Fault tolerant stateful firewalling with GNU/Linux

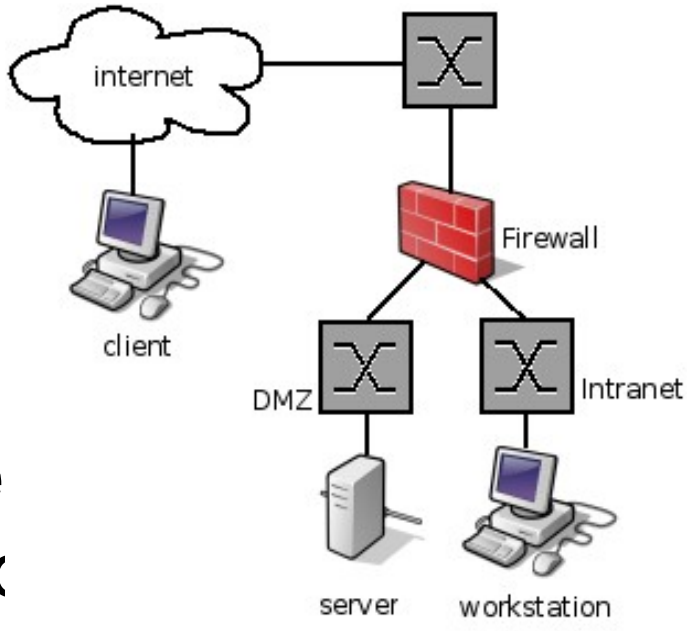
Pablo Neira Ayuso
<pablo@netfilter.org> Proyecto Netfilter
<pneira@us.es> University of Sevilla

Outline

- Introduction:
  - Stateless and stateful firewalls
  - Fault-Tolerance & firewalls
- Architecture & replication protocols
- Preliminary evaluation
- Related works.
- Conclusions & Future Work

Introduction: Firewalls

- **What is a Firewall**?
  - It is a network element that controls the traversal of packets across different network segments.
  - It is a mechanism to enforce an access control list (ACL). The firewall ACL is a list of linearly ordered filtering rules that define the actions that will be performed on packets that satisfy specific conditions.



Introduction: Firewalls

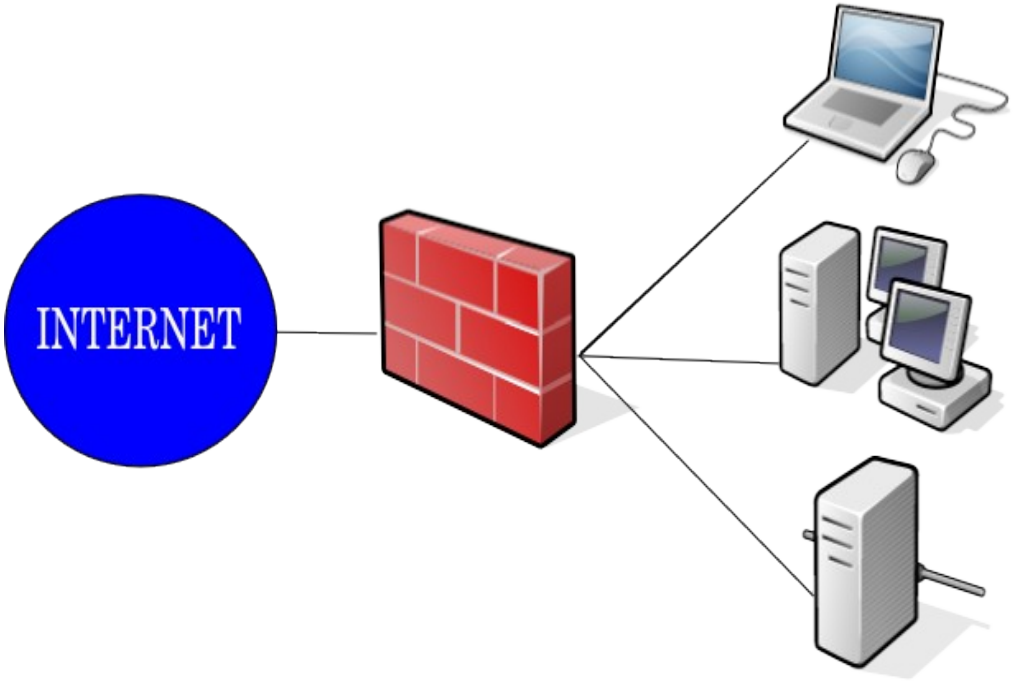
- What is a **stateless** firewall?
  - Filtering decision based on the information available in the packet headers.
- What is a **stateful** firewall?
  - Extend stateless firewalls. A stateful firewall performs correctness checking upon the protocols that it gateways.
  - Filtering decision based on flows.

Introduction: Firewalls

- In Linux, there are four types of states (from the user perspective!):
  - **NEW**: This is the first packet of a connection.
  - **ESTABLISHED**: We have seen packets in both directions.
  - **INVALID**: Malformed, invalid packet in the expected protocol sequence.
  - **RELATED**: This packet is related to an existing connection (ICMP traffic in response to some event).
- http://people.netfilter.org/pablo/docs/

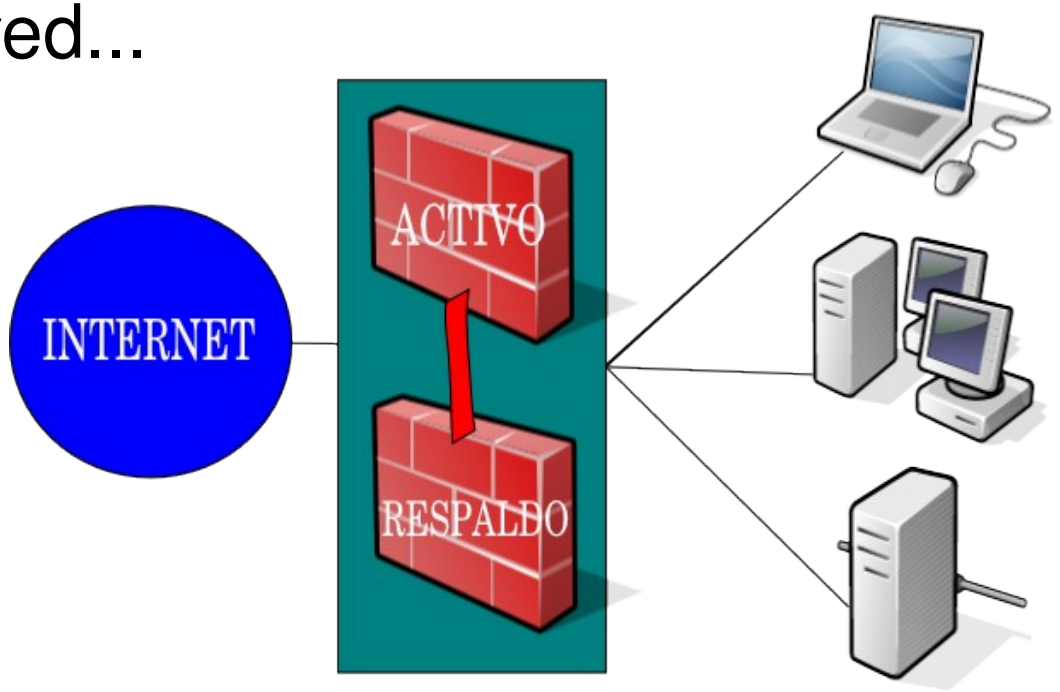
Fault-Tolerant firewalls

- Firewalls introduce a single point of failure in the network schema: The availability of the services depends on the firewall availability.



Fault-Tolerant Firewalls

- Solution: replication
  - Extra cost to add an extra firewall replica.
  - Use of failure-detector to fail over a save firewall replica if a failure arises. e.g. heartbeat, keepalived...



Example scenario

iptables -P FORWARD DROP

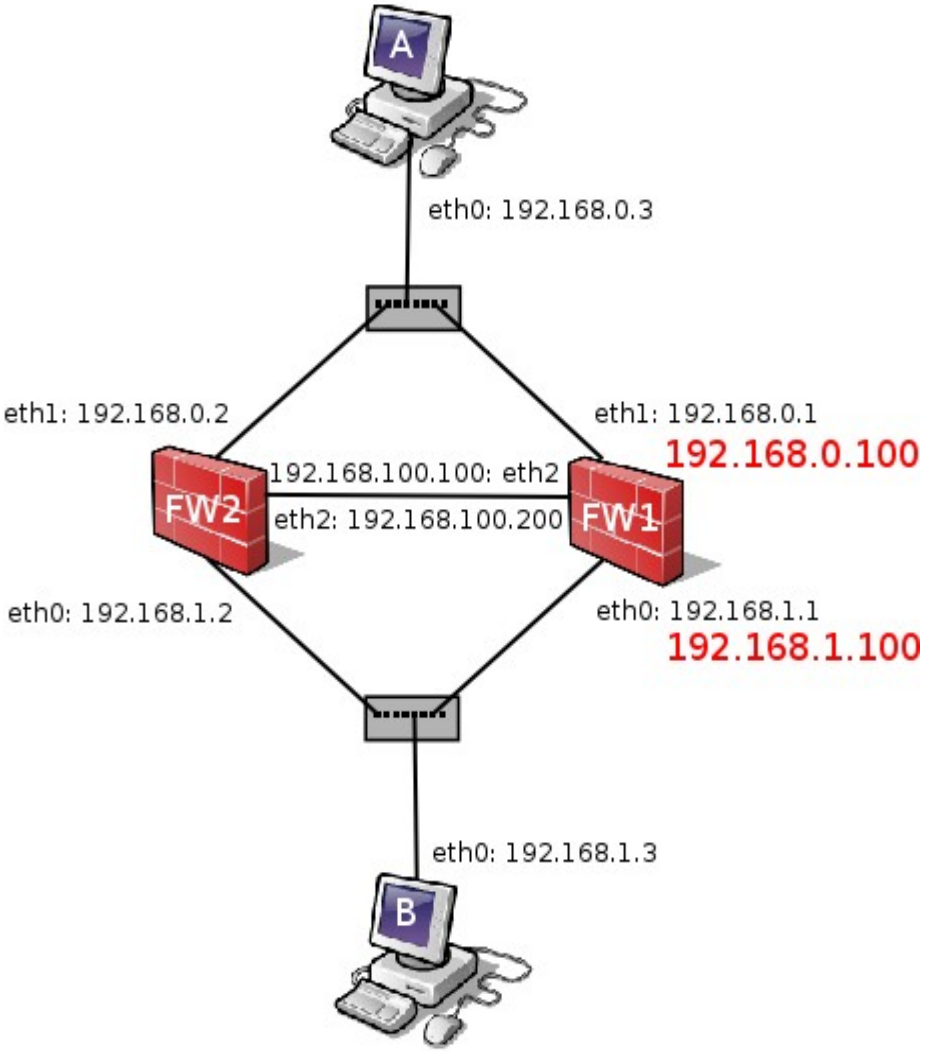
iptables -A FORWARD -i eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT

iptables -A FORWARD -i eth1 -p tcp --syn -m state --state NEW -j ACCEPT

iptables -A FORWARD -i eth1 -p tcp -m state --state ESTABLISHED -j ACCEPT

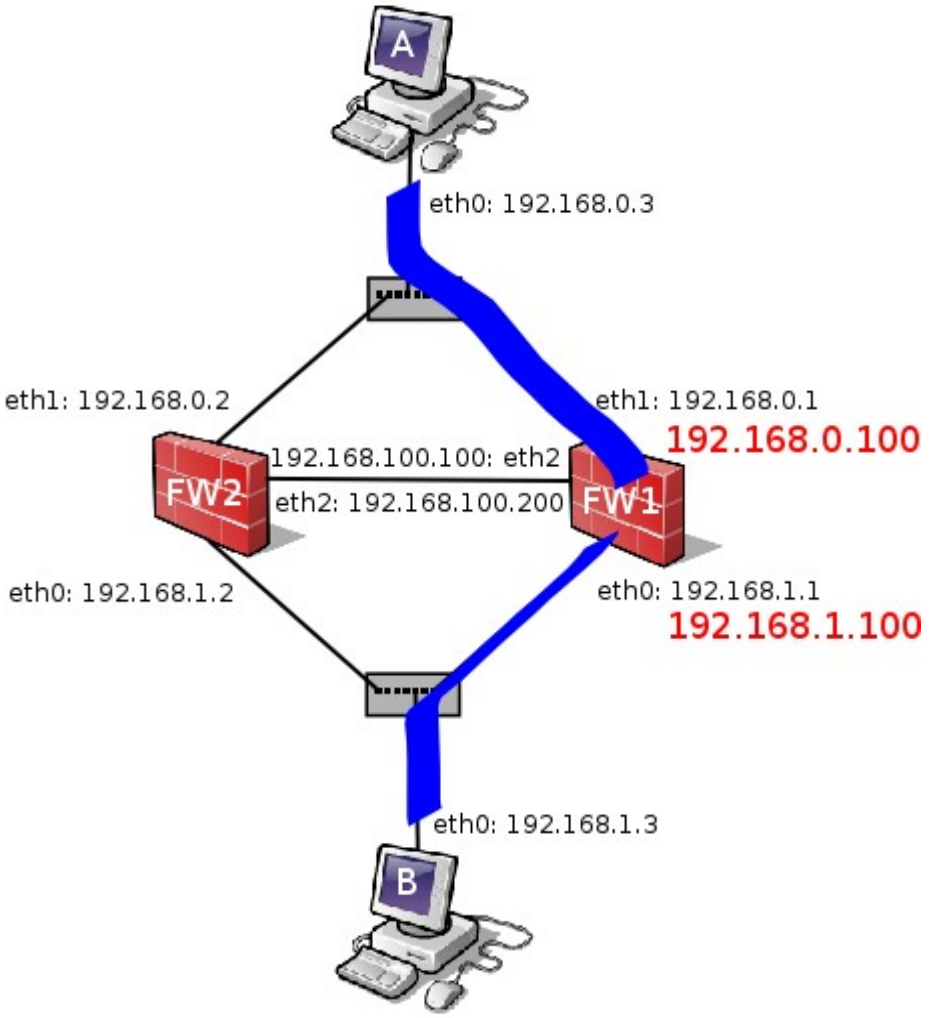
iptables -I FORWARD -j LOG

iptables -I POSTROUTING -t nat -s 192.168.0.3 -j SNAT --to 192.168.1.100



Example scenario

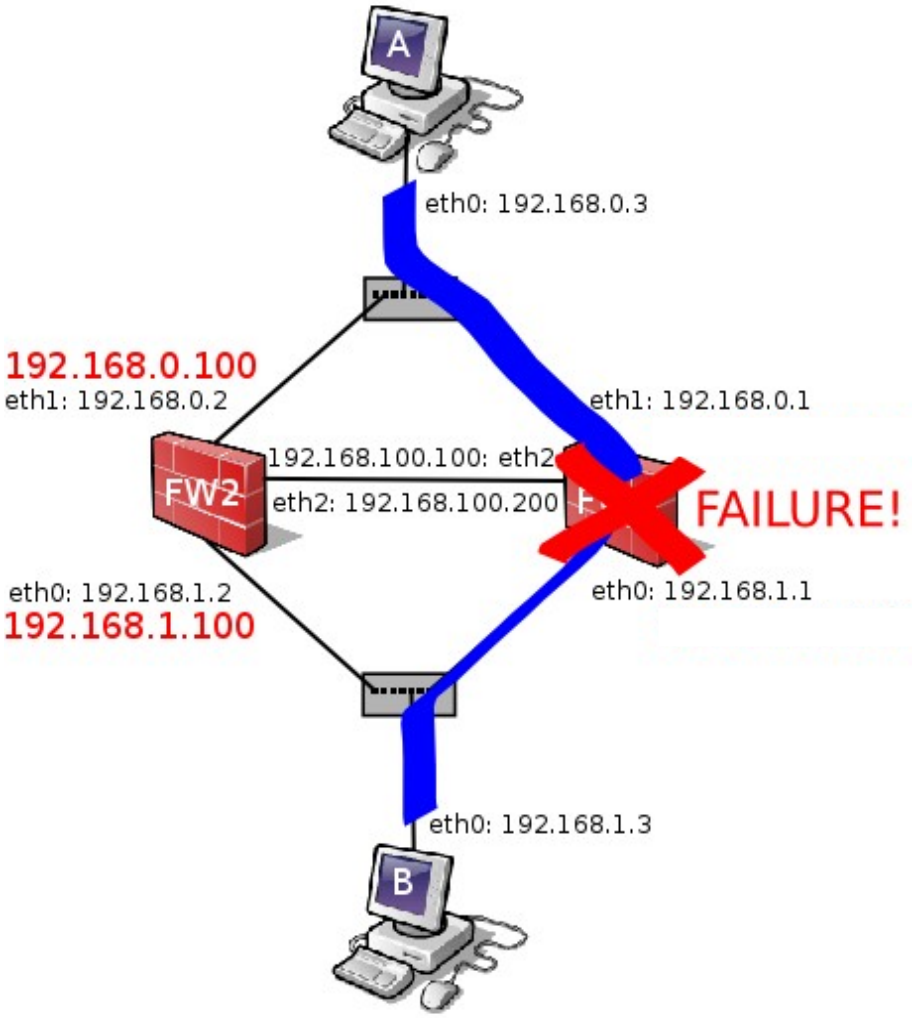
Step 1) Workstation A initiates a SSH connection with B: ESTABLISHED



Example scenario

Step 1) Workstation A initiates a SSH connection with B: ESTABLISHED

Step 2) A failure arises at the primary firewall FW1: the backup FW2 recovers the filtering.



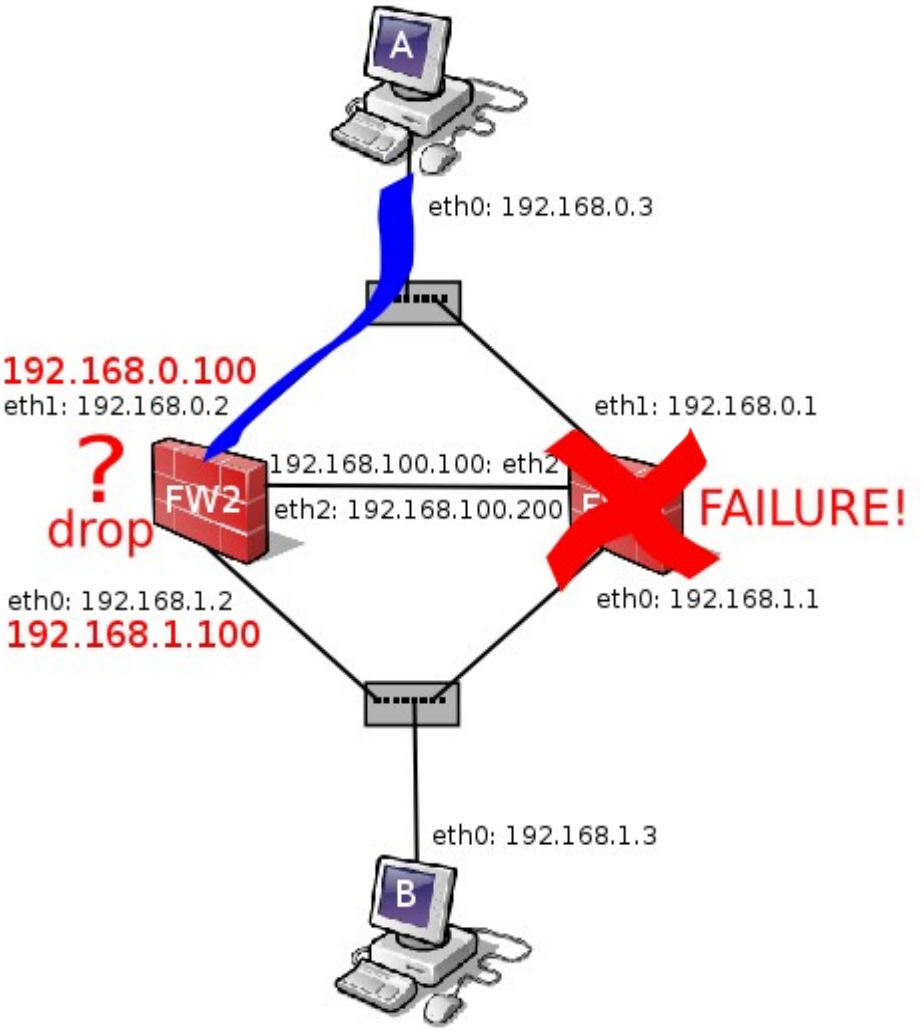
Example scenario

Step 1) Workstation A initiates a SSH connection with B: ESTABLISHED

Step 2) A failure arises at the primary firewall FW1: the backup FW2 recovers the filtering.

Step 3) Applying the stateful fitering policy, the first packet seen by FW2 is in state NEW. Connection hangs.

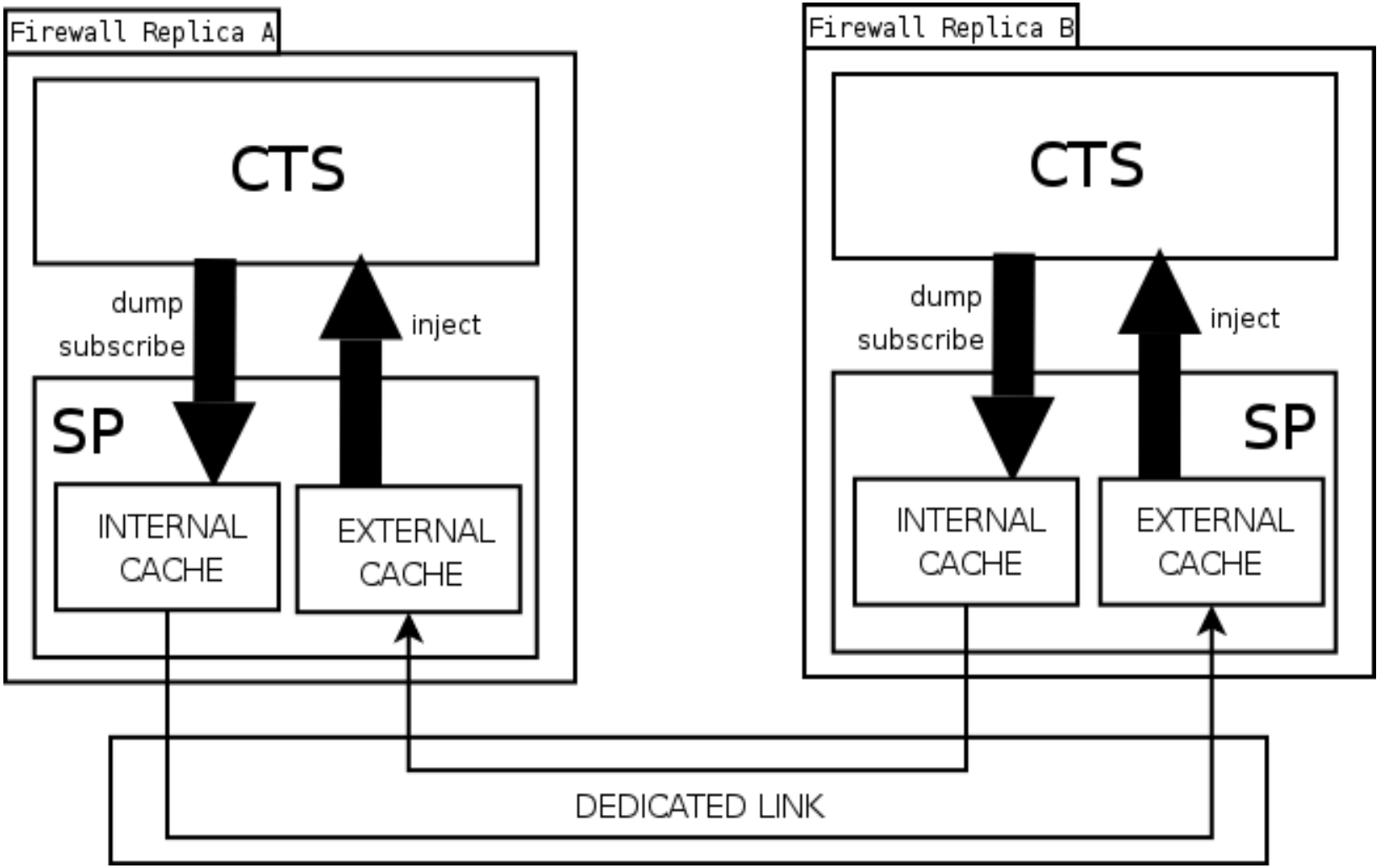
Solution? The backup node needs to know the flow states. Otherwise, it may fail to recover the flows.



Conntrack-tools

- This package contains an userspace daemon written in C – **conntrackd**.
  - Configurable and extensible.
  - Requires Linux kernel >= 2.6.18.
  - Requires HA manager: keepalived, heartbeat...
  - Supported setups:
    - Primary-Backup
    - Multiprimary
  - Under development: current is 0.9.7
  - http://conntrack-tools.netfilter.org/

Conntrackd: Design



Replication protocols

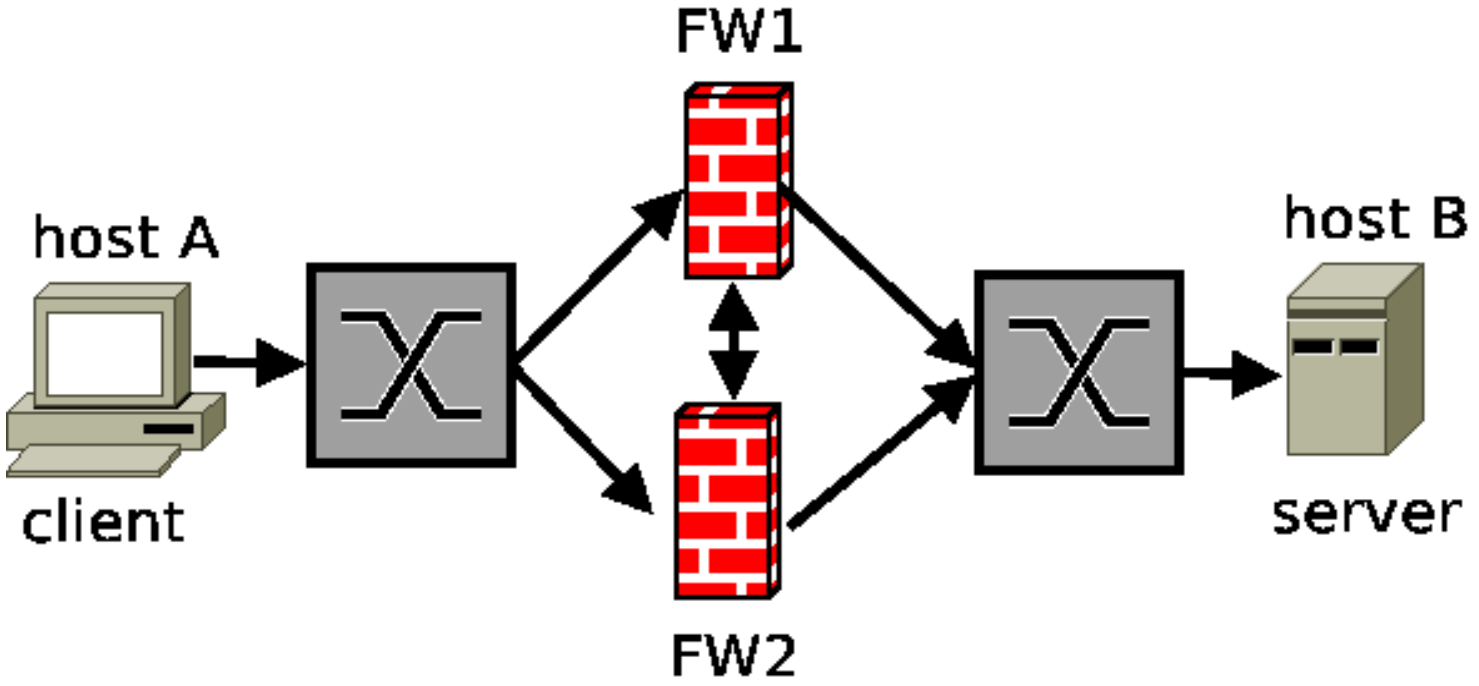
- Three approaches – as for now:
  - **NOTRACK**: like pfsync, best effort, no sequence tracking at all.
  - **ALARM**: Every N seconds a state message is sent (spamming but better for consistency).
  - **FT-FW:** Reliable UDP-based protocol (with sequence tracking).
    - The receiver always handles all messages even those that are out of sequence
    - The sender just resend the last state reached in case of message omission

Supported scenarios

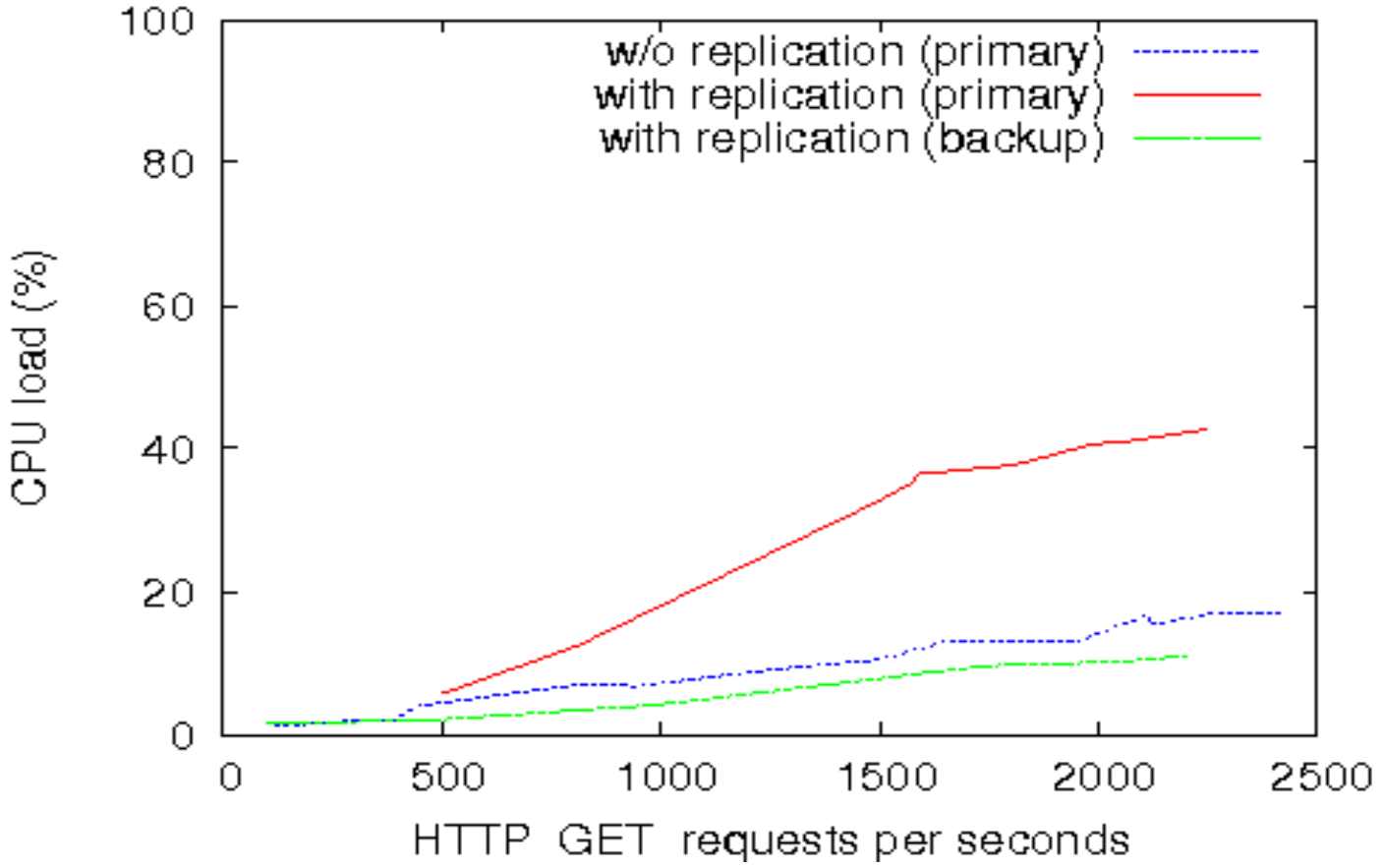
- Primary-Backup: One firewall replica filters and the other acts as backup.
- Multiprimary:
  - Simmetric or flow-based: the firewall replica always handles the same subset of flows.
    - Static: client-based (set different gateways for clients).
    - Dynamic: hash-based (similar to CLUSTERIP)
  - Asymmetric or packet-based: No guarantees on which firewall handles packets (problematic!)

Preliminary evaluation

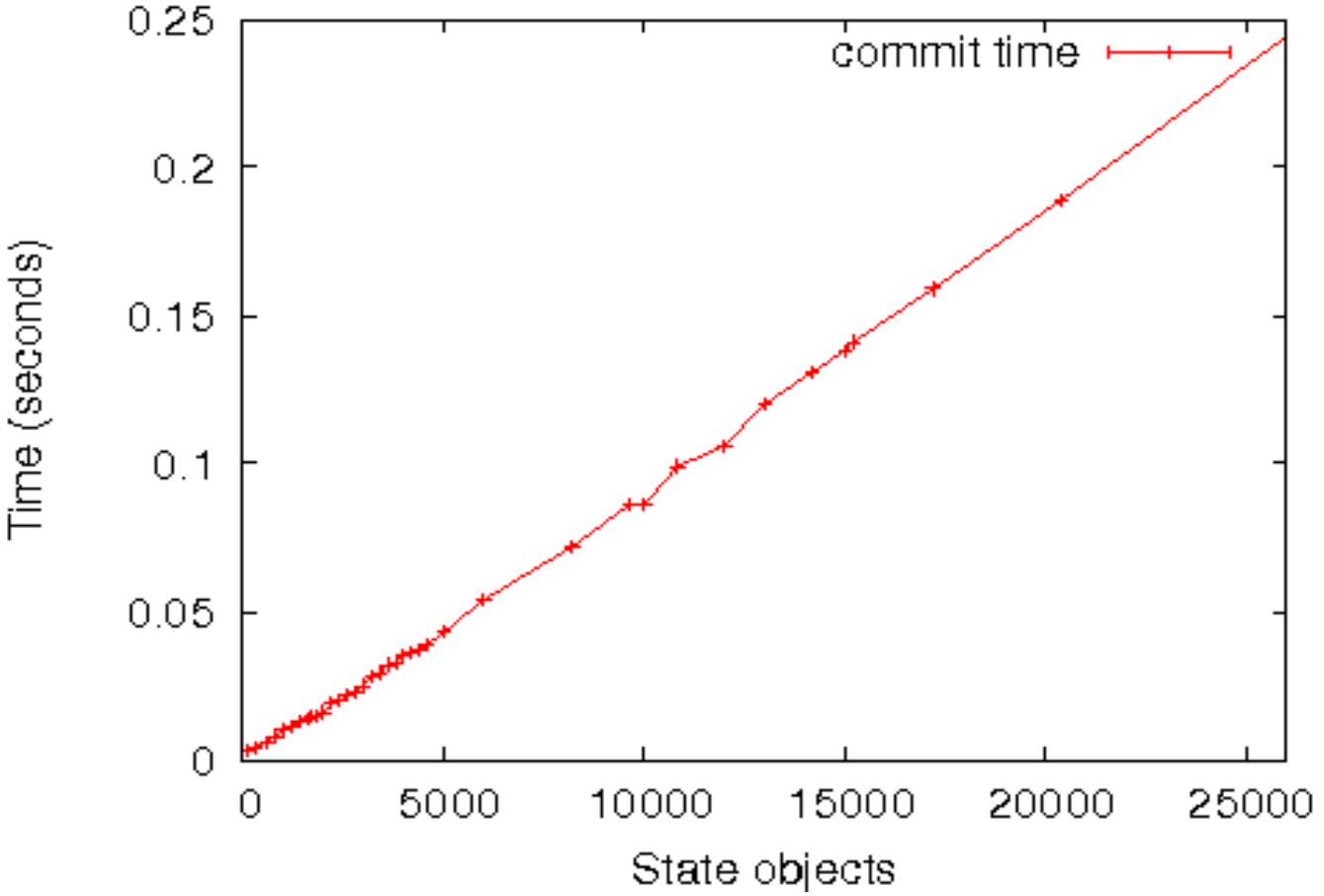
- Testbed: HP Proliant 145G2, AMD 2.2GHz, 1GBit
- HTTP GET requests from A to B (4 KB)



CPU consumption



Recovery time



Related Work

- Apart of the interesting openBSD effort, nothing **open** in this area:
  - **Proprietary**: black box, only commercial papers (buy my nifty box, my solution is great!).
  - **OpenBSD**
    - simple in-kernel state replication
    - it also requires a HA manager, CARP
    - Replication protocol: no sequence tracking at all, no message omission, reordering, duplication handling.

Conclusions & Future work

- The conntrack-tools are free software (GPLv2)
- Userspace implementation.
- Still under development: 1.0 release probably by end of 2008. The remaining issues discussed during the last Netfilter Workshop (sept. 2007) are done.
- It still lacks of documentation for the multiprimary setup.

Thank you!

- Questions ?