# TCPtransform
## (Offline version of TCPopera)

Seung-Sun (Gary) Hong, S. Felix Wu

Security Laboratory
Computer Science Department
University of California - Davis

# Outlines

- Motivations
- Related work
- TCPtransform/TCPopera development
- Design & Implementation
- Validation tests
  - TCPtransform
    - FTP traffic reproduction
  - TCPopera
    - Interactive traffic reproduction, IPS testing
- Conclusions & Future directions

# Motivations

- Industrial request
  - Having test traffic for security products
  - In-line device testing, e.g. IPS, firewall, router
- Internal request
  - Replaying traffic captured from MINOS honeypot on DETER
  - UCDavis is one of major participants in DETER project which is a large-scale network emulation environment for security protocol/product testing.

# Motivations

- Limitation of conventional trace replay tools
  - Not capable of stateful emulation of TCP connections
  - Inconsistent data/control packets generation
    - E.g. generation of ghost packets
  - No good for in-line device testing such as IPS testing
- Live security test environments require
  - Realistic test traffic and packet contents
  - more interactive traffic replay approach

# Related work

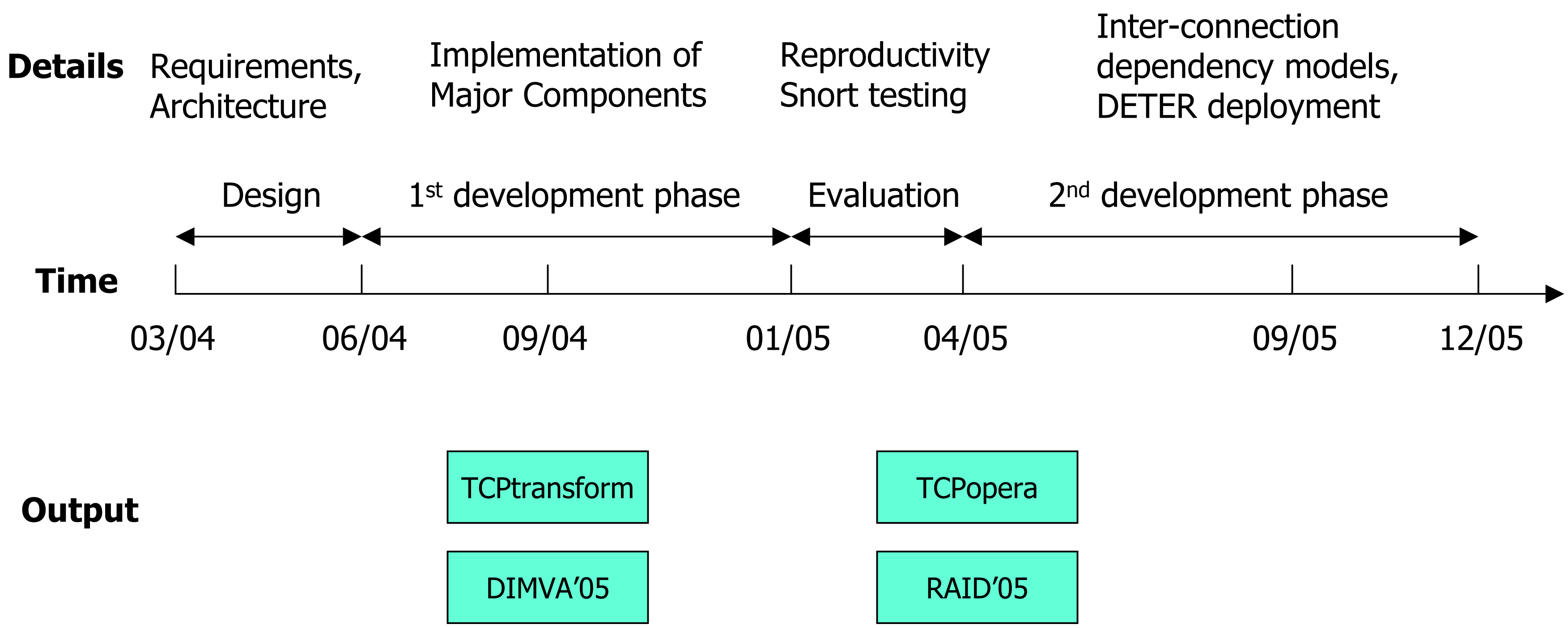
- Trace-based traffic replaying
  - Easy to implement and mimic system behaviors
  - Real traffic, sufficient diversities
  - Hard to adjust trace for various test conditions
    - Assuming the test condition is the same as the time at the trace was recorded
- Analytic-model based traffic generation
  - Easy to control/adjust traffic generation models
  - Statistically identical to traffic models.
  - Hard to support trace contents for security test environments

# Trace-based traffic replaying

- TCPreplay/Flowreplay
  - Static trace replaying mainly for NIDS testing
  - Flowreplay is a TCP client emulator
- TCPivo
  - High-performance trace replay tool
  - I/O management, Timer accuracy, Null-padding payload
- Monkey
  - HTTP emulator for a Google server
  - Monkey See for TCP tracing, Monkey do for TCP replaying
- Tomahawk
  - A tool for testing in-line blocking capabilities (IPS)
  - Operable across layer-2 connection.

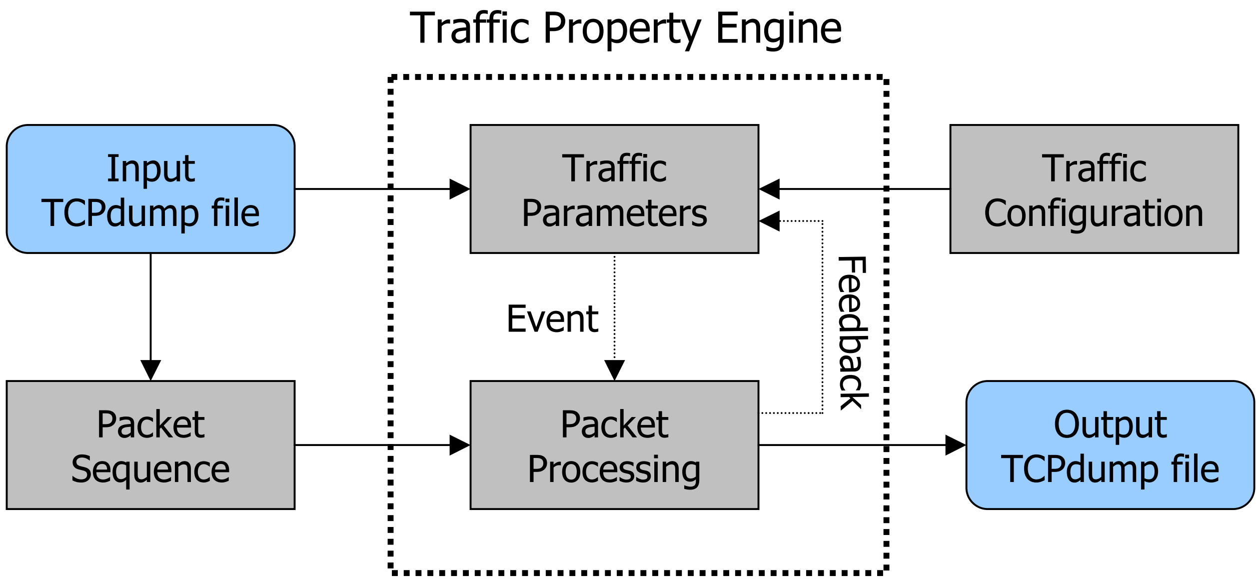
# TCPtransform/TCPopera development

- Milestone of TCPopera development



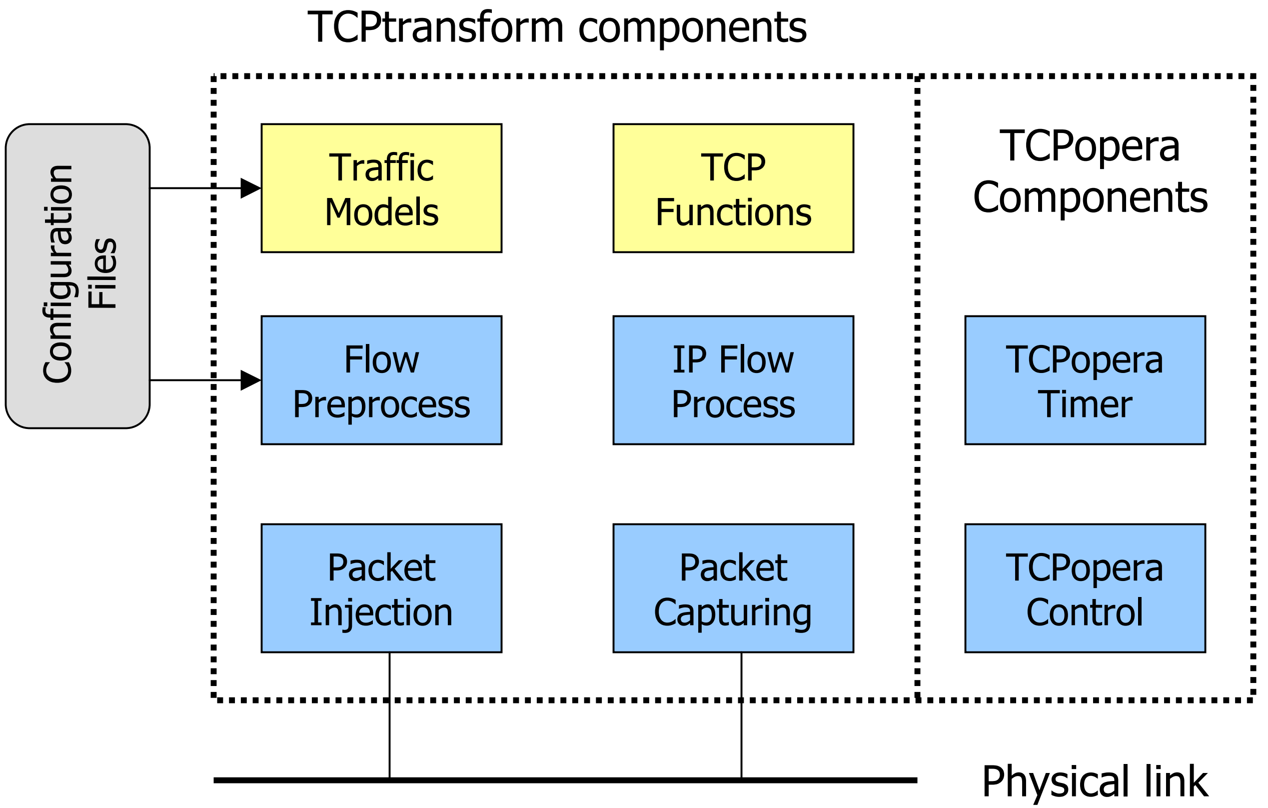
# Design & Implementation

- Property-oriented trace replaying
  - Extract traffic parameters from Input trace records through the reverse-engineering
  - Adjust traffic parameters according to test conditions
  - Feed new traffic parameters to input packet sequence



# TCPtransform Components

- TCPopera/TCPtransform Major Components



# TCPtransform Components

- Flow Preprocess
  - Preparing IP flows
  - Extraction of TCP connection and IP flow parameters
    - RTT, transmission rate, packet loss rate, path MTU
  - Address remapping, ARP emulation
- IP Flow process
  - Creating a POSIX thread for each IP flow
  - TCP control block emulation
- Traffic Models
  - TCP parameters for the initiation of TCP control blocks
  - Gap-based packet loss model

# TCPtransform Components (Cont'd)

- TCP Functions
  - Based on BSD4.4-Lite release (1994) - TCP Reno
  - 8 TCP timers
    - Slow timer (500ms), Fast timer (200ms)
  - Timeout & Retransmission
    - RTT measurement
  - Fast Retransmit & Fast Recovery
  - Flow & Congestion Control
- Packet Injection/Packet Capturing
  - Libnet and Pcap
  - IP/TCP checksum recalculation if a packet is modified

# TCPtransform Validation

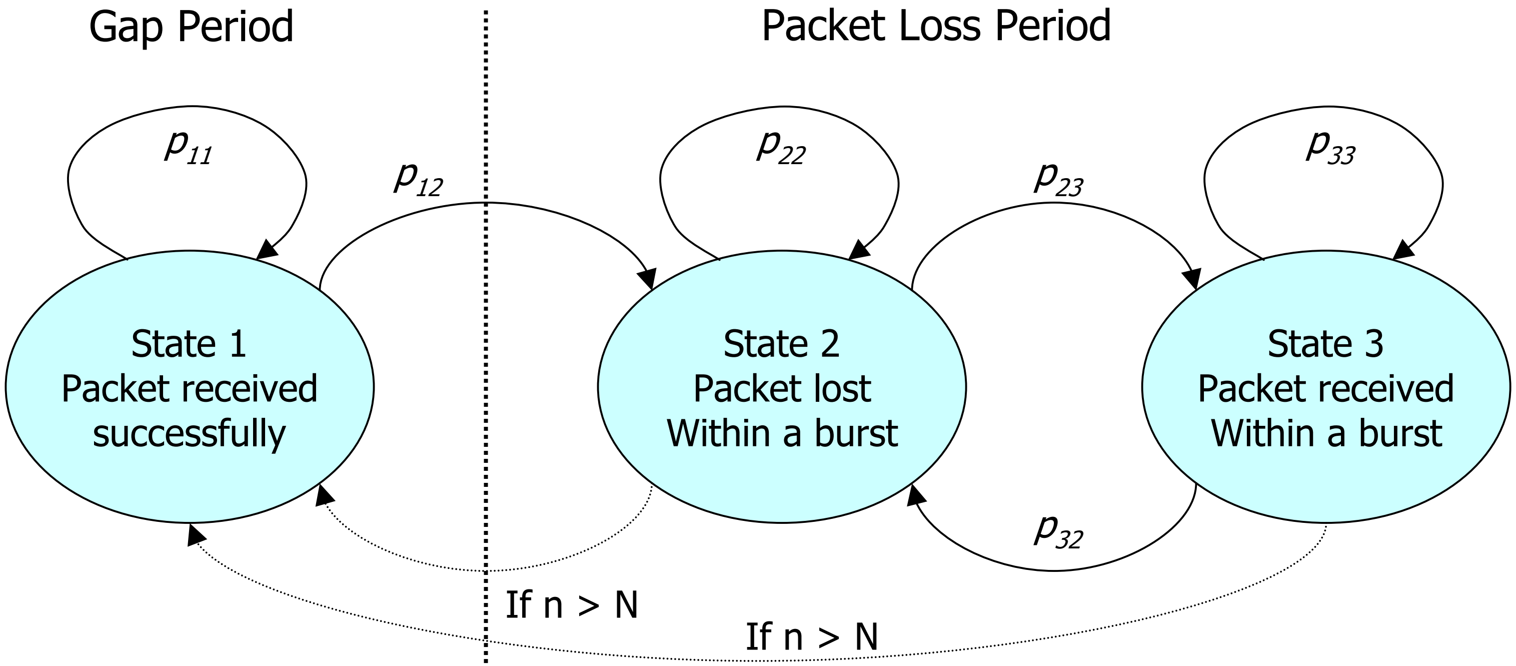
- FTP traffic reproduction
  - Imitating a FTP connection to download a 8M file from 3 different public GNU servers
  - Sampled over 10,000 FTP connections from each server

| Name | Location | Host name (IP address) |
|---|---|---|
| Berlin | German | ftp.cs.tu.berling.de (130.149.17.12) |
| NCTU | Taiwan | ftp.nctu.deu.tw (140.113.27.181) |
| Charlmers | Sweden | ftp.chl.charlmers.se (129.16.214.70) |

- Test setup
  - Collect an input tcpdump file from a local FTP server.
    - To remove any noise, we directly connect the client machine to the FTP server.
  - TCPtransfrom reproduced FTP connections for each server

# TCPtransform Validation

- Gab-based Packet Loss Model




N: Maximum number of packets in Packet loss period,
n: number of packets in Packet loss period

# Q distribution

- $\chi^2$-like test to compare the similarity between long-term and short-term profile.
- Partition sample space S into $bin_i$.
- $N$ : Total number of events
- $Y_i$ :Number of event occurrences for $bin_i$.
- $P_i$ : Probability of event occurrences for $bin_i$ ($Y_i / N$)
- $Y_i'$ and $N'$ for short-term profile.

$$Q = \sum_{i=1}^{k} \frac{\left(Y_i' - N' \times p_i\right)^2}{N \times p_i}$$

# Test conditions

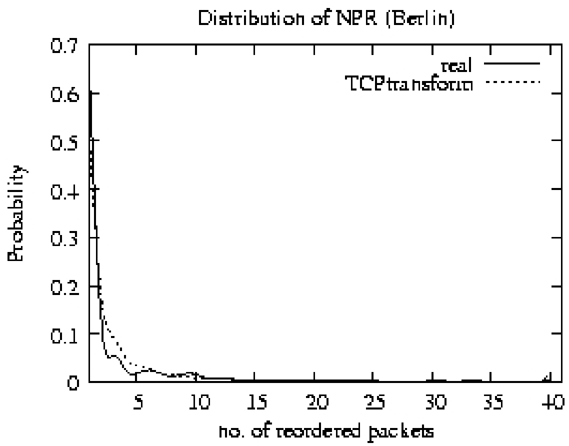
- 4 traffic variables

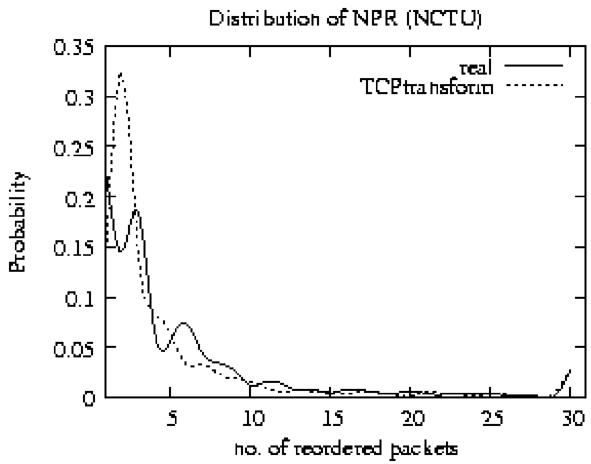
| Server | | Berlin | NCTU | Charlmers |
|---|---|---|---|---|
| Packet loss rate | | 0.0003 | 0.0002 | 0.0001 |
| Loss burst size (Pareto) | Shape | 1.1 | 1.2 | 1.1 |
| | Min | 1 | 1 | 1 |
| Packet burst size (Pareto) | Shape | 1.1 | 1.1 | 1.1 |
| | Min | 20 | 20 | 20 |
| RTT (msec) | Avg. | 152 | 260 | 163 |
| | Stdev | 9.161 | 14.881 | 0.977 |

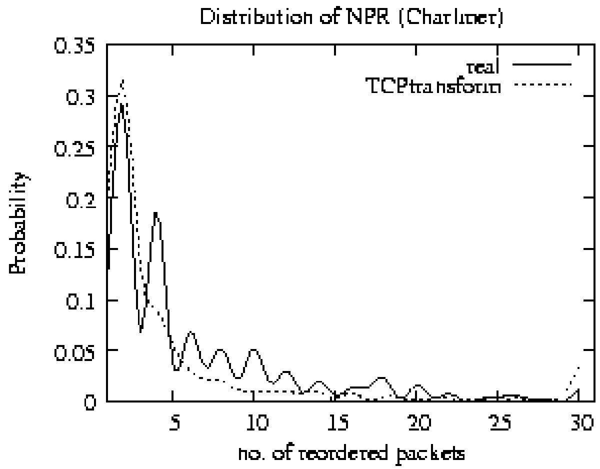
# Test results

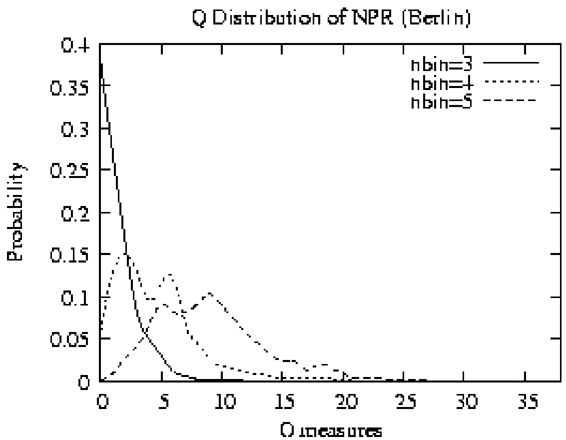
- NPR (Number of Packet Reordering)

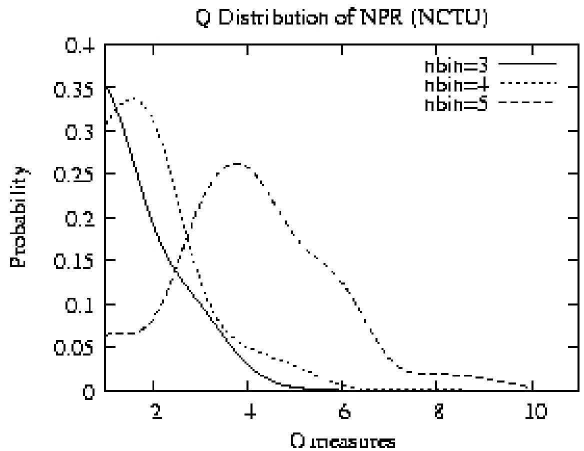
Berlin, German | NCTU, Taiwan | Charlmers, Sweden

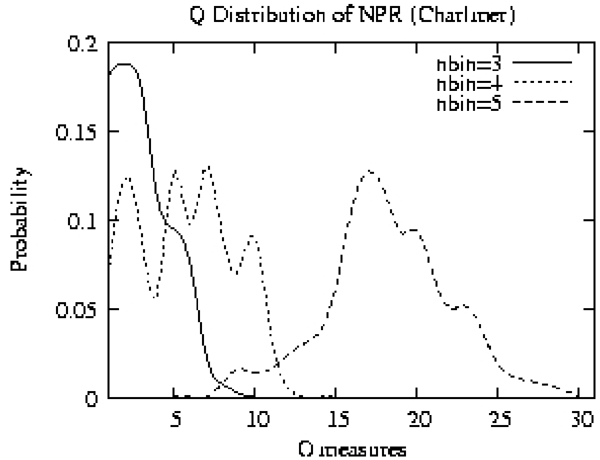


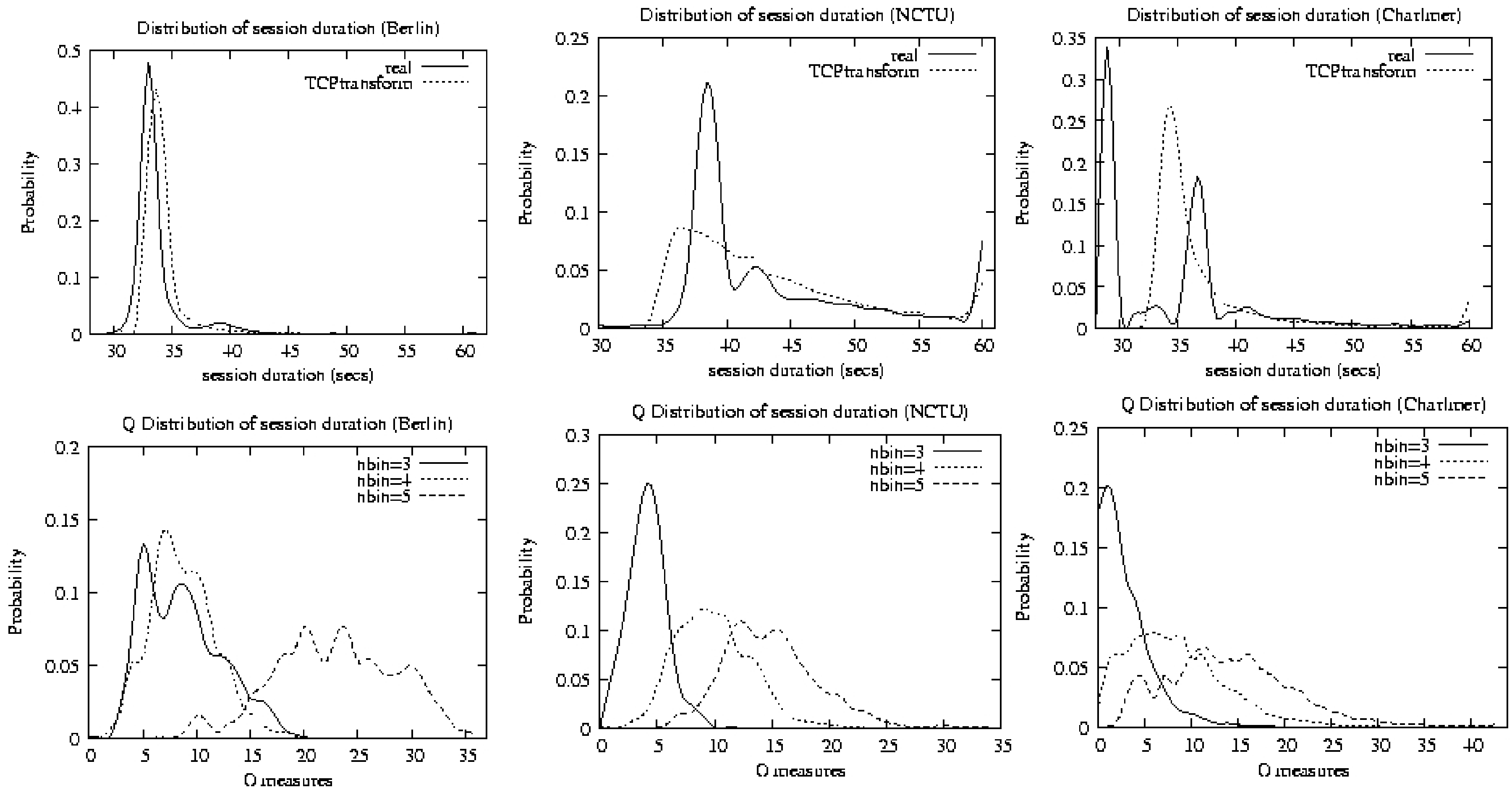



# Test results (cont'd)

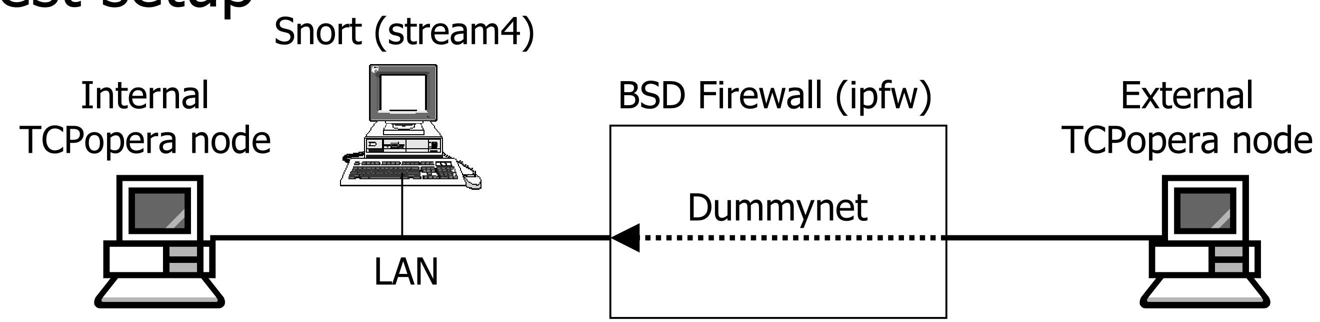
- Session Duration

Berlin, German | NCTU, Taiwan | Charlmer, Sweden



# TCPopera Validation

- Test setup




- TCPopera nodes
  - 2 GHz Intel Pentium 4, 768MB RAM
  - Internal: Redhat 8 (2.4.18), External: Redhat 9 (2.4.20)
- Network Emulator
  - 455MHz Pentium II Celeron, 256MB RAM
  - FreeBSD5.0, IPFW (with Dummynet)
- Snort 2.3
  - 3.2 GHz Intel Pentium 4 Processor, 512MB
  - Slackware 10.0 (2.4.26)
  - All Snort rules are enabled including the Stream4 analysis

# TCPopera traffic reproduction

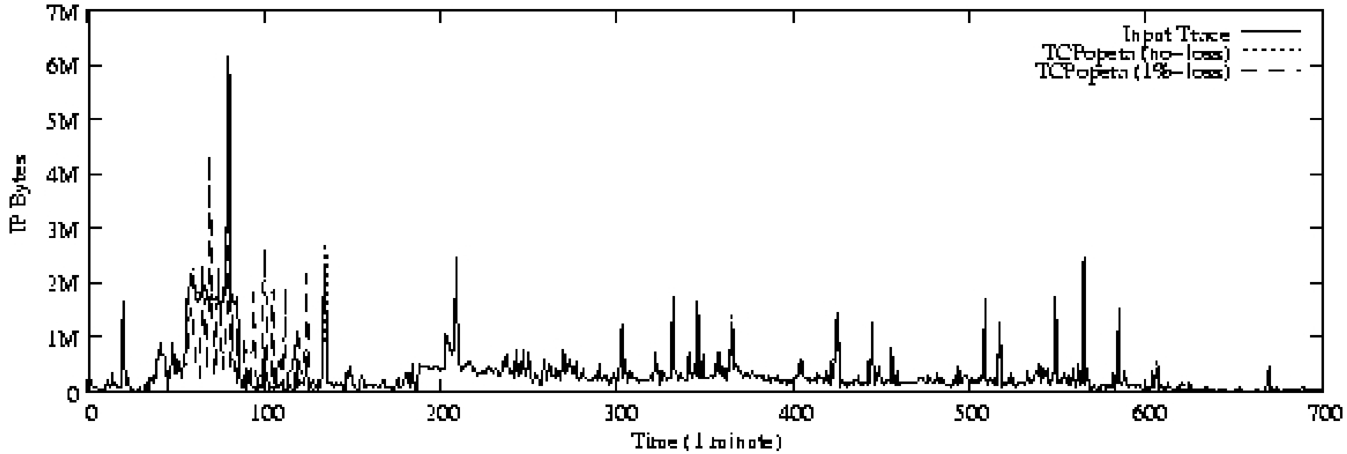
- DARPA IDEVAL99 (first 12 hours of 03/29/99)

| **Category** | | **Input trace** | **TCPopera** | |
|---|---|---|---|---|
| | | | **No loss** | **1% loss** |
| IP | Packets | 1,502,584 | 1,552,882 | 1,531,388 |
| | Bytes | 234,434,486 | 234,991,187 | 232,145,926 |
| TCP | Packets | 1,225,905 | 1,276,195 | 1,254,762 |
| | Bytes | 194,927,209 | 195,483,762 | 192,647,088 |
| UDP | Packets | 276,286 | 276,294 | 276,234 |
| | Bytes | 39,474,602 | 39,495,286 | 39,466,797 |
| ICMP | Packets | 393 | 393 | 392 |
| | Bytes | 32,675 | 32,139 | 32,041 |
| TCP connections replayed | | 18,138 | 18,138 | 18,043 |
| TCP connections completed | | 14,974 | 14,971 | 14,796 |

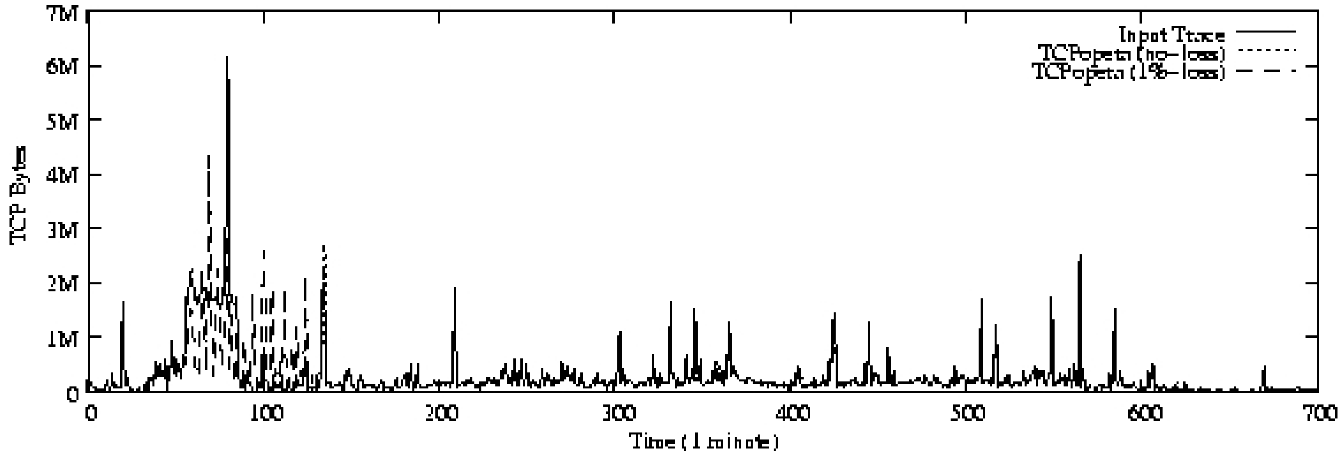
# TCPopera traffic reproduction

- Traffic volume comparison (every minute)

IP Bytes




TCP Bytes



# TCPopera validation (Snort Evaluation)

- ITRI Dataset
  - Collected for 30 minutes from a host within 140.96.114.0/24 segment in Taiwan
  - Major applications: HTTP, P2P (eDonkey), FTP
- Evaluation results

| Signature | No. of alerts | | | |
|---|---|---|---|---|
| | Input trace | TCPopera | | |
| | | No-loss | 1% loss | 3% loss |
| ICMP Destination/Port Unreachable | 5 | 5 | 5 | 5 |
| ICMP Destination/Host Unreachable | 2 | 2 | 2 | 2 |
| ICMP Destination Unreachable Fragmentation needed but DF bit is set | 1 | 1 | 1 | 1 |
| P2P eDonkey Transfer | 3 | 3 | 3 | 3 |
| (stream4) Possible retransmission detection | 38 | 212 | 200 | 181 |
| (stream4) WINDOW violation detection | 488 | 3 | 1 | 4 |
| Total | 537 | 226 | 212 | 196 |

# Snort Evaluation – stream4 analysis

- Possible retransmission detection
  - Detecting an attempt to packet replaying attack
  - TCPopera's delayed ACKs confused the stream4 re-assembler.
- WINDOW violation detection
  - Detecting an attempt to write the outside of the receiver's window.
  - Mishandling of incomplete TCP connections.
    - Mistakenly assume the connection is established.
  - Strict rules on handling RST segments.
    - No resetting TCP connection, instead update the window size an RST segment is carrying.

# Conclusions

- TCPopera does Interactive trace replaying with a stateful emulation of TCP connections.
- Initial evaluation showed a positive sign in the usefulness of TCPopera.
- Providing more methodologies for the security product evaluation.
- Deployable in a large-scale emulation environment like DETER.
- TCPopera is an on-going project.

# Future directions

- Next TCPopera development phase
  - Porting TCPopera into DETER environment.
  - More in-line devices evaluation such as IPS.
    - Adding more evasive techniques for IPS testing
  - Supporting more application-specific inter-connection dependency models
  - Adding more TCP/UDP traffic models
  - Adding a centralized TCPopera GUI to control multiple TCPopera nodes.

# Thanks & Questions

- Many thanks for paying attention to the talk.
- Any question