

# **SIP Intrusion Detection and Prevention: Recommendations and Prototype**

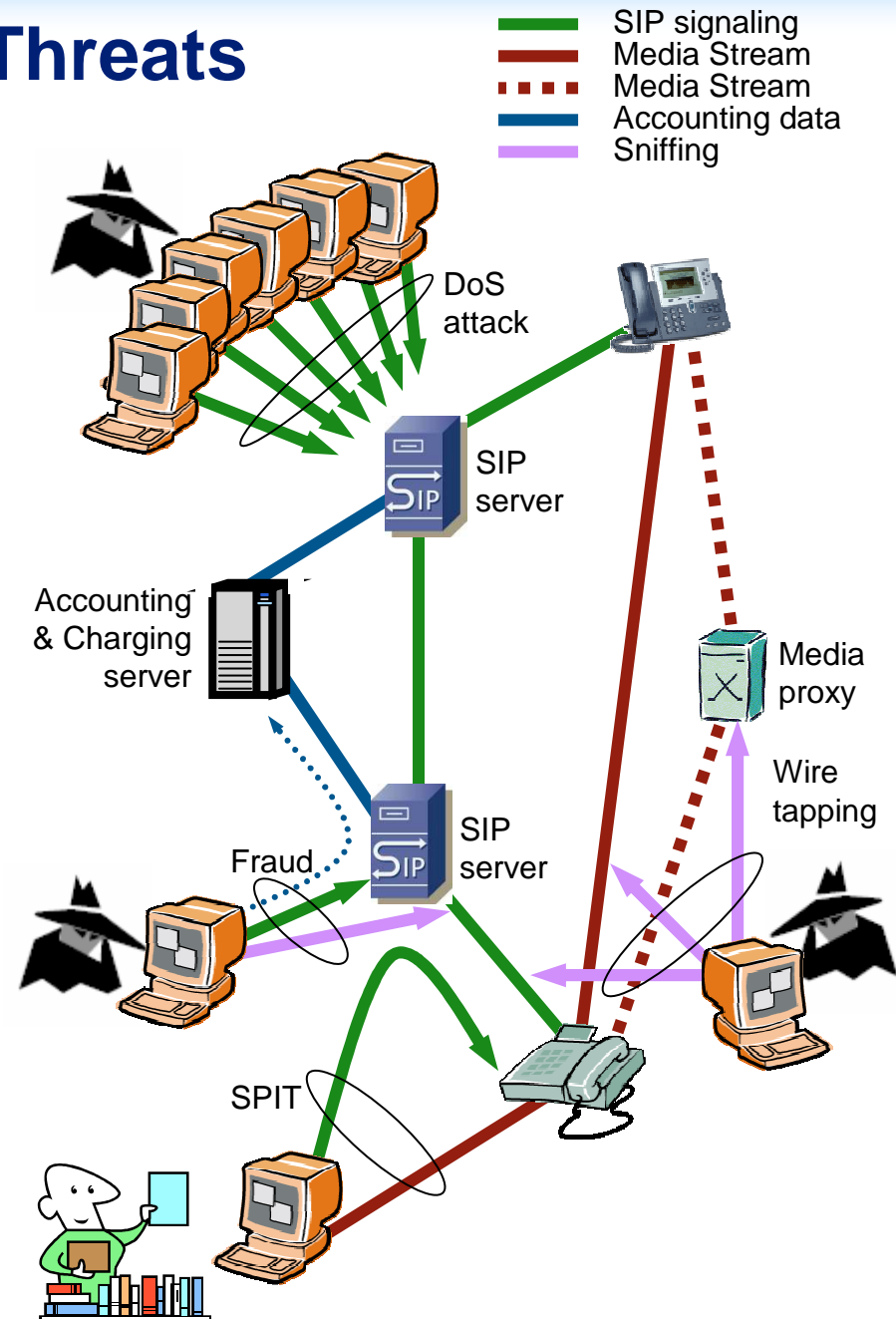
**Saverio Niccolini**

**NEC Europe Ltd., Network Laboratories**

**saverio.niccolini@netlab.nec.de**

# Security Threats

- **VoIP protocols (like SIP) are vulnerable to many attacks**
  - Interruption of Service attacks (Denial of Service, DoS)
    - attacks against infrastructures and terminals
  - Social attacks (SPam over Internet Telephony, SPIT)
    - disturbances and interruptions of work by ringing phone for unsolicited calls
  - Fraud
    - placing calls on other customer's bills, etc.
  - Interception (wire tapping) and Modification of calls
    - conversations may be intercepted (lack of confidentiality)
    - conversations may be modified (lack of integrity)
- see VOIPSA taxonomy



# Security Threats: available solutions?

- **Standard security mechanisms**
  - See talk of this morning by Cullen Jennings
- **Intrusion Detection and Prevention systems (IDS/IPS) needed on top of such standards**
  - to better secure the deployment
    - block the attackers bypassing security mechanisms
  - should control both signaling and media path
    - correlation needed among the two paths
      - media communication can be routed independently of the call setup path
  - three common types of IDS systems
    - host-based
    - network-based
    - stack-based
  - IPS systems can take immediate action
  - techniques
    - signatures (knowledge-based)
    - statistical observation (behavior-based)

# VoIP Intrusion detection and prevention

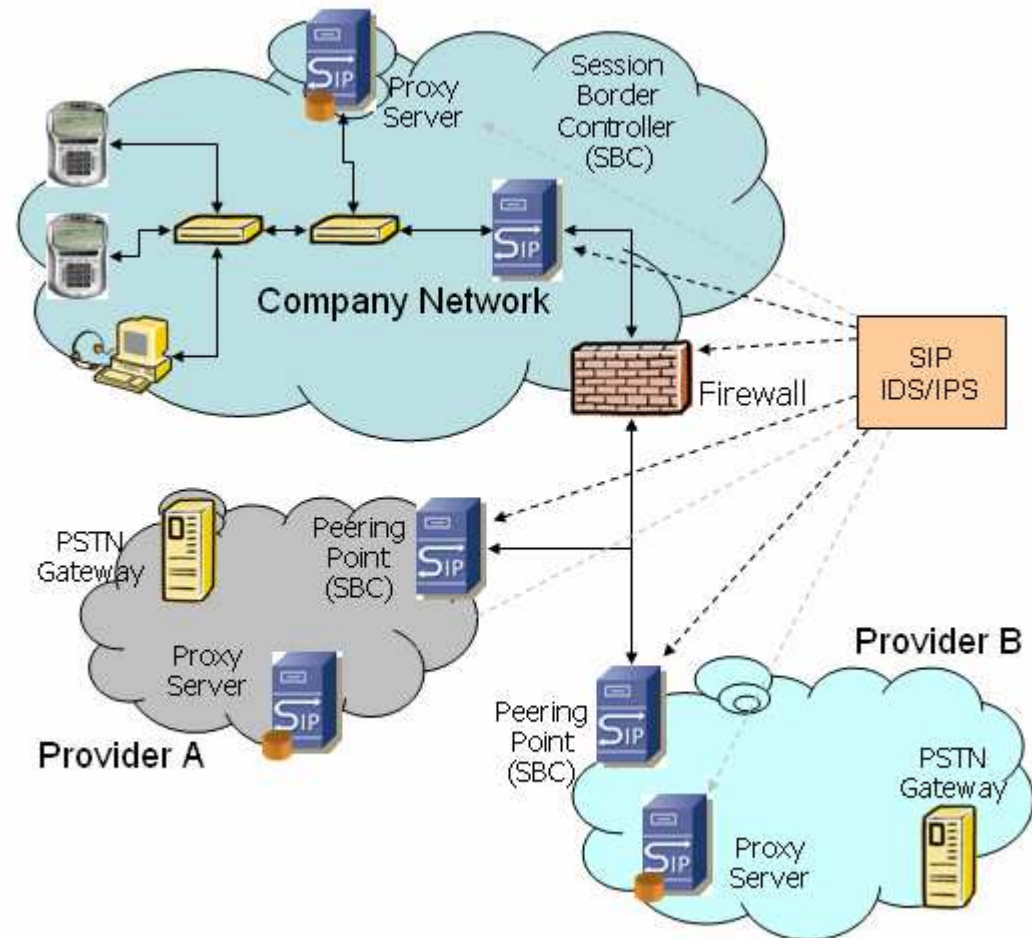
- **Attacks to traditional server oriented applications**
  - the security target is only the server
    - e.g. HTTP, FTP, E-MAIL
- **VoIP deployments have different characteristics**
  - a much higher number of systems to be protected
    - VoIP servers, e.g. Proxies and Gateways
    - terminals
  - stricter requirements in terms of security checks
    - no need of sending high rates of messages
    - few messages are able to cause crashes or reboots

# Which IDS/IPS for VoIP?

- **Network-based IDS**
  - good matching of requirements
- **Host-based/Stack-based**
  - not scalable (unless you want to protect only some servers)
- **Techniques**
  - knowledge-based first
    - blocking malicious traffic
  - behavior-based second
    - statistical analysis
- **Knowledge-based techniques share info with behavior-based**
  - writing info in a shared memory area
    - IP addresses
    - SIP URIs
    - Ports
    - Message rate
  - increase in scalability (message known to be malicious are already filtered out)
  - decrease in false positives

# Network-based IDS

- **Must be implemented in devices able to observe the traffic to be analyzed**
  - the entry point of the SIP network is the most suited point
- **SIP devices**
  - SIP-aware firewall
  - Peering points
    - Session Border Controllers (SBCs)
      - B2BUA in SIP
    - SIP gateways

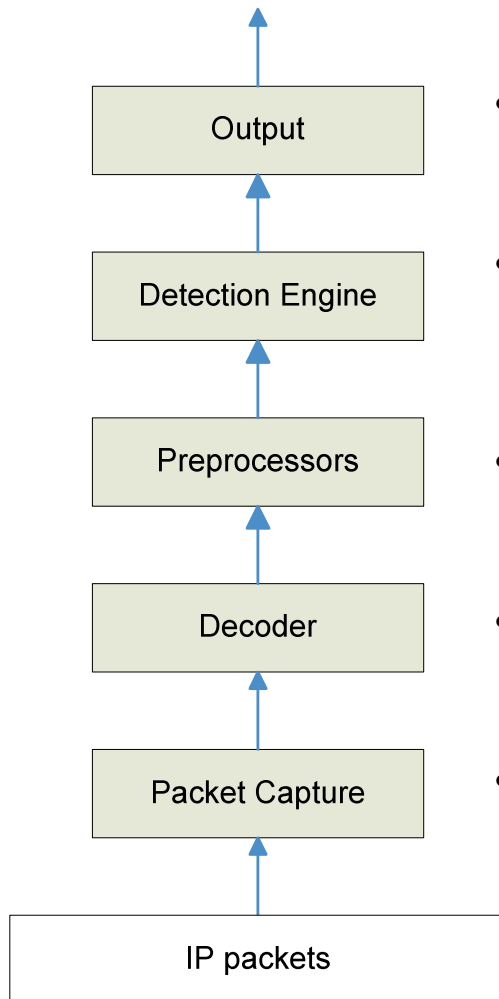


# SIP IDS/IPS prototype

- We used Snort\_Inline framework
- Snort is capable of performing real-time traffic analysis and packet logging
  - acts only as IDS (can only detect. not block packets)
  - it perform protocol analysis and content searching/matching
- Using Snort\_Inline as IPS
  - modification and blocking of packets (accepts packets from IPTABLES using the ip\_queue module)
  - works in bridge modality, invisible to attackers

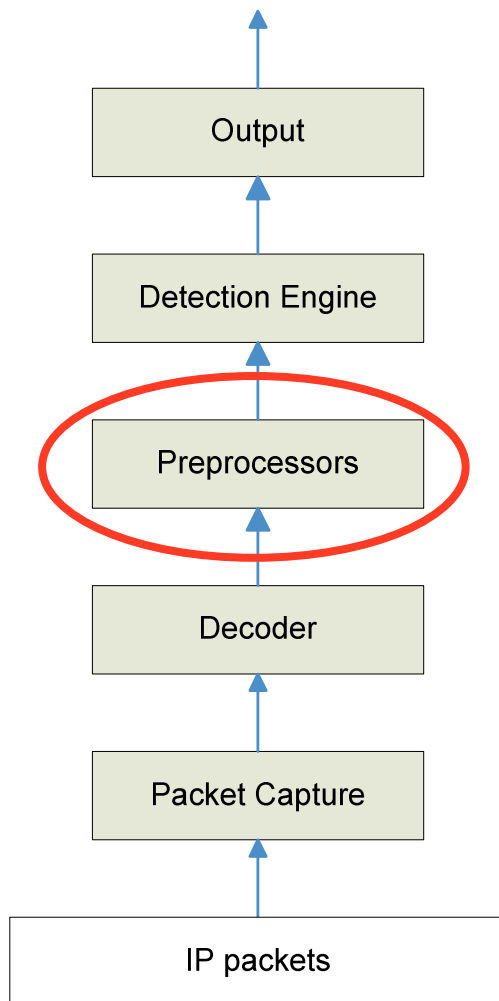


# Snort architecture



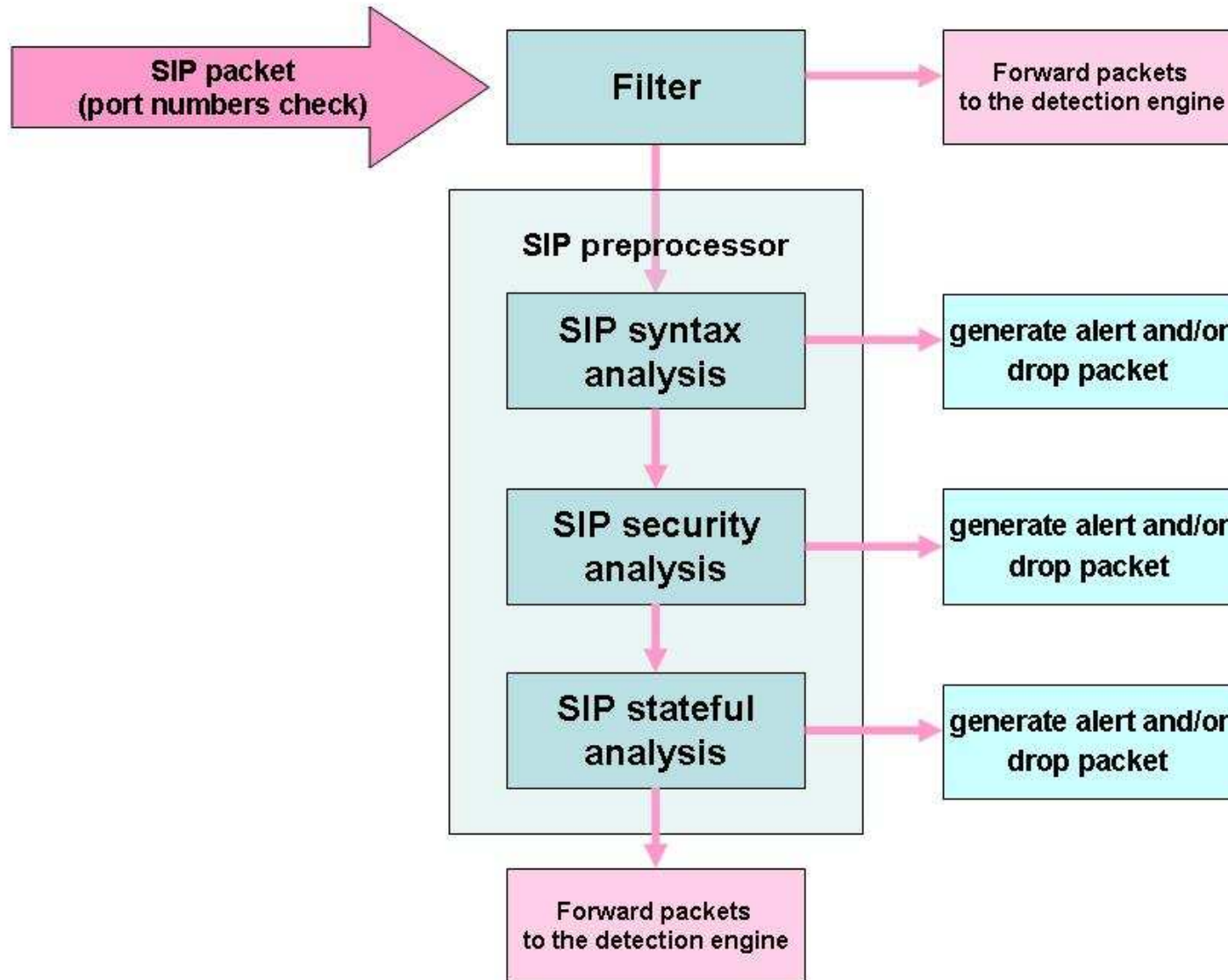
- **Output block**
  - manages the log output
  - output log is configurable (e.g. text files, databases or user-defined)
- **Detection Engine block**
  - analysis of protocols of any layer using signatures and rules
  - stateless mode
  - rule sets defined before start time
- **Preprocessors block**
  - analysis of protocols of any layer using custom made C/C++ programs
  - stateful mode
- **Decoder block**
  - syntax analysis at layer 2, 3 and 4 of the IP packet (MAC, IP and TCP/UDP)
  - Layer 2, 3 and 4 headers are inserted in a shared portion of memory
- **Packet Capture block**
  - capture the packets, it uses either libpcap or iptables depending on the Snort mode

# SIP IDS/IPS prototype software (I)



- **SIP preprocessor wrote from scratch**
- **It uses oSIP libraries**
- **What it does**
  - **SIP syntax analysis (parsing)**
  - **Security check**
    - looks at mandatory fields in a SIP message
  - **Stateful analysis (soft states are used)**
    - it computes message rate and compare them to a threshold
      - by looking at SIP URIs
      - by looking at IP addresses
    - it is customizable to prevent specific DoS/SPIT attacks
  - **Generation of logs of suspicious packets in a tcpdump format**
    - can be later analyzed using Ethereal
    - can be exported to correlate with media analysis

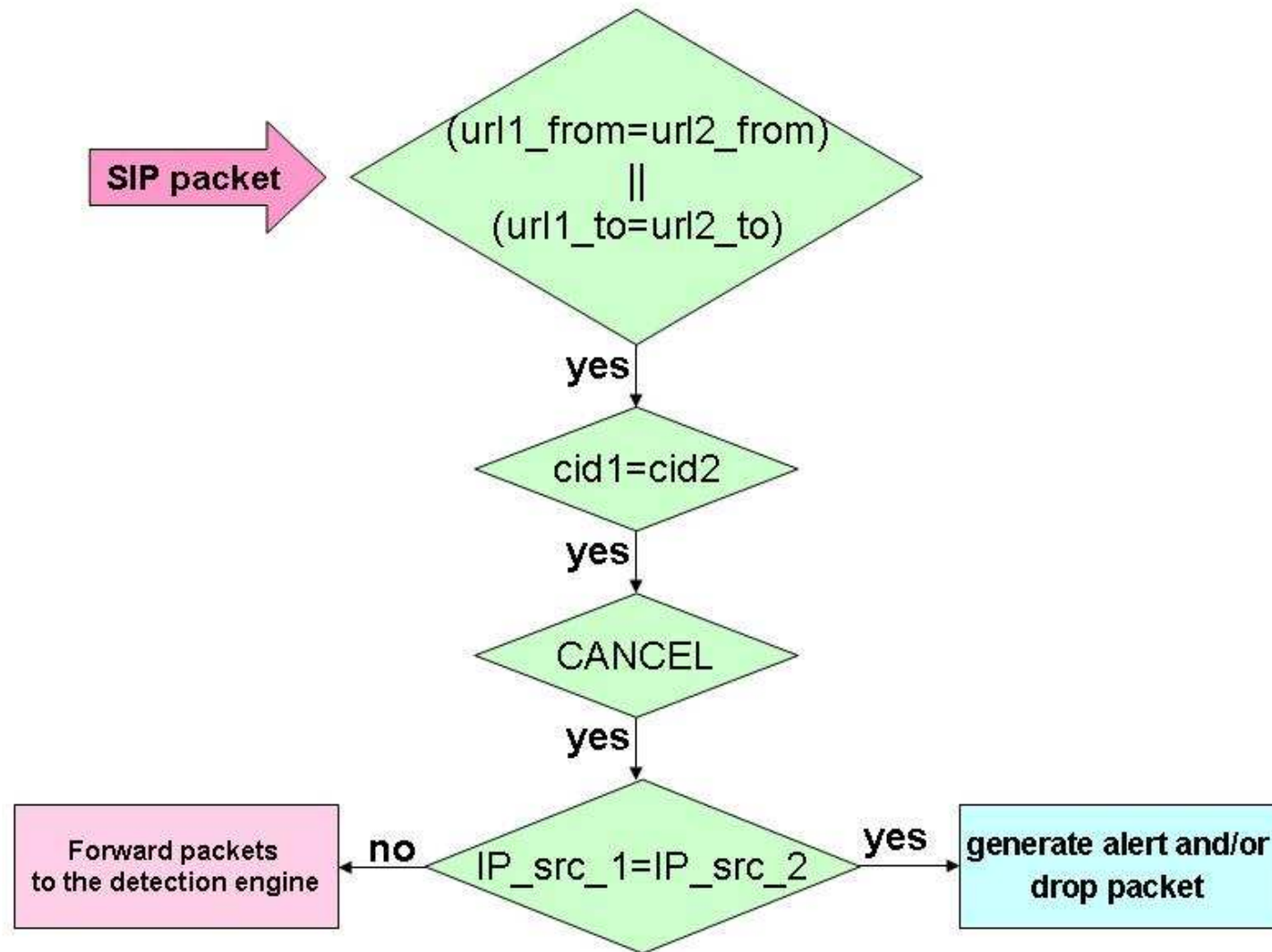
# SIP IDS/IPS prototype software (II)



# Examples of additional features implemented

- **Blocking SPIT attacks**
  - checking the INVITE rate
    - of a SIP URI
    - of a source IP address
      - configurable thresholds if UA or Proxy Server
- **Blocking DoS attacks**
  - checking total SIP message rate
    - of a SIP caller
    - of a source IP address
      - configurable thresholds if UA or Proxy Server
- **Blocking Call Tear-Down attacks**
  - checking that CANCEL/BYE comes from one of the parties involved in the call
    - looking at IP addresses
  - this attack can be done only spoofing To; From; Call-ID fields

# Blocking Call Tear-Down attacks

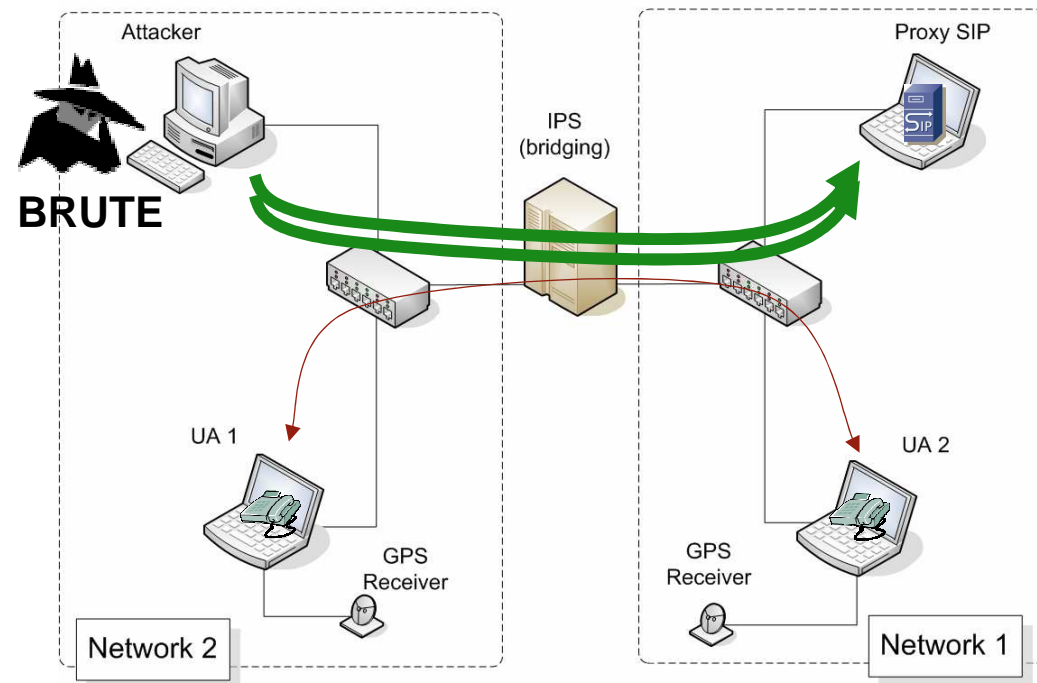


# Prototype testing

- What happens to QoS of communications?
  - We stressed the IDS/IPS knowledge-based techniques generating malformed messages with different rates
    - wrote SIP plug-in generator for BRUTE
      - high performance packet generator
      - precise message rate
  - RTP media session between UA1 and UA2 at the same time
    - mean end-to-end delay
    - packet losses
    - mean jitter
    - packets with jitter > 50 ms

— SIP signaling  
— Media Stream

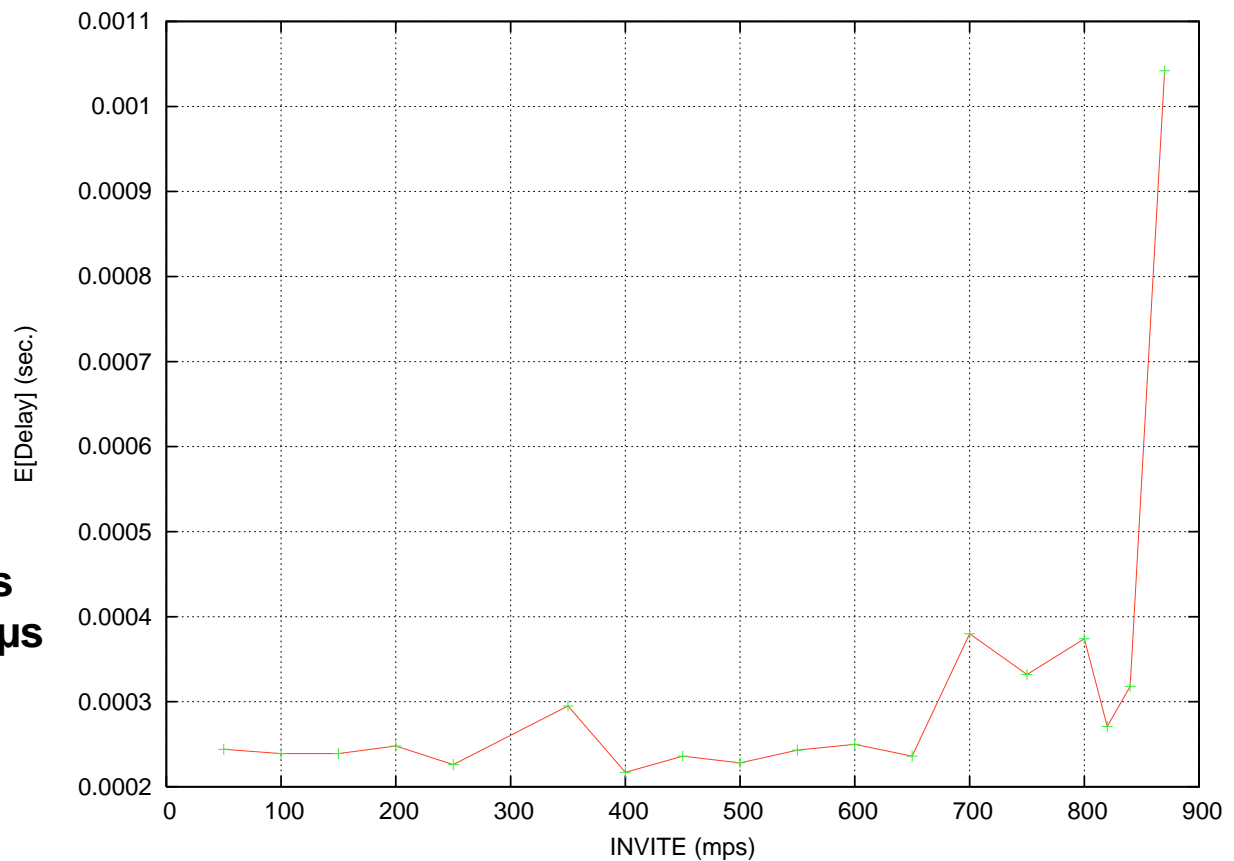
- Experimental set-up
  - SIP proxy (SIP Express Router, SER)
  - SIP UAs (Kphone)
    - running on Linux OS
    - GPS-synchronized to compute One Way Delay
  - Attacker (BRUTE generator)



# End-to-end delay

- **Message generation rate lower than 860 mps**
  - mean end-to-end delay introduced by IDS/IPS ok
- **Message generation rate higher than 860 mps**
  - ip\_queue module receiving packets from the iptables becomes full

- **Other parameters**
  - rate < 860 mps
    - no packet losses
    - mean jitter: 180  $\mu$ s
    - jitter > 50 ms: 10 out of 15000 packets



## Conclusions/Future work

- **Guidelines for IDS/IPS for VoIP deployments**
- **Prototype implementation on top of Snort framework**
- **SIP plug-in for high performance tool (BRUTE)**
- **Evaluation of prototype implementation**
  
- **Future work**
  - **hybrid solution with knowledge-based checks implemented at OS kernel level (modification to iptables)**
    - **behavior-based techniques still in user space because of the flexibility required**
  - **modeling VoIP-specific DoS attacks**
  - **modeling VoIP communications**
  - **advanced stateful analysis**
  - **statistical pattern filter**
  - **signaling/media correlation**

Empowered by Innovation

**NEC**