Detecting SYN Flooding Attacks

Haining Wang, Dandle Zhang, Kang G. Shin

Serhat TÜRKMEN

N10163791

Outline

- Introduction
- Related Issues
- Attack Detection
- Performance Evaluation
- Related Work
- Conclusion

Introduction

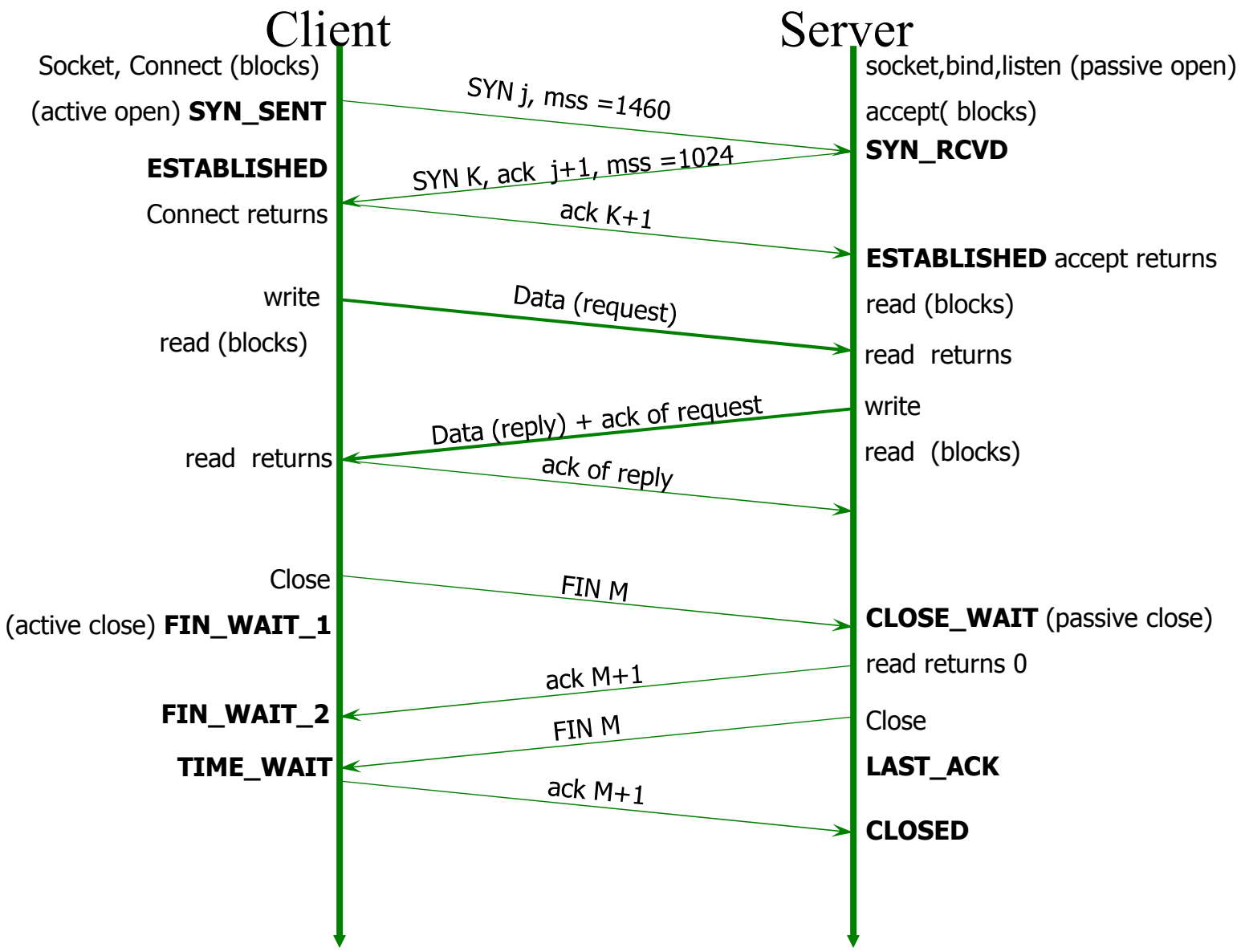
- Attacks on popular sites
- Most of them are DoS using TCP
  - TCP SYN flooding is the most common
  - Web Server, Mail Server, FTP Server
- SYN Flooding exploits TCP 3-way hand-shake
- Internet routing infrastructure can not differentiate legitimate and spoofed SYN

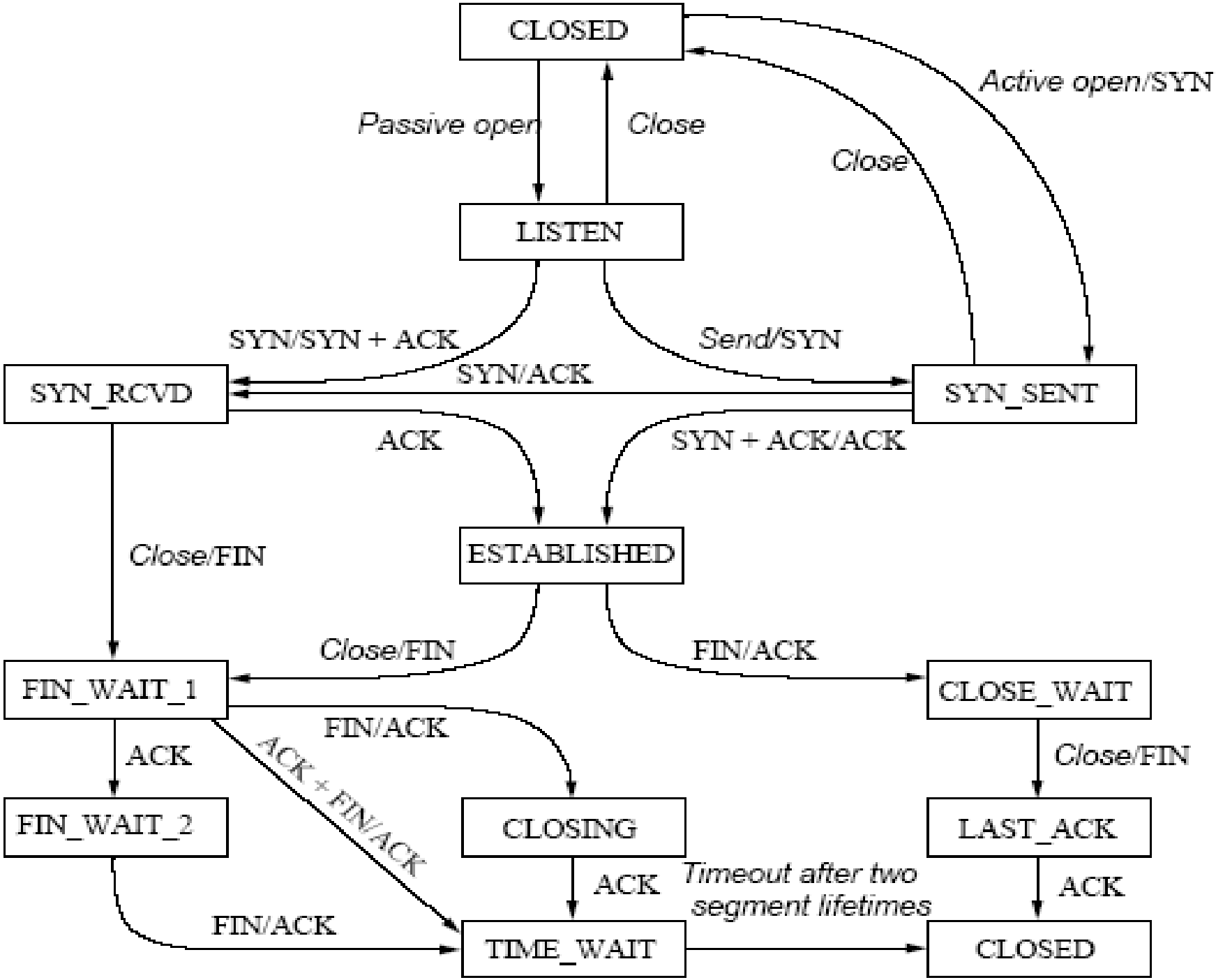
Introduction (cont)

- Syn Cache, Syn cookies, SynDefender, Syn Proxying and SynKill
- Installed on firewall or victim server
- Need expensive traceback to detect attacker
- These mechanisms are vulnarable to SYN flood.
- Specialized firewalls become worthless with 14000 packets per sec.

Introduction (cont)

- FDS – Flooding Detection System
- Stateless, low computation overhead
- Installed on leaf routers (First-mile or Last-mile routers)
- FDS uses key feature of TCP SYN-FIN pairs behavior.


Client
Server
Socket, Connect (blocks)
(active open) SYN_SENT
ESTABLISHED
Connect returns
write
read (blocks)
read returns
Close
(active close) FIN_WAIT_1
FIN_WAIT_2
TIME_WAIT
socket,bind,listen (passive open)
accept(blocks)
SYN_RCVD
ESTABLISHED accept returns
read (blocks)
read returns
write
read (blocks)
CLOSE_WAIT (passive close)
read returns 0
Close
LAST_ACK
CLOSED
SYN j, mss =1460
SYN K, ack j+1, mss =1024
ack K+1
Data (request)
Data (reply) + ack of request
ack of reply
FIN M
ack M+1
FIN M
ack M+1


CLOSED
Active open/SYN
Passive open
Close
Close
LISTEN
SYN/SYN + ACK
Send/SYN
SYN/ACK
SYN_RCVD
SYN_SENT
ACK
SYN + ACK/ACK
Close/FIN
ESTABLISHED
Close/FIN
FIN/ACK
FIN_WAIT_1
CLOSE_WAIT
FIN/ACK
ACK
Close/FIN
ACK + FIN/ACK
FIN_WAIT_2
CLOSING
LAST_ACK
ACK
Timeout after two
segment lifetimes
ACK
FIN/ACK
TIME_WAIT
CLOSED

Related Issues

- Packet Classification
- Placement of Detection Mechanism
- Discrepancy between SYN’s and FIN’s

Packet classification

- TCP packet classification (SYN, FIN, RST) is done at leaf router
- SYN (beginning) FIN (End) for each TCP connection
- No means to distinguish active FIN and passive FIN
- RST violates the SYN-FIN pairs
- First two steps confirm that it is a TCP packet
- Code Bits in IP packet equals the sum of the length of IP header and offset of code BIT’s in TCP

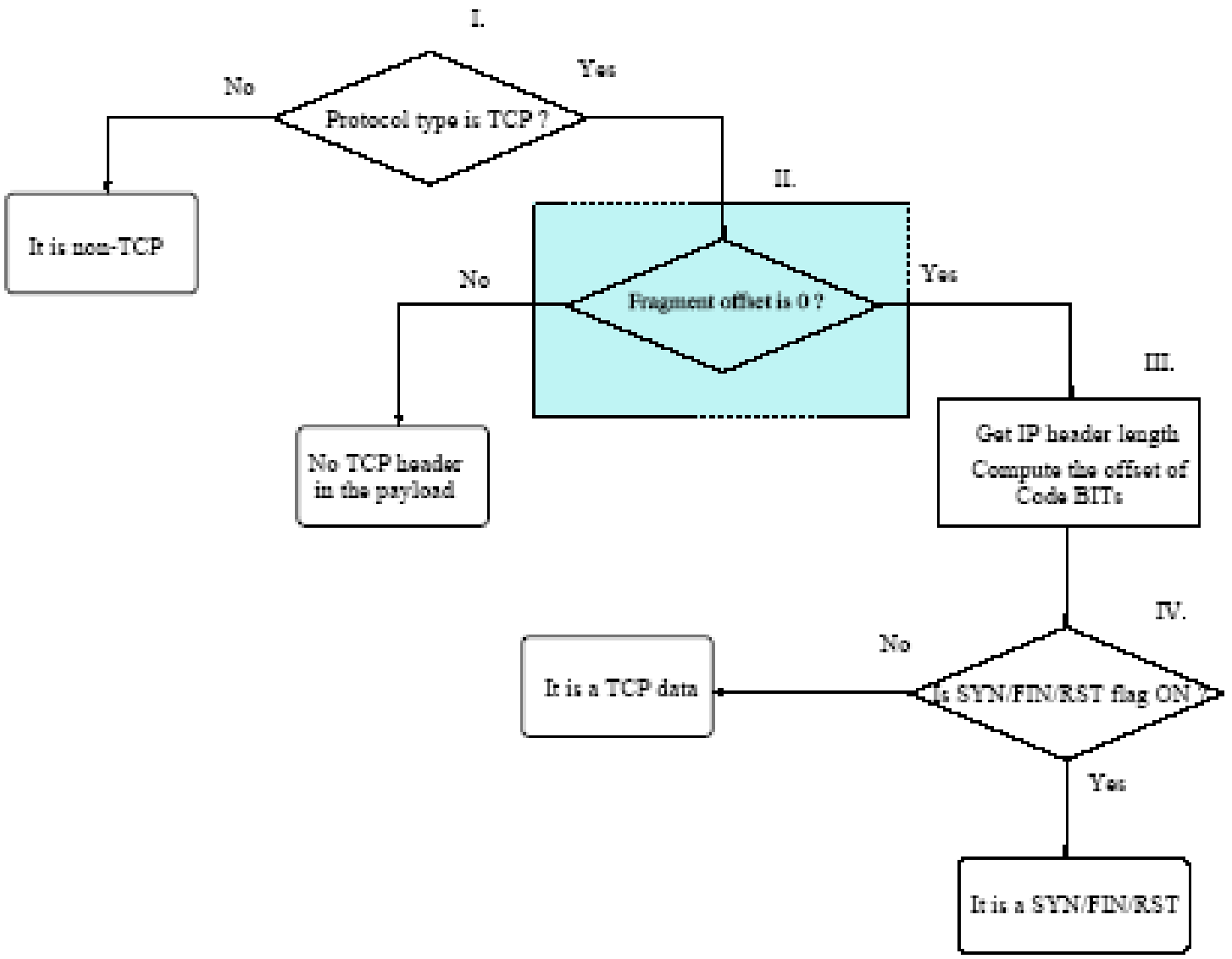


Fig. 2. The flowchart of the packet classification at leaf routers

Placement of Detection Mechanism

- FDS is installed at the first-mile or last mile router
- First-mile is more likely to catch flooding detection source due to proximity to attack sources.
- Last-mile quickly detects the flooding but cant provide hint about flooding sources.
- FDS is not installed at core due to
  - it is close to neither flooding sources not the victim
  - packets of the same flow could traverse different paths

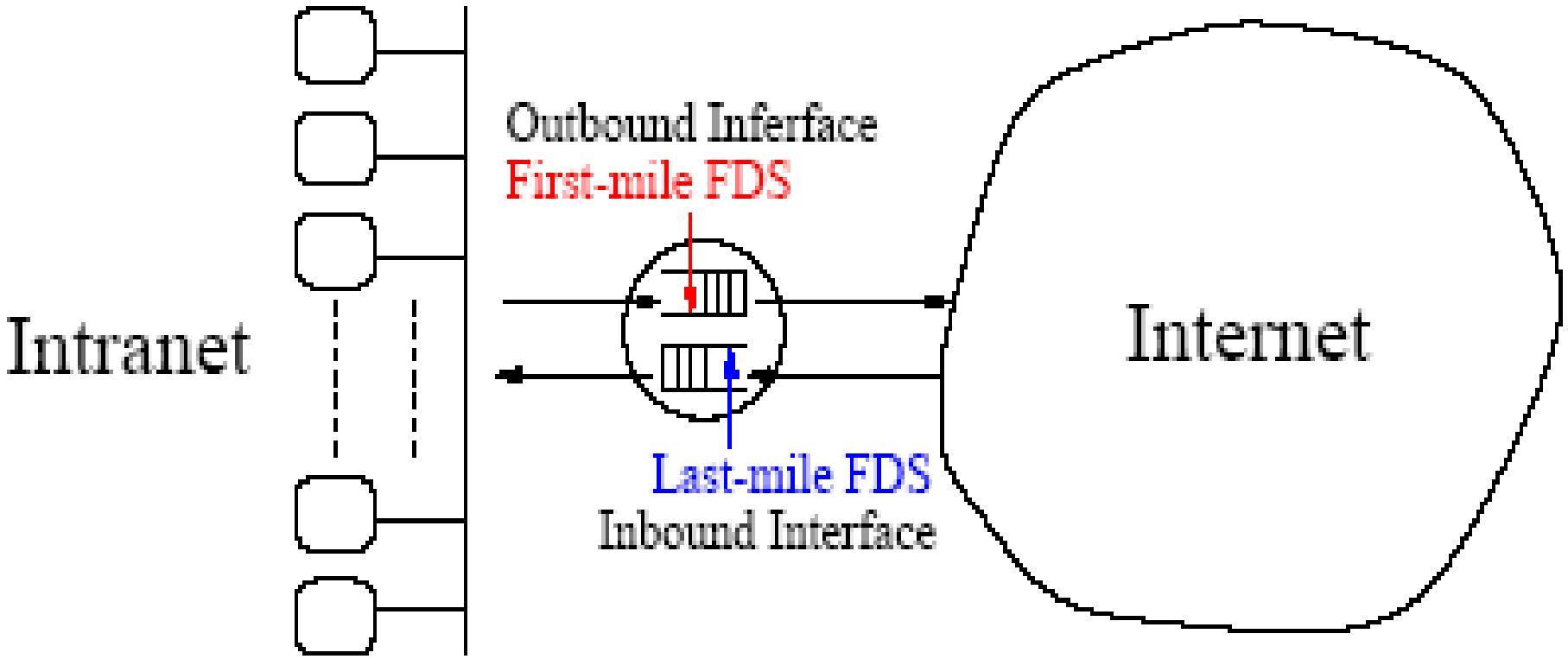


Fig. 3. The installation of FDS at a leaf router

Discrepancy between SYN's and FIN's

- Single RST packet can terminate a TCP session
  - Violate the SYN-FIN pairs.
- Passive RST transmitted in response to arrival of a packet to the closed port
- Active RST transmitted in response to abort a TCP connection and associated with a SYN
- Normal behavior of TCP:
  - (SYN,FIN), (SYN/ACK,FIN) and (SYN,RST_{active})
- FDS cannot differentiate between active and passive RST

Discrepancy between SYN's and FIN's

- All RST is active; reduce sensitivity
- All RST is passive; raise false alarm

Normal Conditions :

1) SYN and RST have a strong correlation
2) Difference between SYNs and FINs is equal to RSTs
   - Threshold is set at 75%, i.e., 3 out of 4 RSTs are active
   - CUSUM withstand the effects of incorrectly classified passive RST's

Attack Detection

- Data Sampling and Detection Mechanism
  - SYN and FIN packets collected over time t_0
  - Sampling time of FIN(RST) t_d later than SYN
  - Recent study : TCP Connections 12-19 sec
  - t_d set to 10 sec and t_0 is set to 20 sec
  - The correlation between the number of SYNs and FINs (RSTs) is not sensitive to the request. (time, sites….)
- Minimum flooding rate
  - Unprotected Server: 500 SYNs per second
  - With a specialized Server: 1400 SYNs per second
  - 300.000 SYNs to shut down a server for 10 minutes

Attack Detection

- Change Point Detection
  - Posterior Change Point Detection
  - Sequential Change Point Detection
    - Quicker response
    - Saves computation and memory
    - Modelling of X_n is only problem
    - Non-model spesific tests are needed
    - Non-parametric CUSUM fit this requirement
- False Alarm Time: the time duration with no false alarm reported when there is no attack
- Detection Time: The detection delay after the attack starts.

Attack Detection

- The CUSUM algorithm
  - $\{\Delta_n, n=0,1,..\}$ Number of SYNs-FINs.
  - $\{\Delta_n\}$ is Normalized by average number of F of FINs(RSTs)

$$\bar{F}(n) = \alpha \bar{F}(n-1) + (1-\alpha)\mathrm{FIN}\ (\mathrm{RST})(n),$$

  - $X_n = \Delta_n / F$
  - X_n is no longer dependent on the network size or time of the day.
  - $\{y_n\}$ large value indicates of an attack.

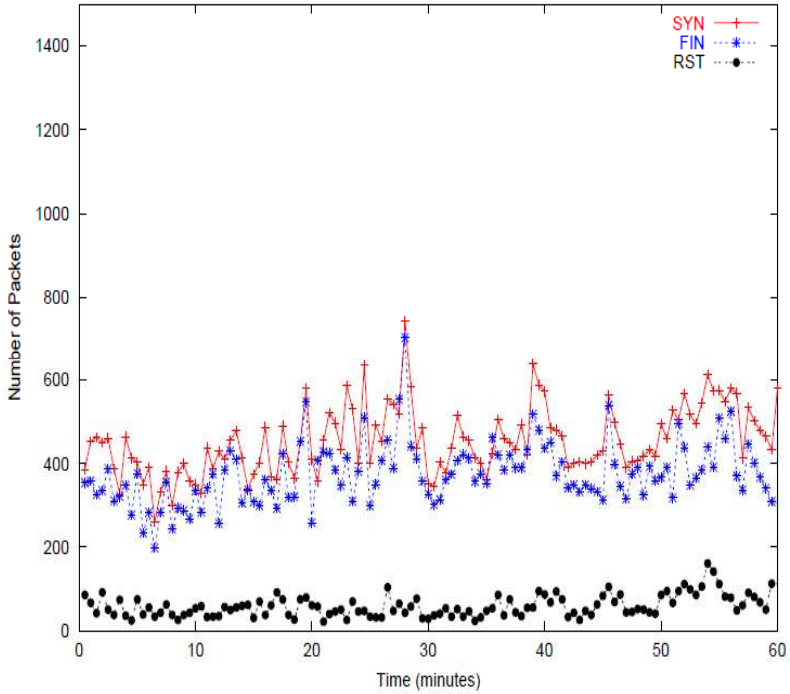
Performance Evaluation

TABLE I

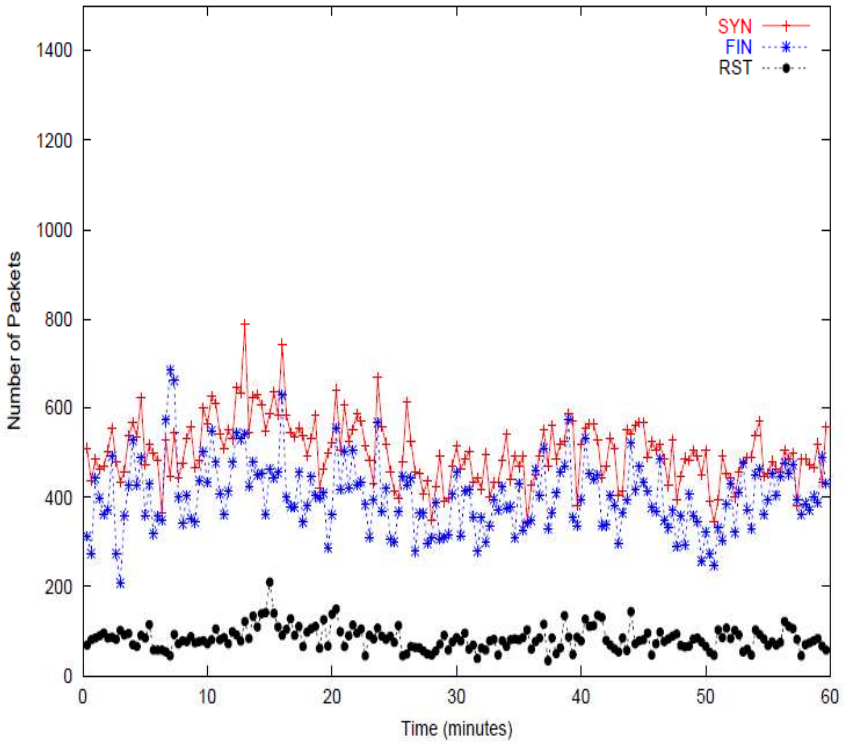
A SUMMARY OF THE TRACE FEATURES

Trace	Starting time	Traffic type
DEC-1	2:00, Thu Mar 9, 95	Bi-directional
DEC-2	10:00, Thu Mar 9, 95	Bi-directional
Harvard-1	12:39, Thu Mar 13, 97	Bi-directional
Harvard-2	16:39, Thu Mar 13, 97	Bi-directional
UNC-in	19:30, Wed Sept 27, 00	Uni-directional
UNC-out	19:30, Wed Sept 27, 00	Uni-directional

The Dynamics of SYN and FIN (RST) packets

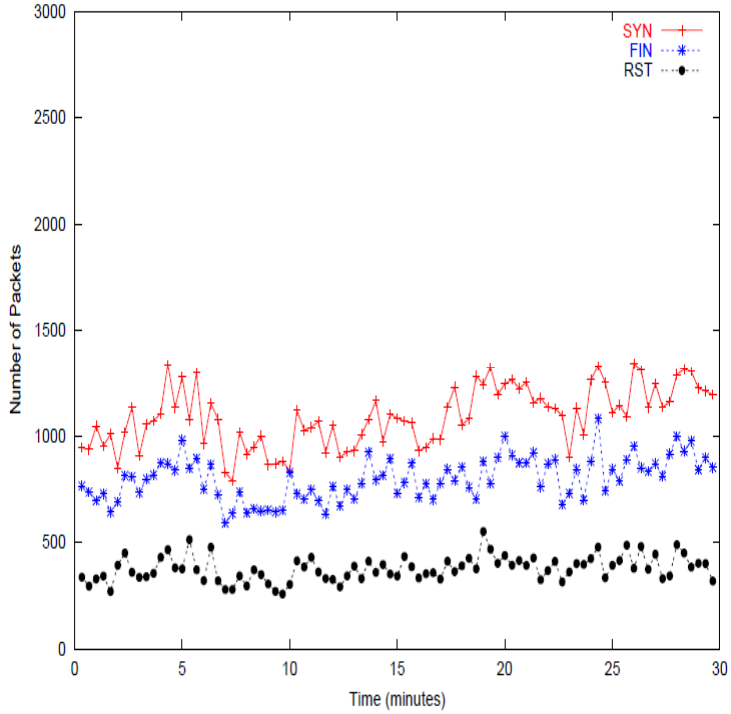


(a) DEC-1

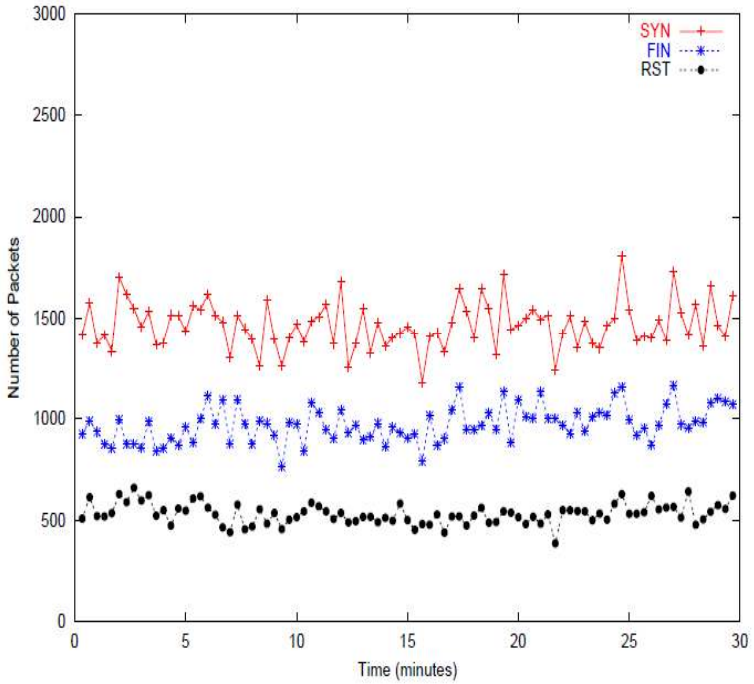


(b) DEC-2

The Dynamics of SYN and FIN (RST) packets

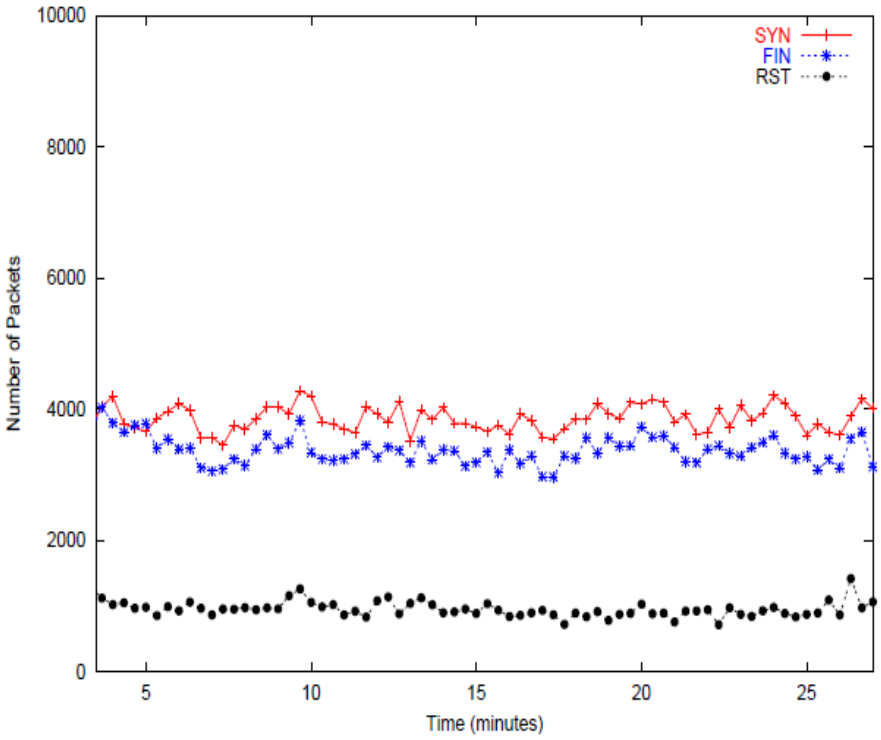


(c) Harvard-1

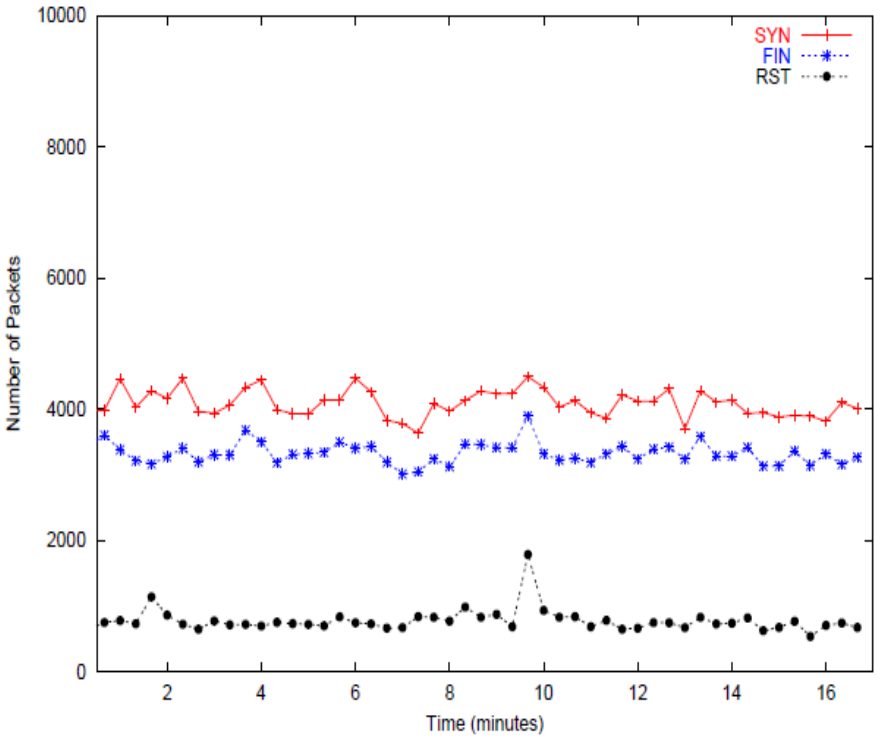


(a) Harvard-2

The Dynamics of SYN and FIN (RST) packets



(b) UNC-in

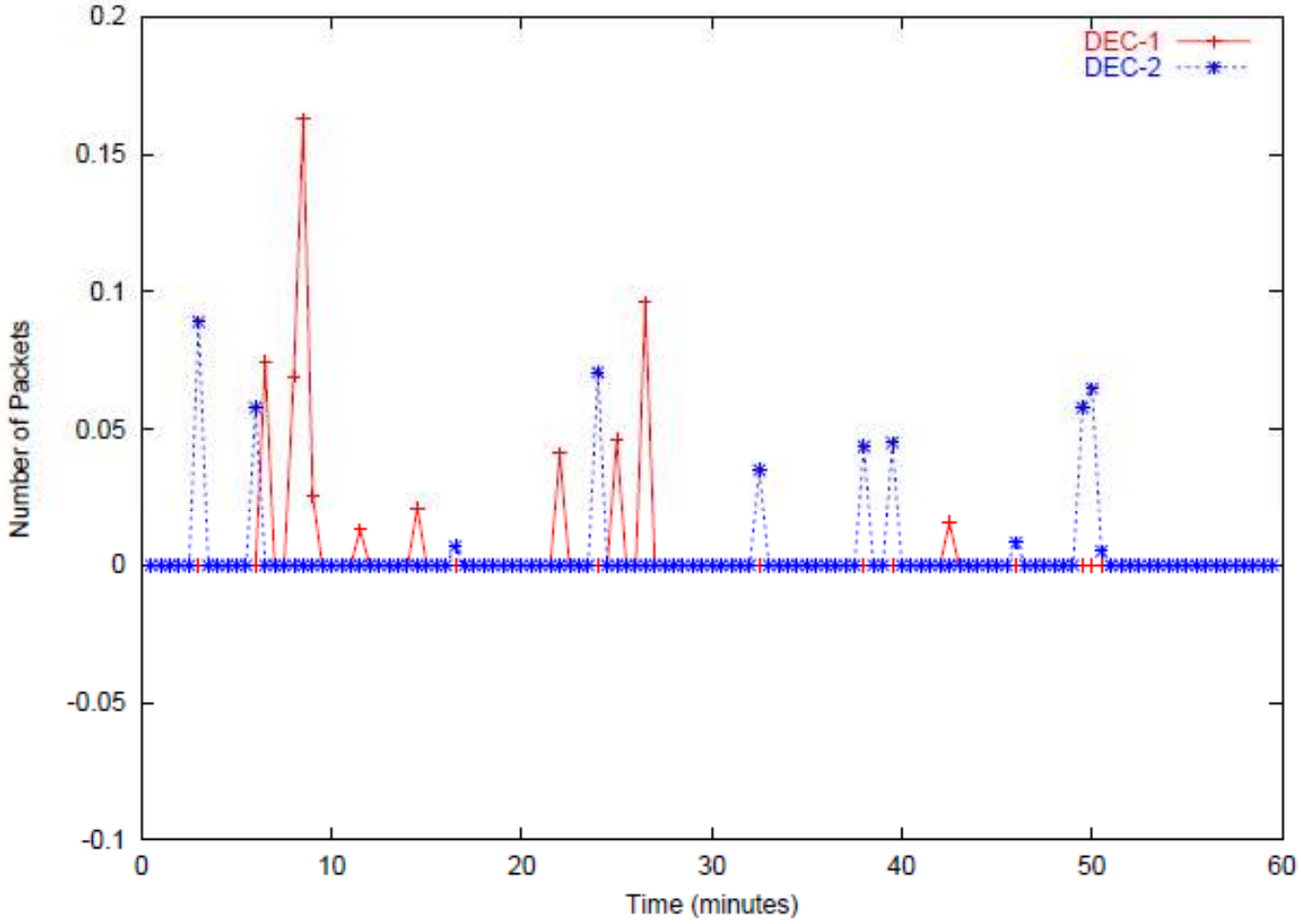


(c) UNC-out

CUSUM Test Statistic Under Normal Operation

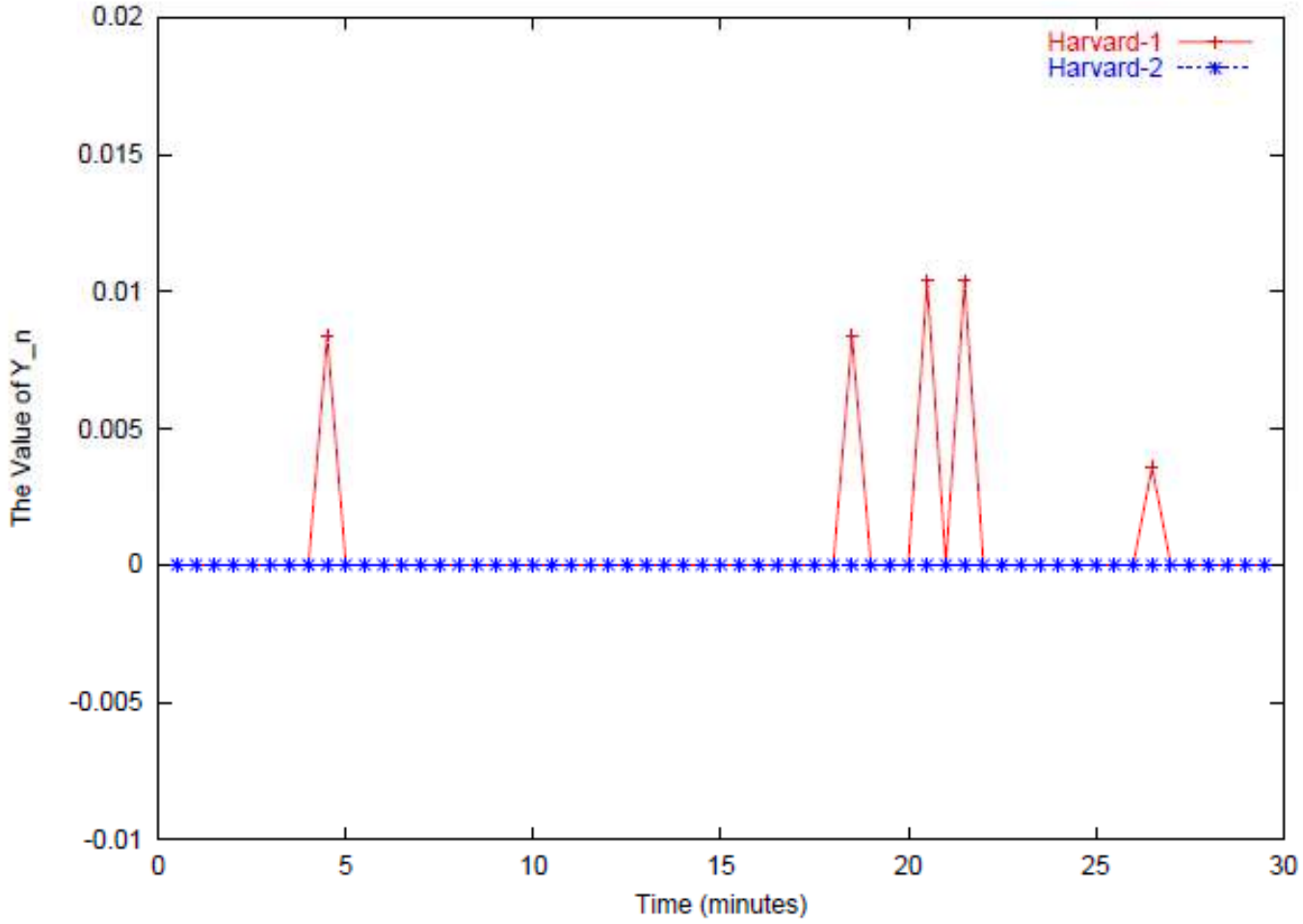
N=0.6 for FM

N=1 for LM



(a) DEC

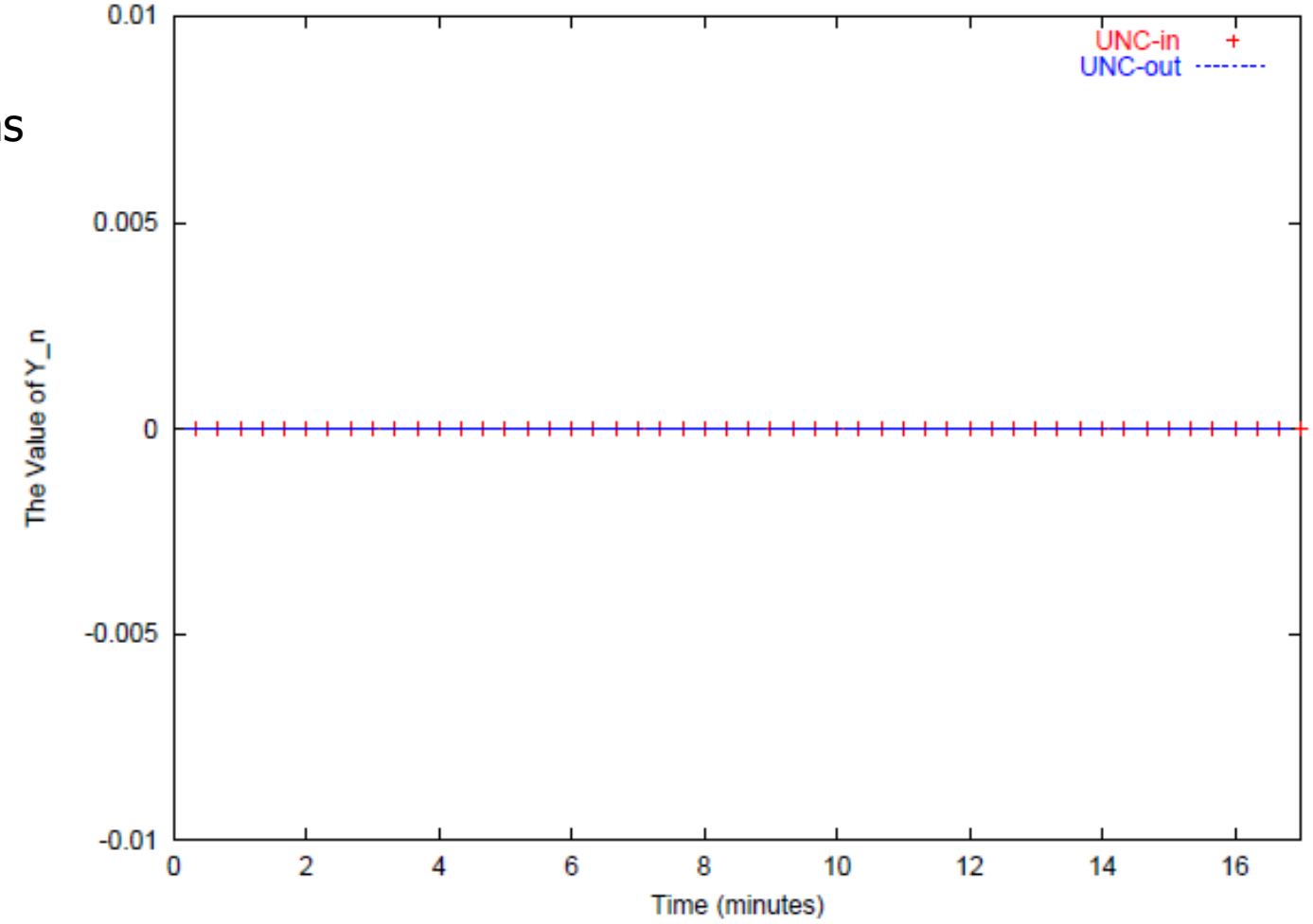
CUSUM Test Statistic Under Normal Operation



(b) Harvard

CUSUM Test Statistic Under Normal Operation

No false alarms



(c) UNC

SYN Flooding Detection

- Many DDOS attack tools developed.
- Although they use different ways to coordinate the attacks, their flooding behaviours are similar.
- Mechanism of DDOS attaks
  - Master sends control packets to the previously comprimised slaves,
  - Instructed them to target a given victim.
  - The slaves generate and send high volume of flooding mesagges.

SYN Flooding Detection

- UNC 2000 used a normal traffic
- Flooded traffic is mixed and FDS is simulated at the leaf router
- Because of the non-parametric CUSUM, the flooding pattern or behaviour does not effect the detection sensitivity
- The detection sensitivity only depends on the total volume of flooding traffic.

SYN Flooding Detection

- UNC_in inbound as Last-mile monitoring
- UNC_out outbound as First-mile monitoring

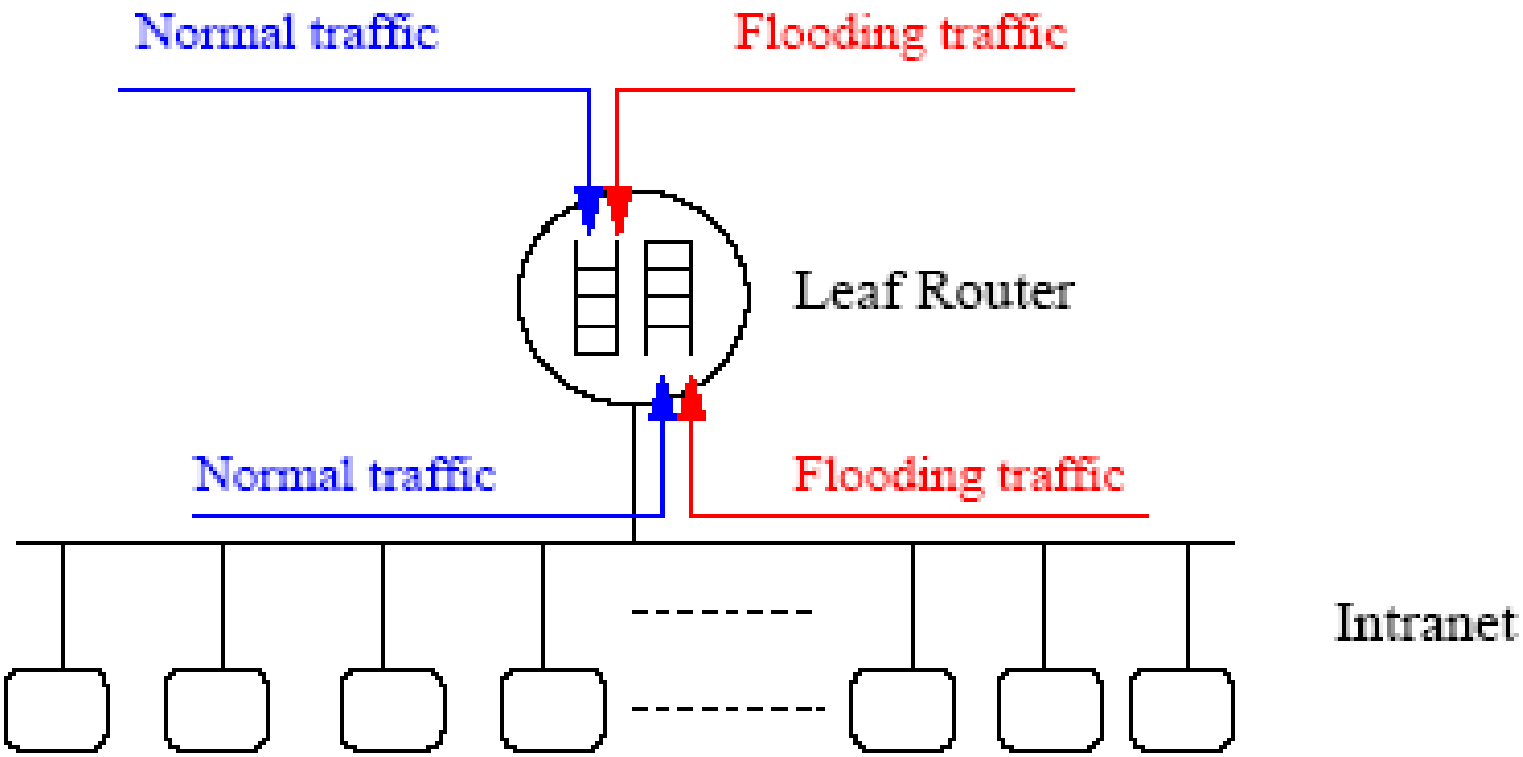
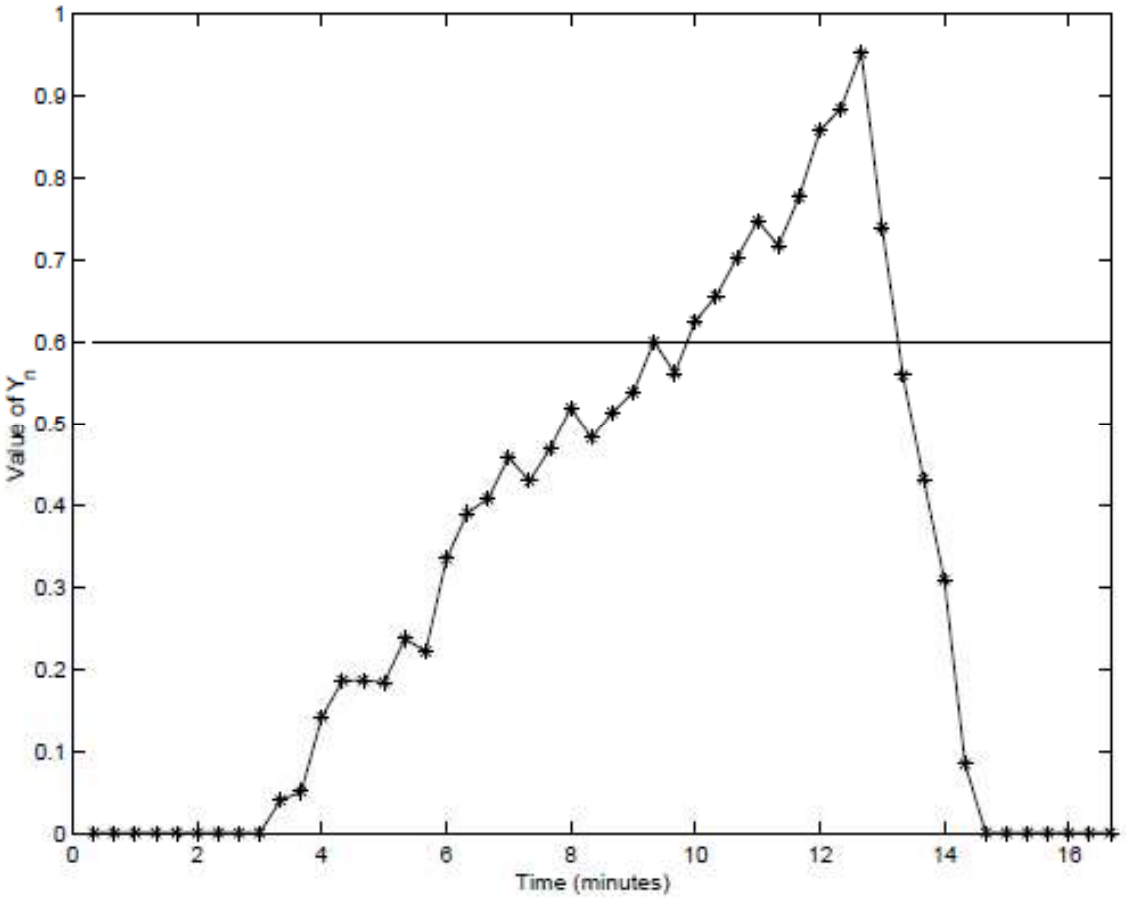


Fig. 8. The trace-simulation flooding attack experiment

SYN Flooding Detection

- The flooding traffic seen by the first and last mile is quite different.
- The traffic passing through the last mile is the aggregation of the flooding traffic.
- The aggregation makes the detection much more easier.
- The less flooding source in a network, the harder to detect the flooding
- Flood duration is set to 10 minutes,
- Attacks begin randomly between 1 and 5 min.

SYN Flooding Detection (First Mile)

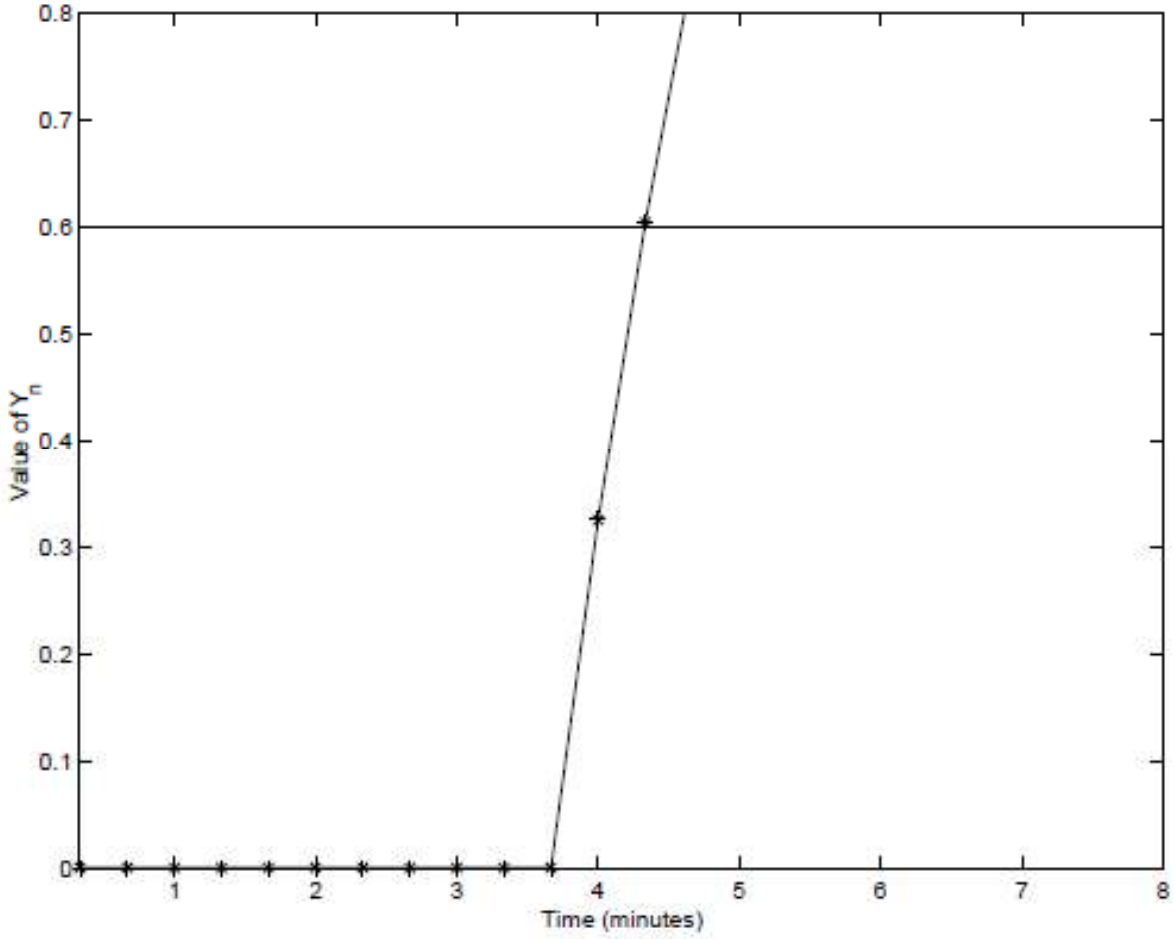


(a) 35 SYNs per second

Threshold: 0.6

Detection takes 6 minutes.

SYN Flooding Detection (First Mile)

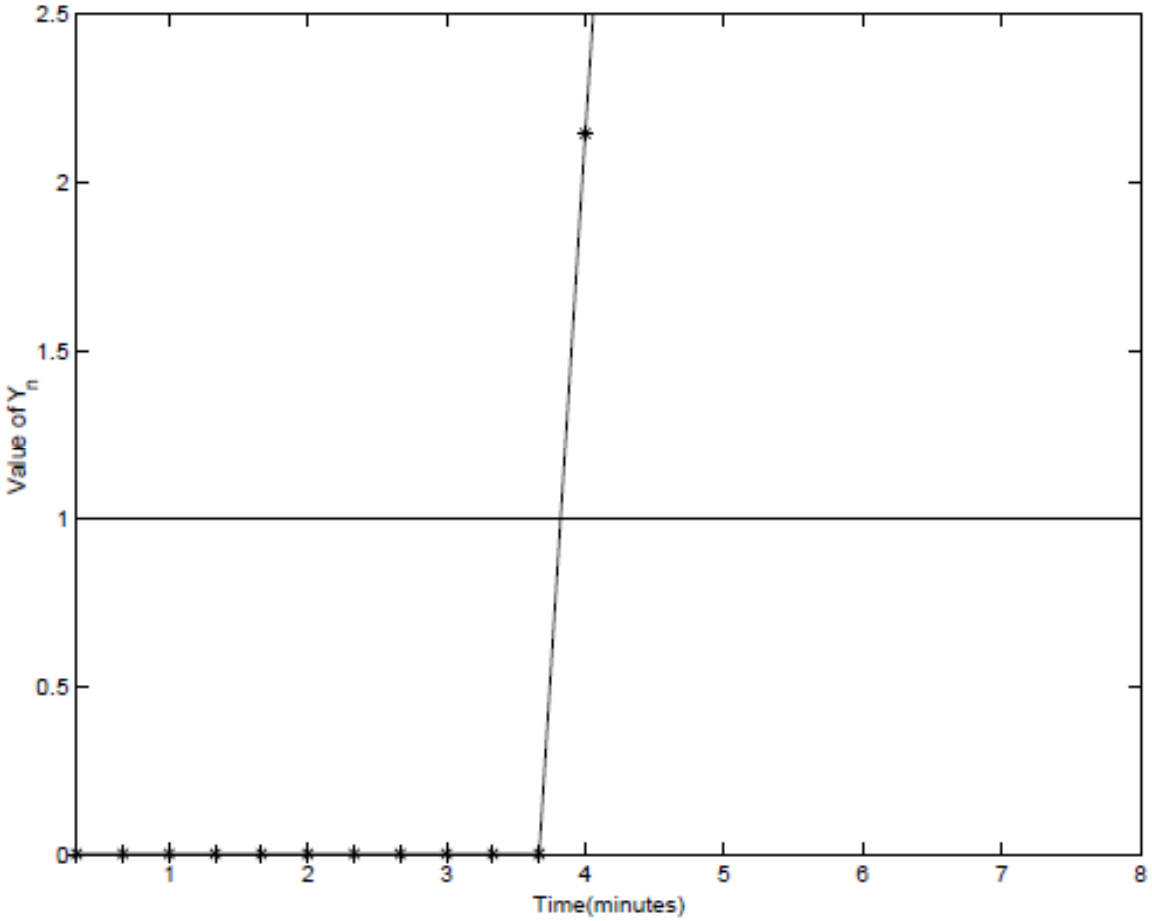


(b) 80 SYNs per second

Threshold: 0.6

Detection takes 40 seconds.

SYN Flooding Detection (Last Mile)



(c) 500 SYNs per second

- Threshold: 1
- Detection takes 20 seconds.
- Once the flooding attack detected, protection ssytem can be triggered.

TABLE II

DETECTION PERFORMANCE OF THE FIRST-MILE FDS

f_i (SYNs/s)	Detection Prob.	Detection Time
33	70%	24.36
35	100%	17.25
40	100%	9.2
50	100%	4.75
60	100%	3.0
70	100%	2.4
80	100%	1.8
90	100%	1.2
100	100%	1.0

Related Work
SYN flood defense categories

- Firewall based
- Server based
- Agent based
- Router based

Firewall based

- Examples: SYN Defender, SYN proxying
- Filters packets and requests before router
- Maintains state for each connection

- Drawbacks: can be overloaded, extra delay for processing each packet

Server Based

- Examples: SYN Cache , SYN cookies

- SYN cache receives packets first and then uses a hash table, to partially store states,
- Removes the need to watch half open connections
- Implemented in LINUX

SYN kill

- SYN kill monitors the network and if it detects SYNs that are not being acked,
- It automatically generates RST packets to free resources,
- It classifies addresses as likely to be spoofed or legitimate…

MULTOPS

- Monitors the packets going to and from a victim
- Blocks IPs from outside of network…

- limiting IP range of attack.

Route-based Distributed Packet filtering

- Uses packet information to determine if packet arriving at router has a spoofed Source / Destination addresses

- Results show many packets can be filtered and those that can't can be traced back easily

Future Work

- Any ideas on how to break the SYN-FIN pair scheme??
- SYN-FIN detection paralyzed is the attacker sends SYNs and FINs
- Just send FINs along with the SYNs…
- Will result in more traffic…

Conclusion

- SYN flooding detection installed at leaf router
- FDS is stateless and low computation overhead
- In-sensitive to the site
- Does not under mine the end-to-end TCP performance.