

---

# TCP Analysis Based on Flags

---

Cheng Guang

College of Computer Science & Engineering

Southeast University, Nanjing, P.R.China

---

# Introduction

- We have known that, some TCP segments carry data while others are simple acknowledgements for previously received data. Such as, the popular 3-way handshake utilises the SYNs and ACKs available in the TCP to help complete the connection before data is transferred.
  - So we can get a conclusion that each TCP segment has a purpose, and this is determined with the help of the TCP flag options, allowing the sender or receiver to specify which flags should be used so the segment is handled correctly by the other end.
-

---

# Flags

- Six Flags
    - Urgent Pointer
    - ACKnowledgement
    - PUSH
    - Reset (RST) Flag
    - SYNchronisation Flag
    - FIN Flag
  - All flags, a value of '1' means that a particular flag is 'set'.
  - Each flag is one bit long, and since there are 6 flags, this makes the Flags section 6 bits in total.
-

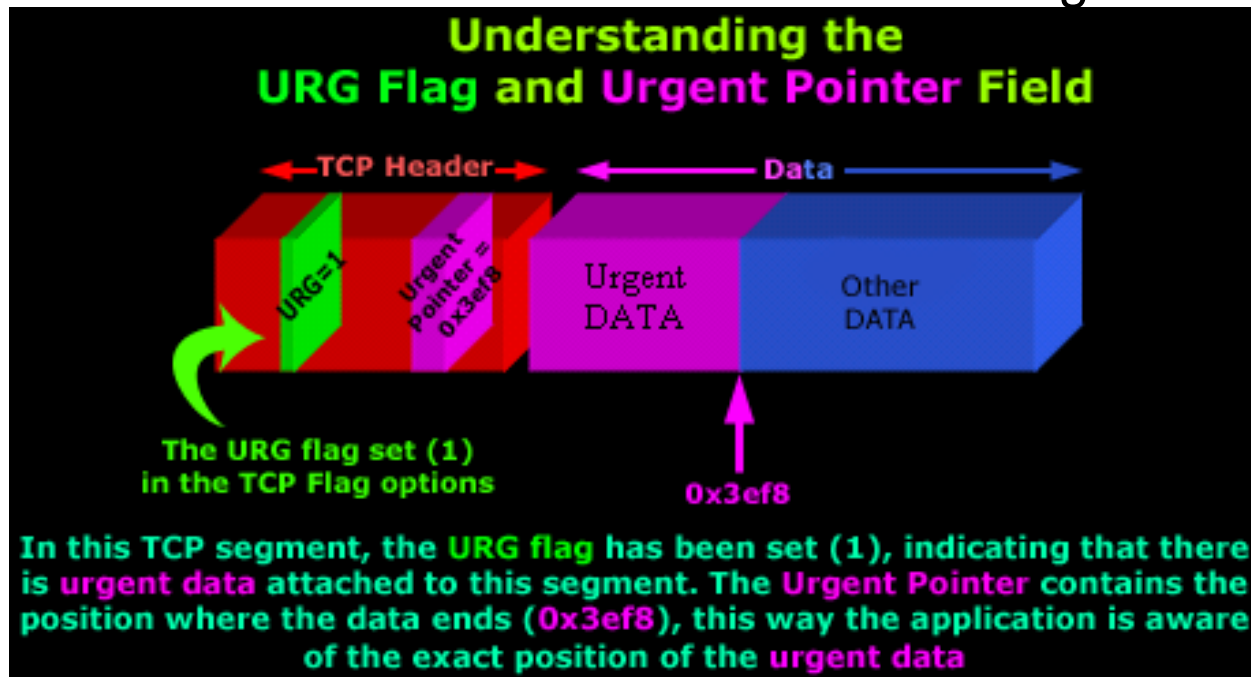
---

# 1st Flag - Urgent Pointer

- This flag is used to identify incoming data as 'urgent'
  - incoming segments are sent directly and processed immediately.
  - An Urgent Pointer could be used during a stream of data transfer where a host is sending data to an application running on a remote machine.
  - By setting the abort signal's segment Urgent Pointer flag to '1', the remote machine will give that specific segment priority, processing it immediately and stopping all further data processing.
-

# The Urgent Pointer

- The urgent pointer flag allows us to mark a segment of data as 'urgent', while this urgent pointer field specifies where exactly the urgent data ends.
- Urgent Pointer can also be used when attacking remote hosts.



---

## 2nd Flag - ACKnowledgement

- The ACKnowledgement flag is used to acknowledge the successful receipt of packets.

---

## 3rd Flag - PUSH

- The Push flag ensures that the data is given the priority and is processed at the sending or receiving end.
  - This particular flag is used quite frequently at the beginning and end of a data transfer, affecting the way the data is handled at both ends.
  - Push flag is usually set on the last segment of a file to prevent buffer deadlocks.
-

---

## 4th Flag - Reset (RST) Flag

- The reset flag is used when a segment arrives that is not intended for the current connection.
  - This 'feature' is used by most hackers in order to scan hosts for 'open' ports.
  - All modern port scanners are able to detect 'open' or 'listening' ports thanks to the 'reset' function.
-

---

# 5th Flag - SYNchronisation Flag

- The SYN flag is initially sent when establishing the classical 3-way handshake between two hosts
  - During the 3-way handshake we are able to count a total of 2 SYN flags transmitted, one by each host.
  - As files are exchanged and new connections created, we will see more SYN flags being sent and received.
-

---

## 6th Flag - FIN Flag

- This flag is used to tear down the virtual connections created using the previous flag (SYN).
  - FIN flag always appears when the last packets are exchanged between a connection.
-

---

# Analysis worm's SYN scan

- If a worm was designed to spread via TCP, during its propagation there will be a lot of corresponding TCP SYN packets sent out as it seeks vulnerable services in other hosts.
    - the destination host is alive,
    - the destination host it attempts to connect to is not living
    - the destination is alive but connection attempts from the worm are not functional
  - when worm tries to propagate, the destination addresses are typically generated at random, and normally there will be a large number of destination hosts that are not living or functional. Therefore we should expect to see a large number of SYN bits set in the flow records associated with the worm-infected host.
-

---

# Analysis RST/ACK

- a closed port will send back a RST/ACK to a TCP request,
  - If a worm is scanning a large block of living hosts, those hosts with closed ports would send back a RST/ACK.
  - if a destination host receives too many RST/ACK responses, this destination IP is very likely infected with a worm.
-

---

# SYN flood attack

- attacker creates a random source address
  - SYN flag set in each packet is a request to open a new connection
  - victim responds to spoofed IP address, then waits for confirmation that never arrives
  - victim's connection table fills up waiting for replies
  - after table fills up, all new connections are ignored
-

---

# SYN

- SYN+ACK Packet SA\_SYN
  - RST+ACK packet RA\_SYN
  - Non Response Non\_SYN
  - $SYN\# = SA\_SYN\# + RA\_SYN\# + Non\_SYN\#$
-



# Related Model

- $\text{SYN\#} = \text{SA\_SYN\#} + \text{RA\_SYN\#} + \text{Non\_SYN\#}$
- $\text{SYN\_ACK\#} = \text{FA\_SA\#} / 2 + \text{RA\_SA\#} + \text{RT\#} + \text{Non\_SA\#}$
- $\text{SA\_SYN\#} = \text{SYN\_ACK\#}$
- $\text{RESET\_ACK\#} = \text{RA\_SYN\#} + \text{RA\_SA\#}$
- $\text{RESET\#} = \text{RT\#}$
- $\text{FIN+ACK\#} = \text{FA\_SA\#}$
- Have 7 variables, but only 6 equations
- ? Non\_SYN#, RA\_SYN#, RA\_SA#, Non\_sa#

# Hypothesis

- We only consider the SYN attacks.
- Let  $\text{Non\_sa\#} = 0$ ;
  - $\text{SYN\#} = \text{SA\_SYN\#} + \text{RA\_SYN\#} + \text{Non\_SYN\#}$
  - $\text{SYN\_ACK\#} = \text{FA\_SA\#} / 2 + \text{RA\_SA\#} + \text{RT\#}$
  - $\text{SA\_SYN\#} = \text{SYN\_ACK\#}$
  - $\text{RESET\_ACK\#} = \text{RA\_SYN\#} + \text{RA\_SA\#}$
  - $\text{RESET\#} = \text{RT\#}$
  - $\text{FIN+ACK\#} = \text{FA\_SA\#}$
  - Have 6 variables, but only 6 equations
  - ?  $\text{Non\_SYN\#}$ ,  $\text{RA\_SYN\#}$ ,  $\text{RA\_SA\#}$

# EXAMPLES

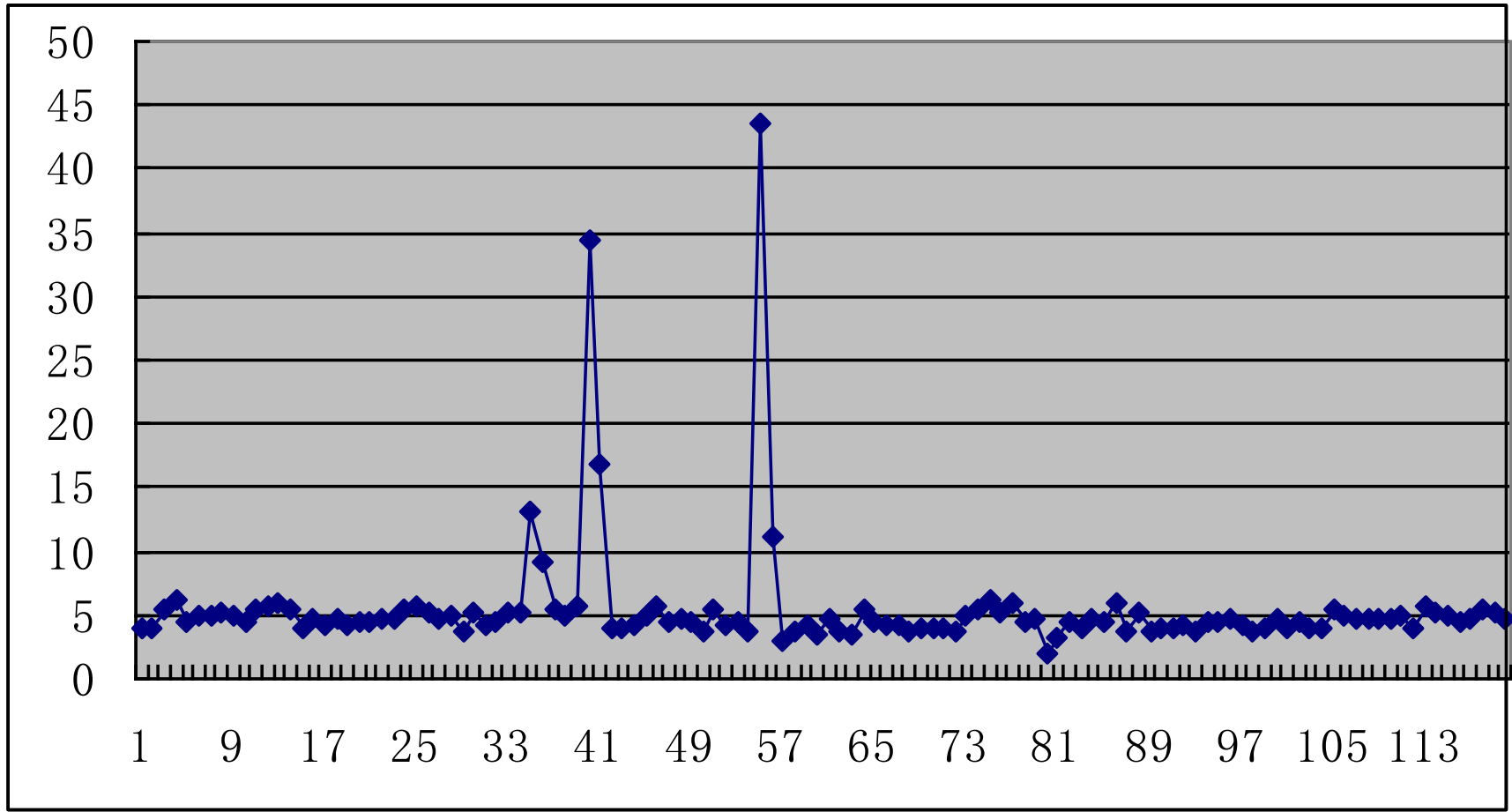
- | ■ | SYN       | SYN+ACK | FIN  | RST   |
|---|-----------|---------|------|-------|
| □ | (1) 27390 | 6739    | 1837 | 17983 |
| □ | (2) 30387 | 7736    | 1998 | 14607 |
| □ | (3) 29644 | 5411    | 2336 | 17129 |
- 
- So we can compute the SYN # which don't have any response.
    - (1) Non\_SYN# = 8488
    - (2) Non\_SYN# = 14780
    - (3) Non\_SYN# = 11347

---

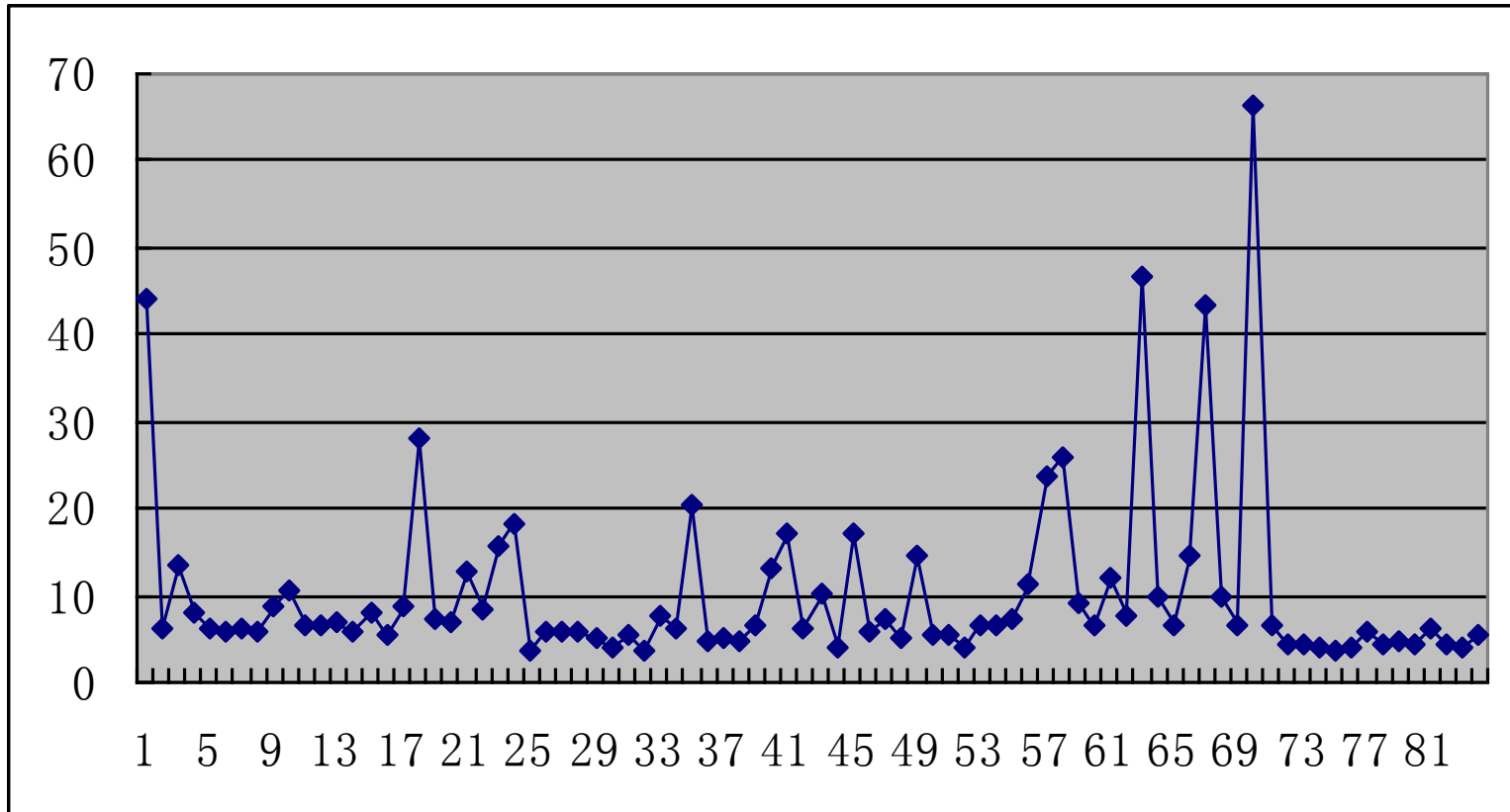
# Problems

- In this example, we only consider the SYN flood and scan security, and our aim is to compute the number of SYN which is abnormal SYN packets.
  - In the future, we will consider more conditions, so that, we can estimate more variables.
-

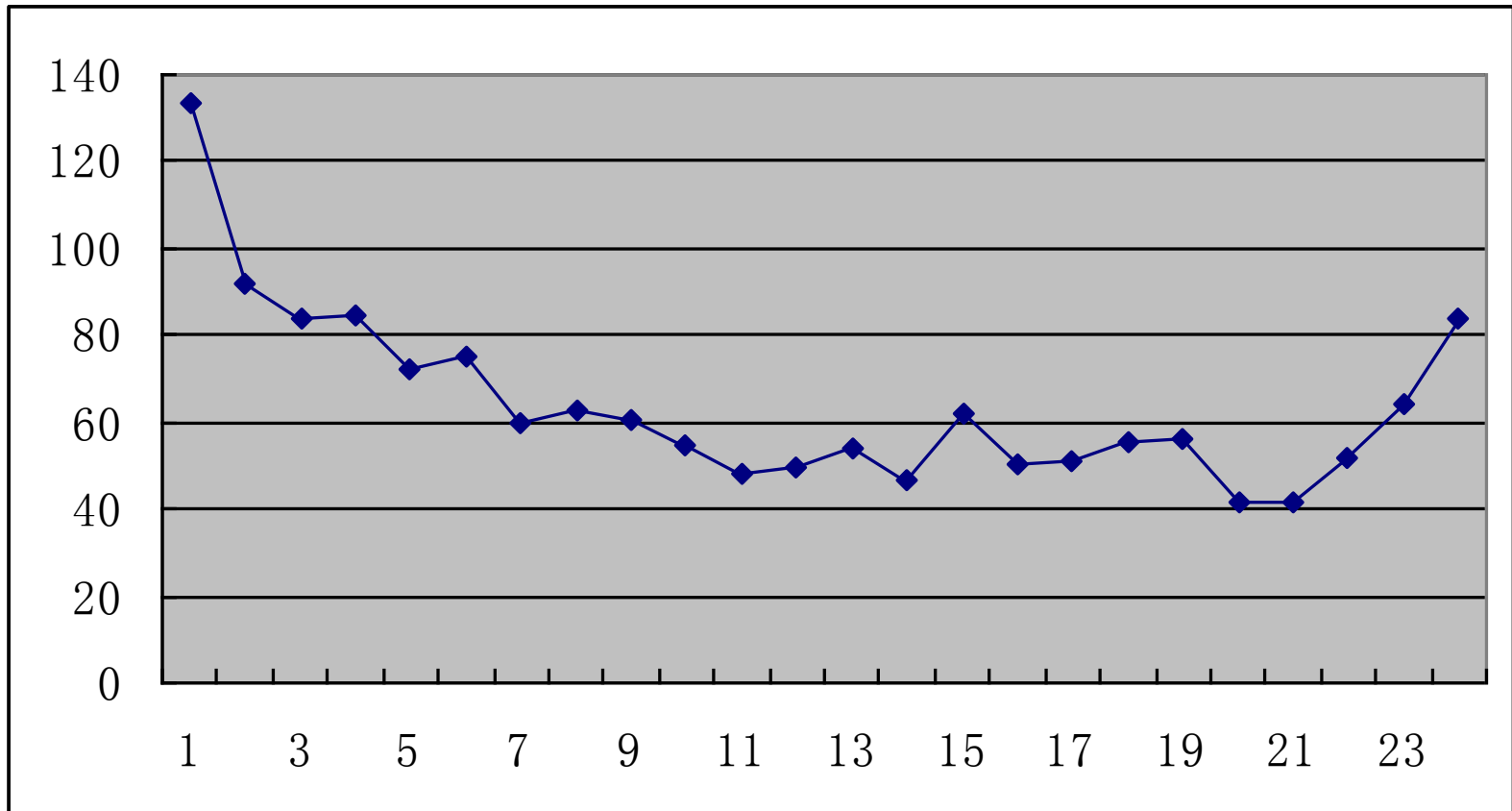
Timestamps = 10s SYN/SYN+ACK



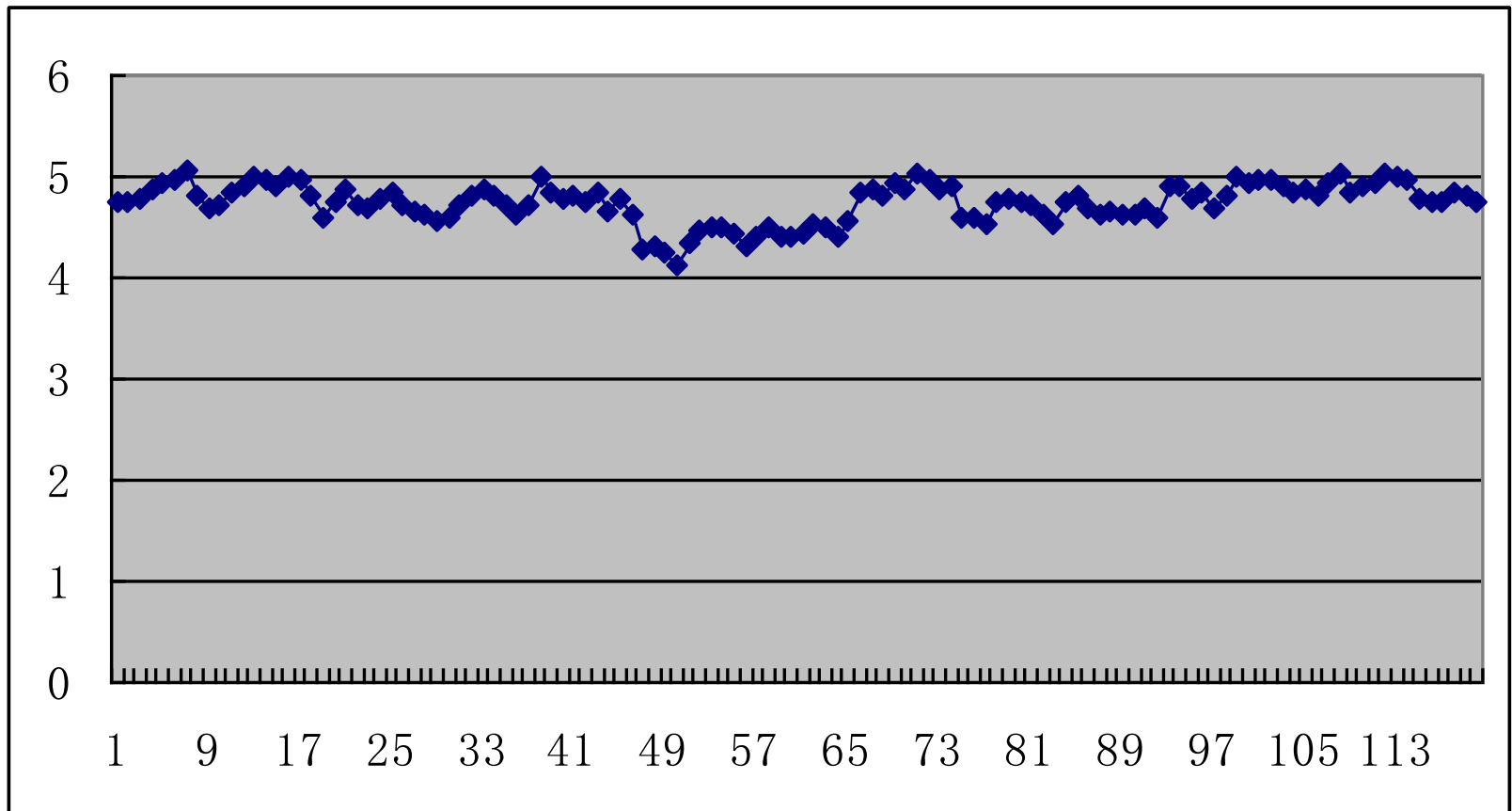
# Timestamps = 600s SYN/SYN+ACK 2005



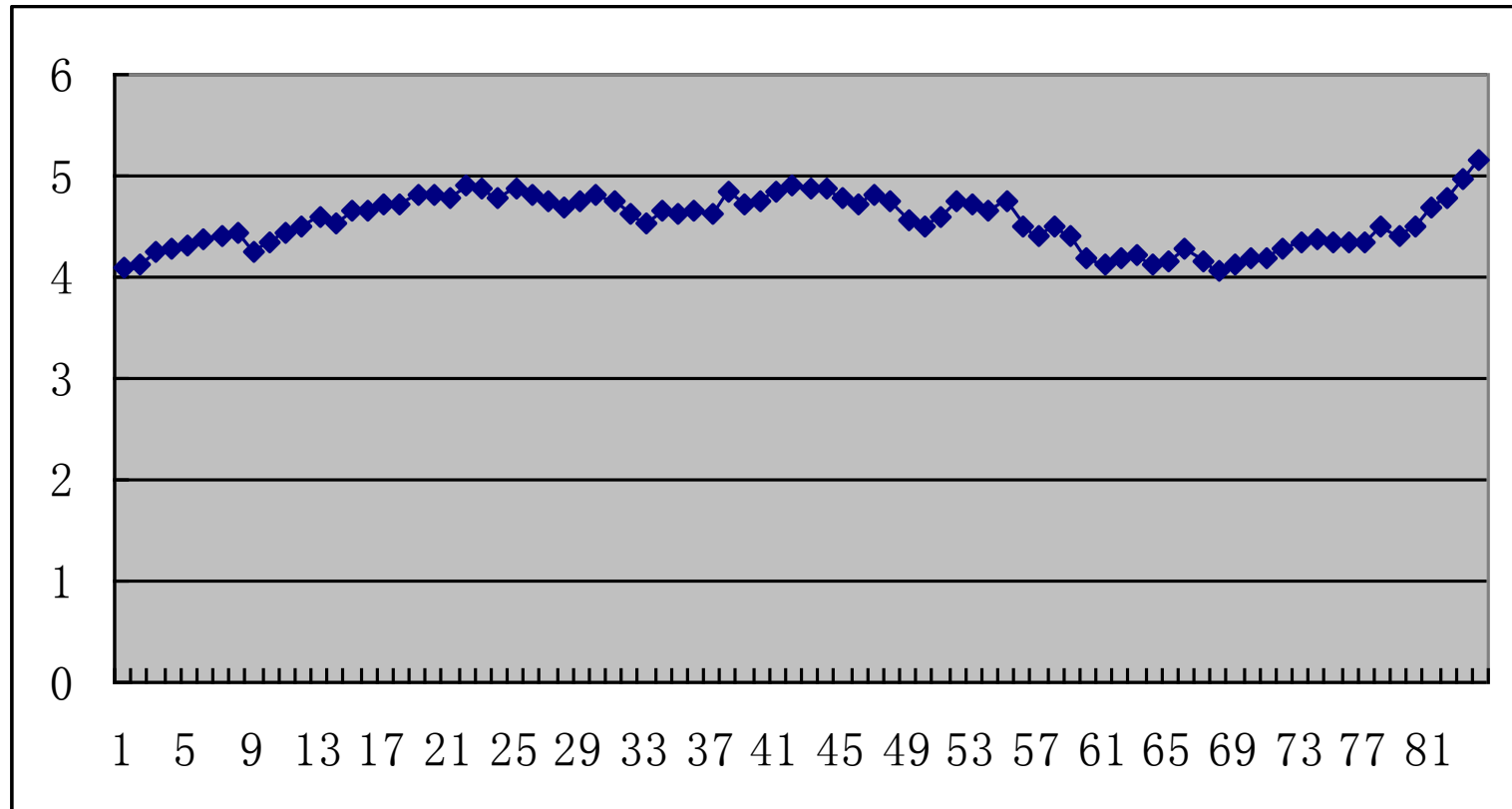
Timestamps = 600s SYN/SYN+ACK  
2004.4.17



Timestamps = 10s ACK/PUSH

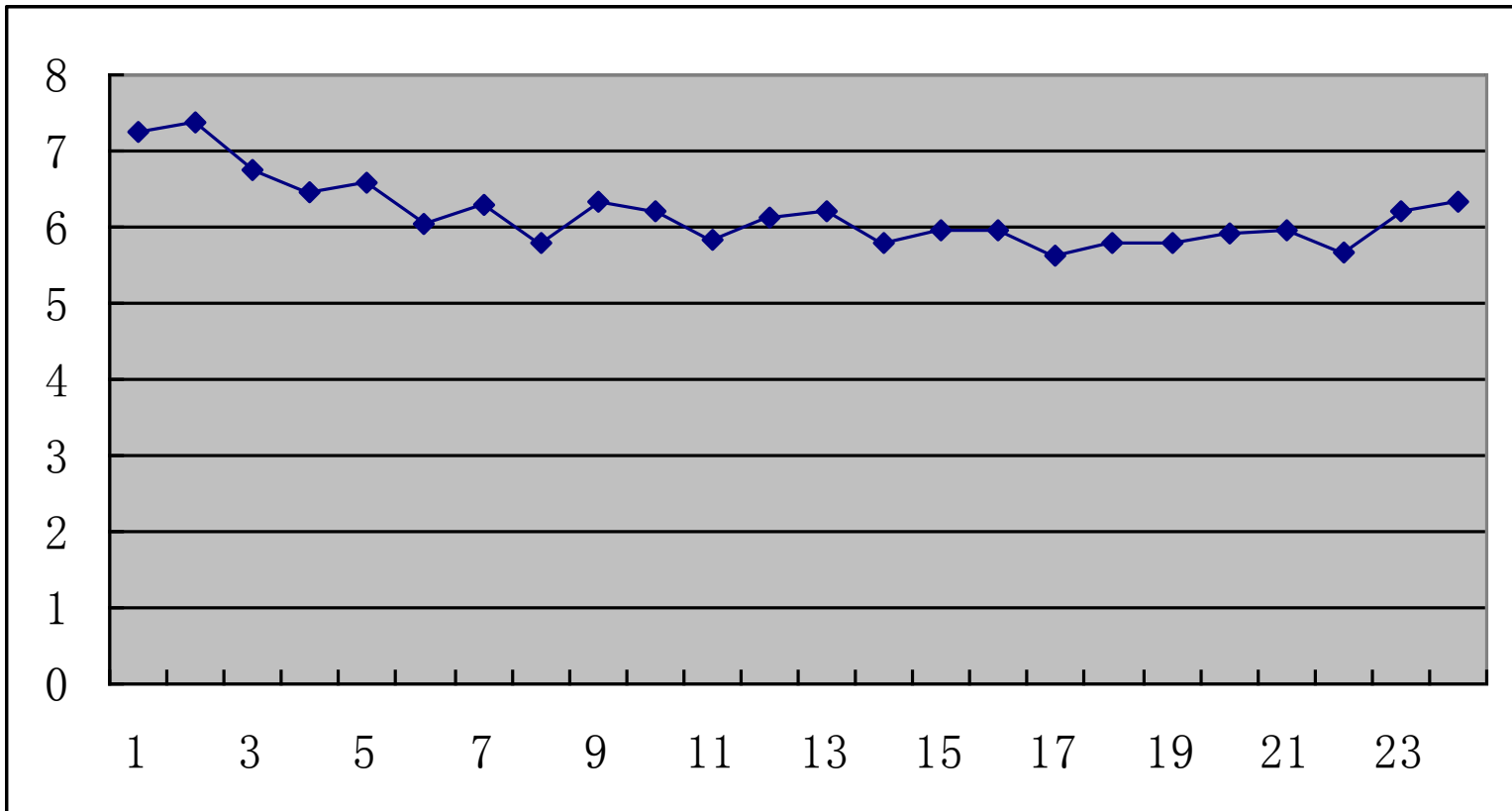


# Timestamps = 600s ACK/PUSH 2005



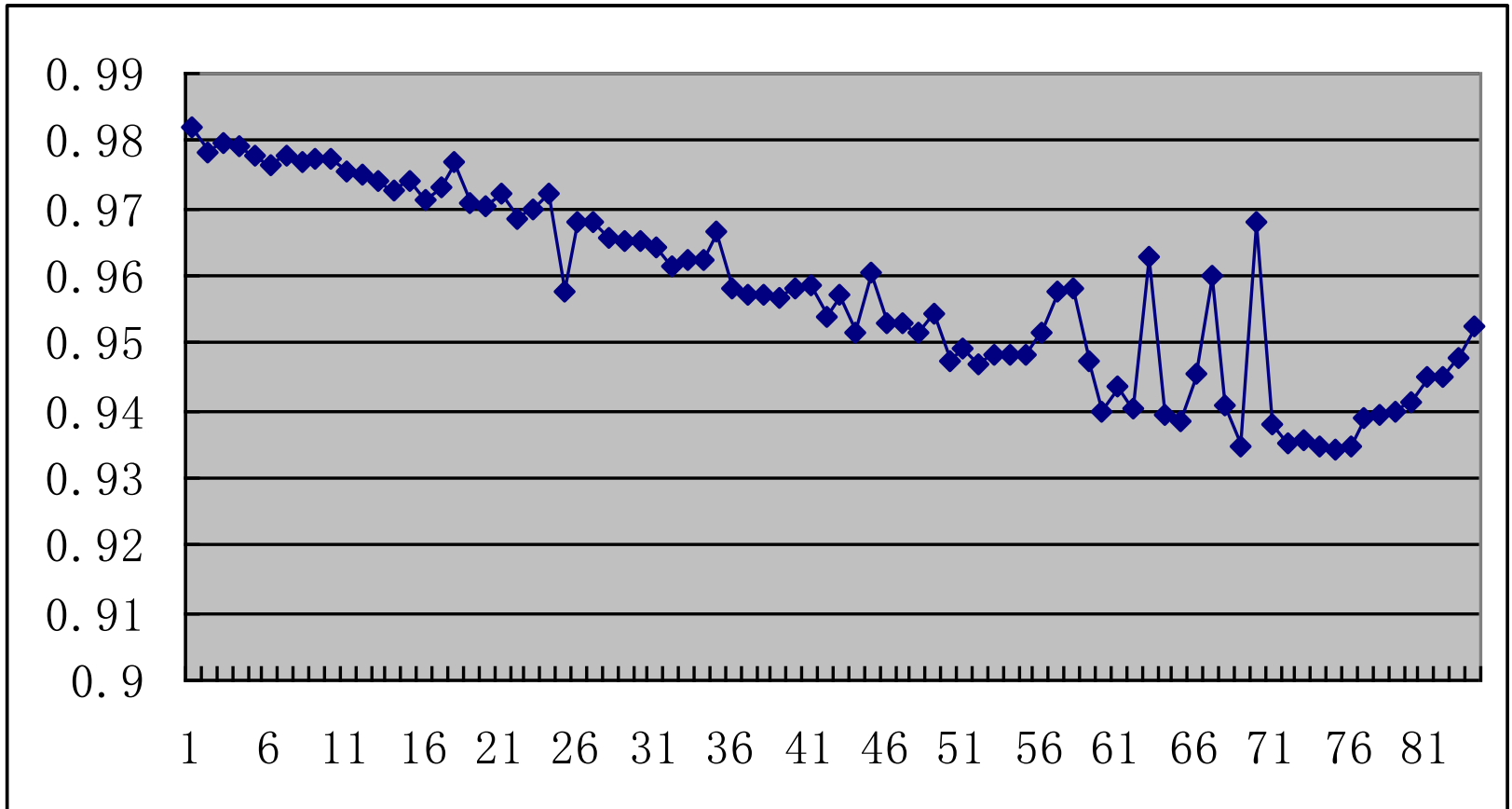
Timestamps = 600s ACK/PUSH

2004.4.17



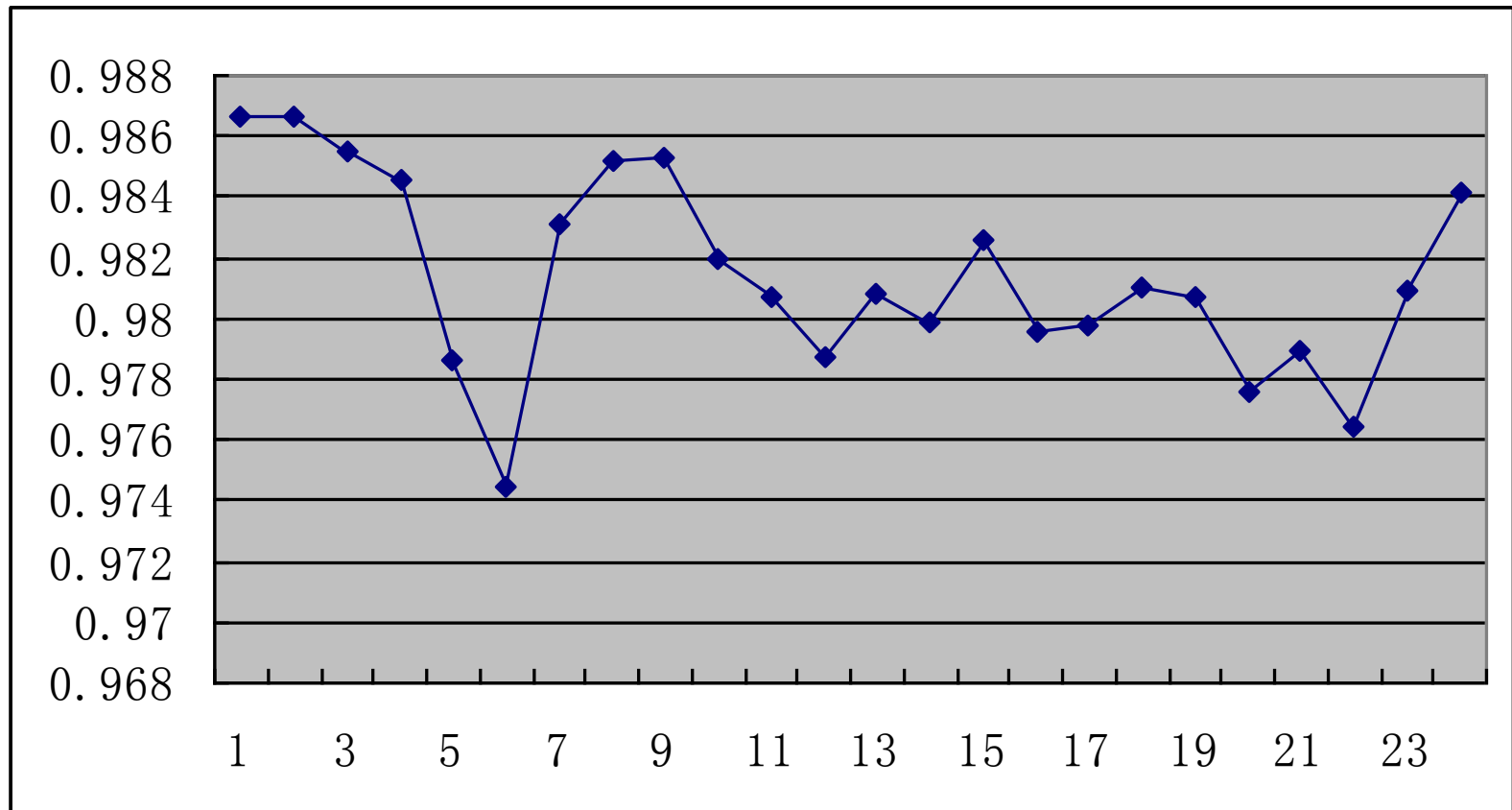
# ACK+SYN+PUSH/SUM

2005

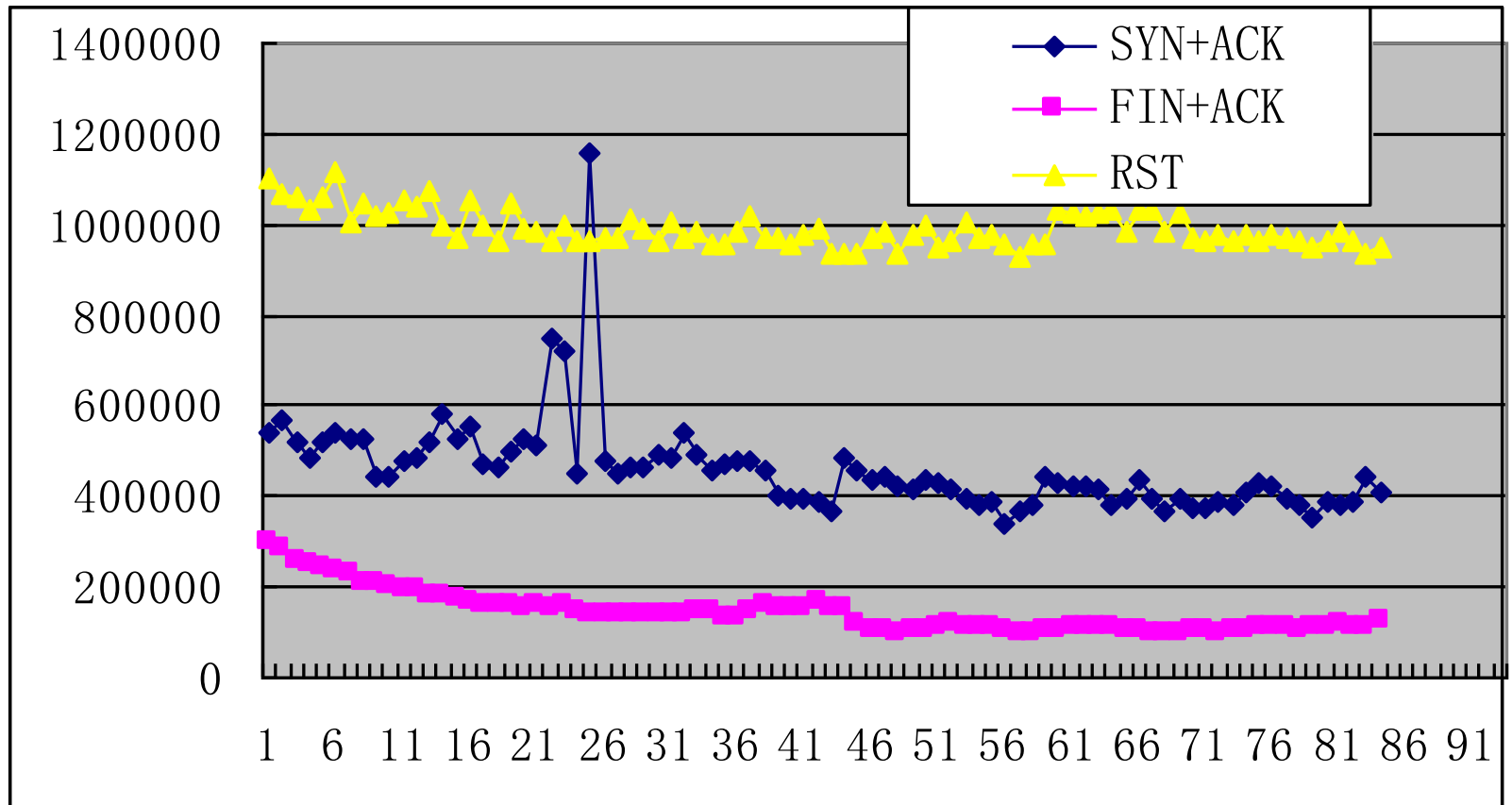


# ACK+SYN+PUSH/SUM

2004.4.17



# Time Series



---

Thank You!

---