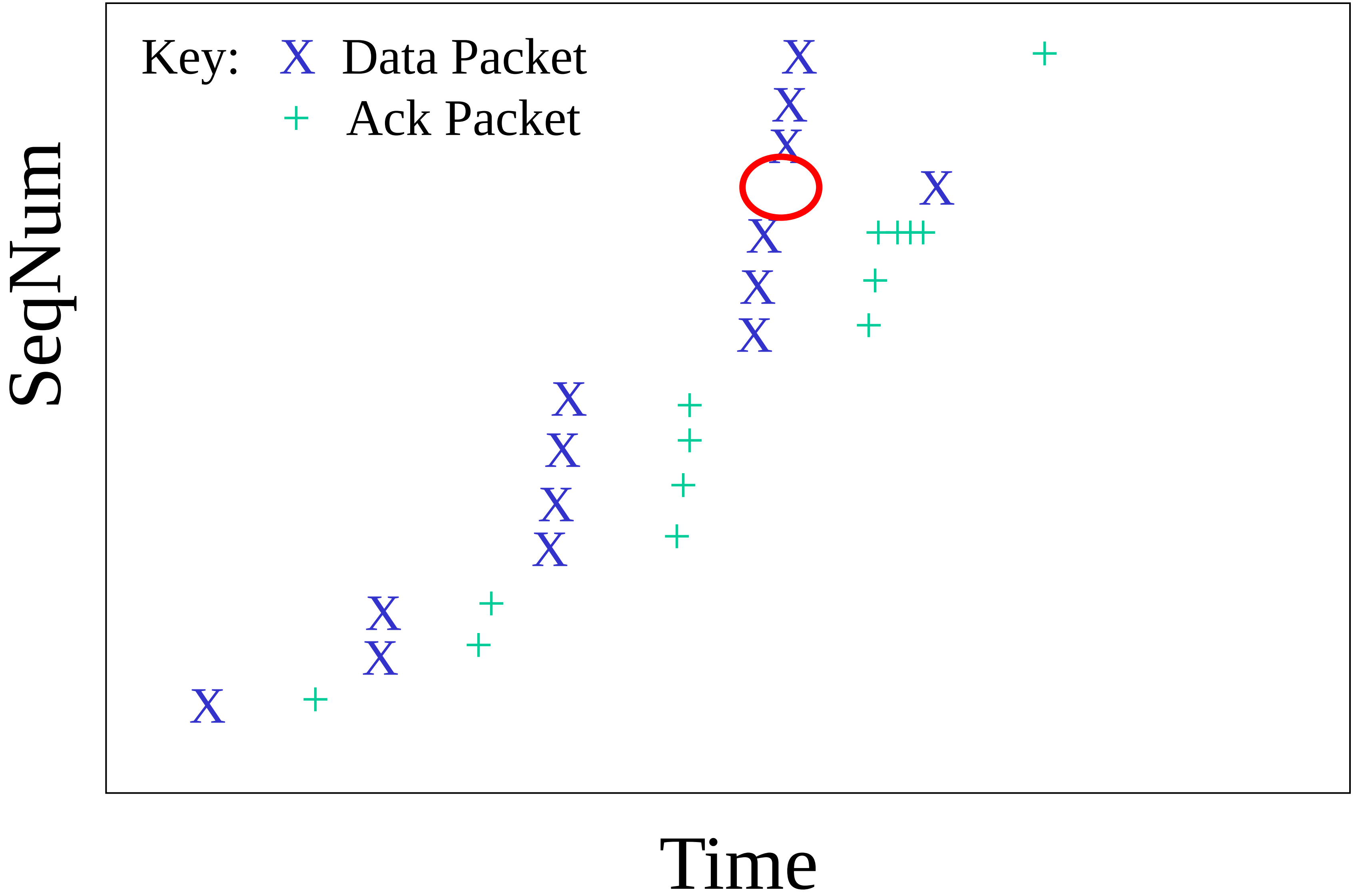

Key:
X Data Packet
+ Ack Packet
SeqNum
Time

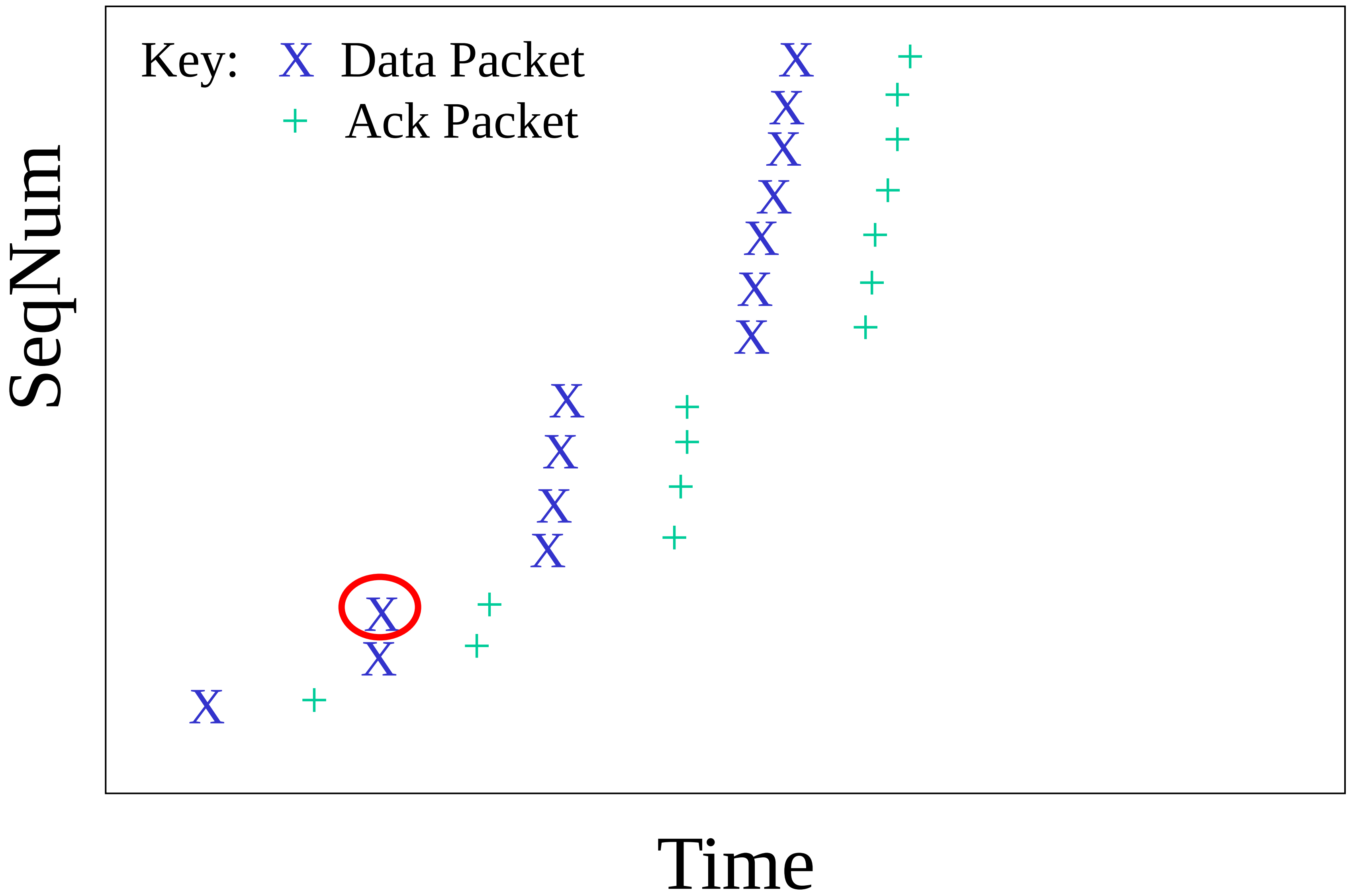

Key:
X Data Packet
+ Ack Packet
?
SeqNum
Time

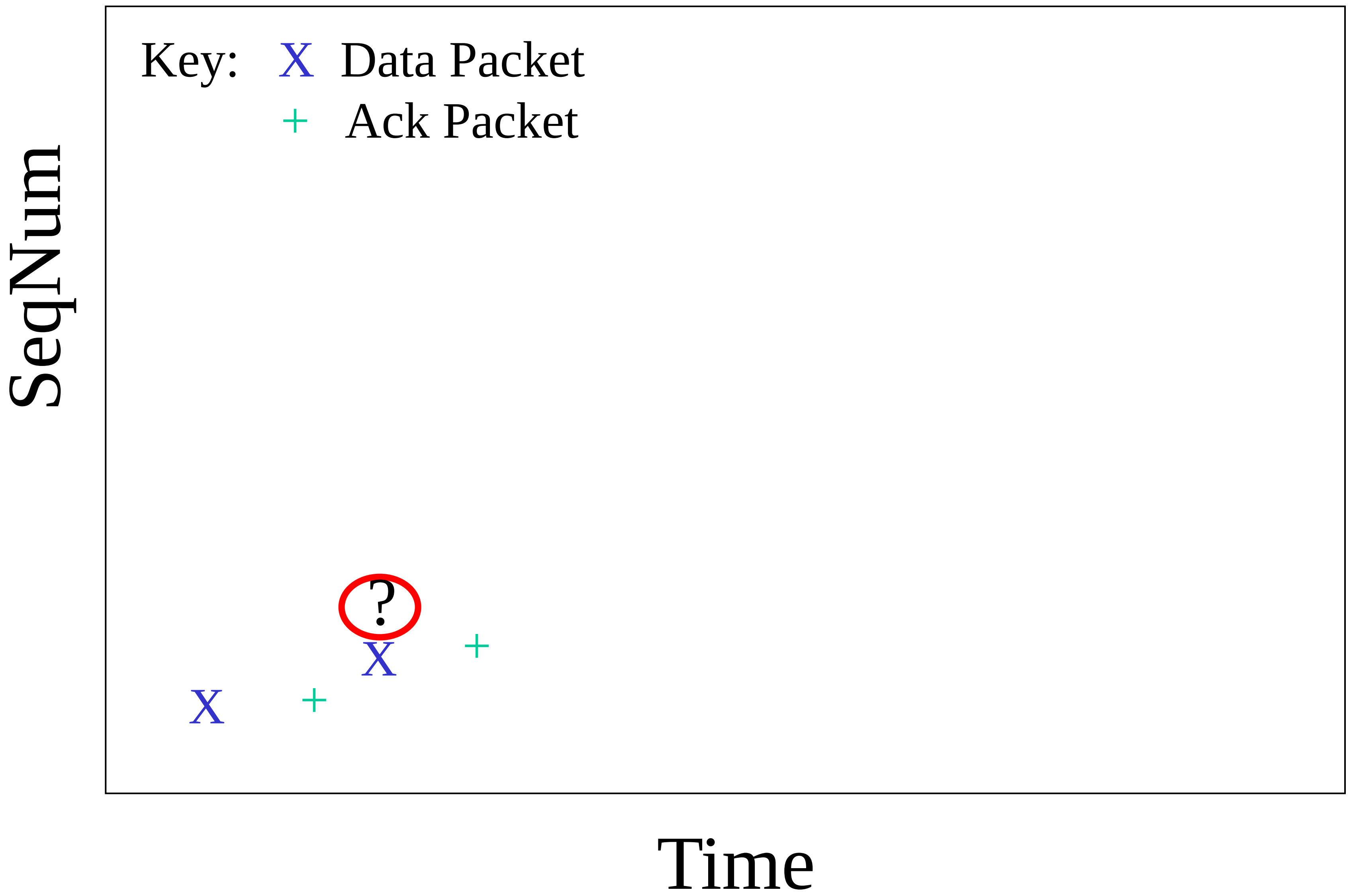

Key:
X Data Packet
+ Ack Packet
SeqNum
Time

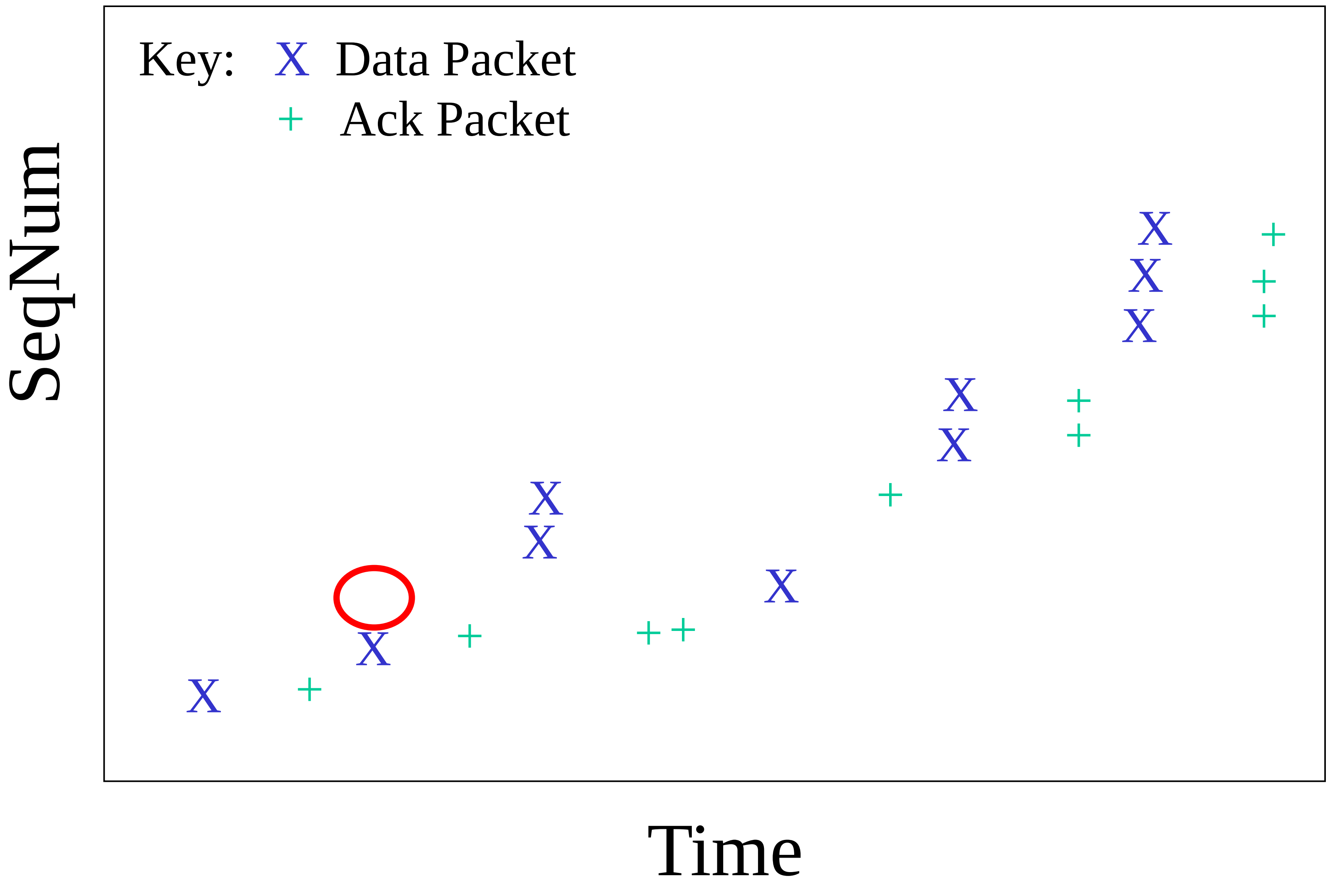

Key:
X Data Packet
+ Ack Packet
SeqNum
Time


Key: X Data Packet
+ Ack Packet
?
SeqNum
Time


Key:
X Data Packet
+ Ack Packet
SeqNum
Time


Key:
X Data Packet
+ Ack Packet
SeqNum
Time


Key: X Data Packet
+ Ack Packet
?
SeqNum
Time


Key:
X Data Packet
+ Ack Packet
SeqNum
Time

TCP 101 (Cont'd)

- **Main observation:**

  “Not all packet losses are created equal”
- **Losses early in the transfer have a huge adverse impact on the transfer latency**
- **Losses near the end of the transfer always cost at least a retransmit timeout**
- **Losses in the middle may or may not hurt, depending on congestion window size at the time of the loss**