LIP Laboratory
ENS-Lyon - France

Towards a Dependable Architecture for Highly Available Services

Narjess Ayari*, Pablo Neira Ayuso**, Laurent Lefèvre***, Denis Barbaron*, Rafael M. Gasca**

* France FT R&D – LANNION, ** Universidad de SEVILLA, ***France, INRIA / LIP, France.

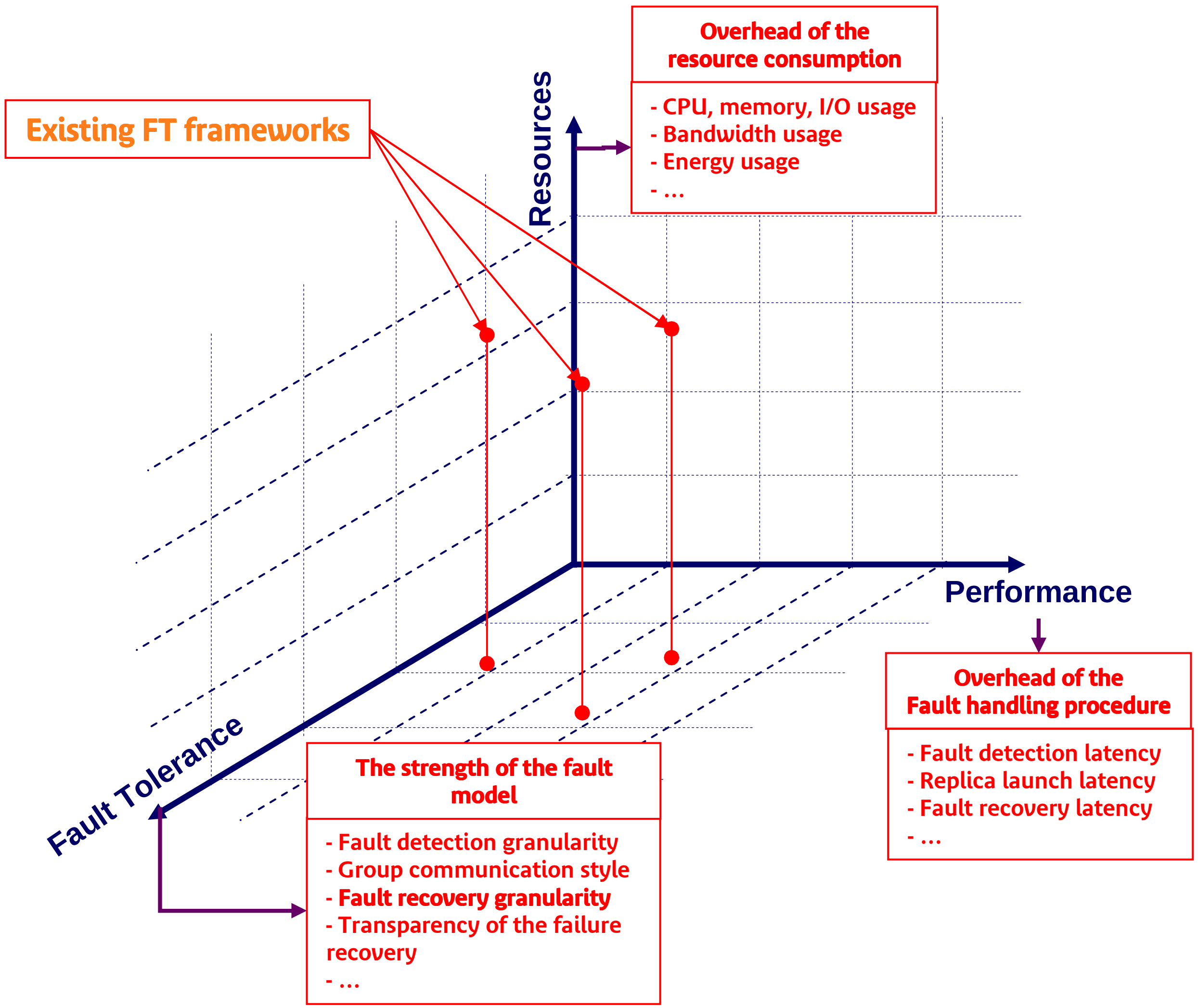
IEEE International Conference on Availability, Reliability and Security - ARES 2008 – Barcelona, Spain, March, 4td – 7th 2008.

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Agenda

- Current Fault Tolerance framework's deficiencies & constraints
- An end-to-end highly available framework
- The FT-FW architecture
- The Active Replication Architecture
- Conclusion & Future work

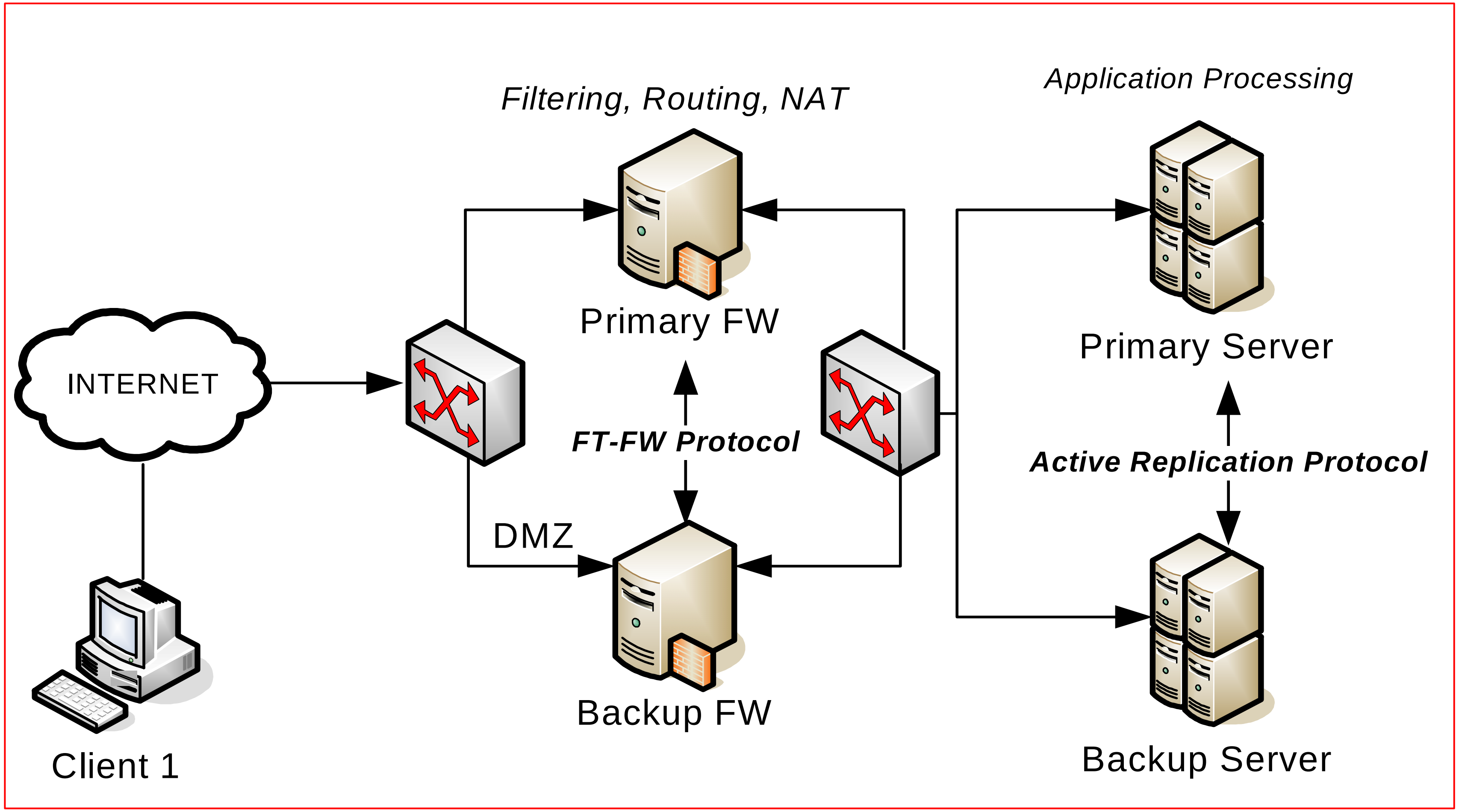
Current FT Frameworks Deficiencies & Constraints



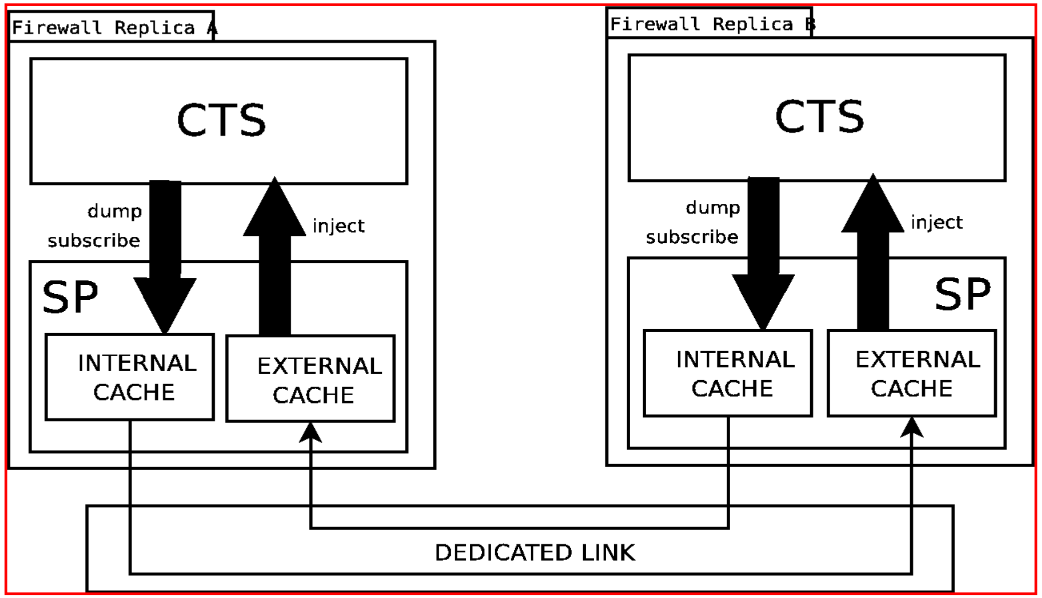
Firewalls and Stateful Devices Requirements

- **Kernel level and application level states require to be properly replicated and correctly spawned in case of failure of a legitimate device**
  - Reactivity to failure
  - Transparency
  - Simplicity
  - Low Cost
  - Efficiency

An end-to-end Highly Available Architecture



FT-FW Architecture



- **Event driven architecture**
  - CTS: Connection Tracking System tracks connections and store states, we extended it with a **framework** to inject states and to receive state change events.
  - SP: State proxy, replication daemon which interact with CTS. It stores two caches.

FT-FW Replication Protocol

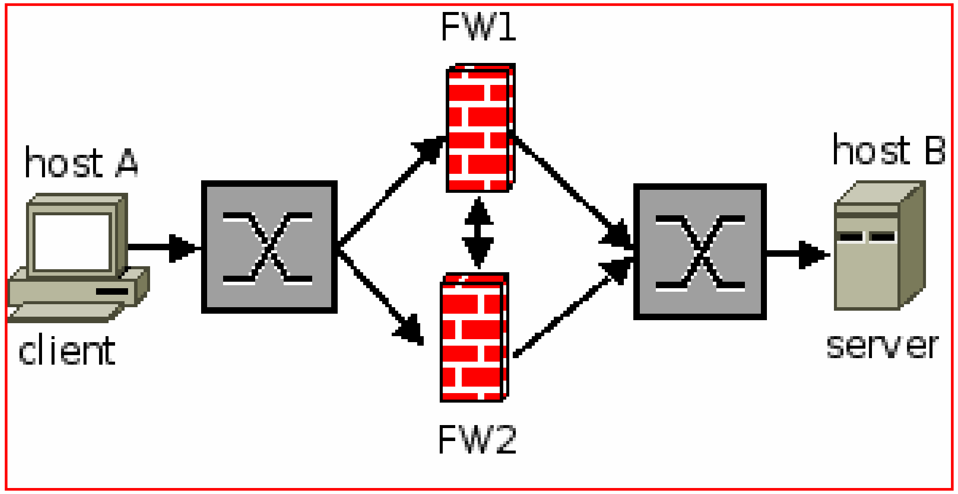
- **The SP is composed of two parts, the sender and the receiver, the main ideas of the replication protocol are:**
  - The **sender** never stops sending messages
  - The **receiver** handles all messages (even those that are out of sequence).
- **The protocol reduces the number of retransmitted messages in case of message omission.**

LIP Laboratory
ENS-Lyon - France

france telecom

Experiments

Evaluating FT-FW: The Design Space

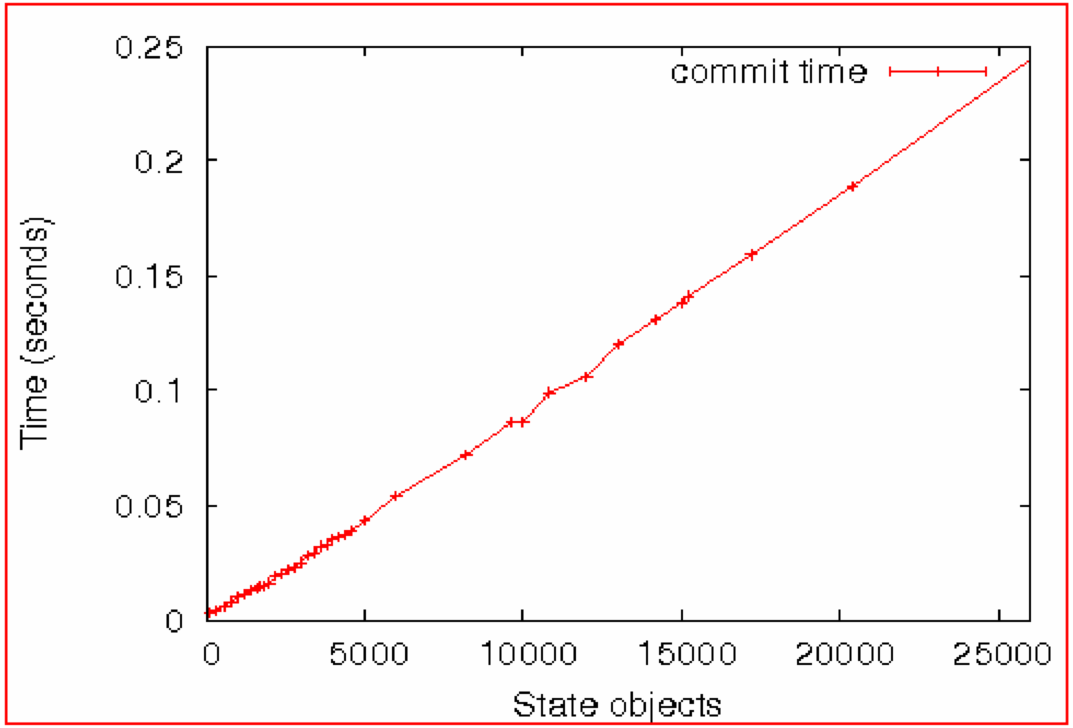


➲ **Machine specs:**

- **HP Proliant 145g2, AMD Opteron2.2GHz, 1 GEthernet.**
- **conntrack-tools: free software (GPL) user-space daemon which implements the SP.**

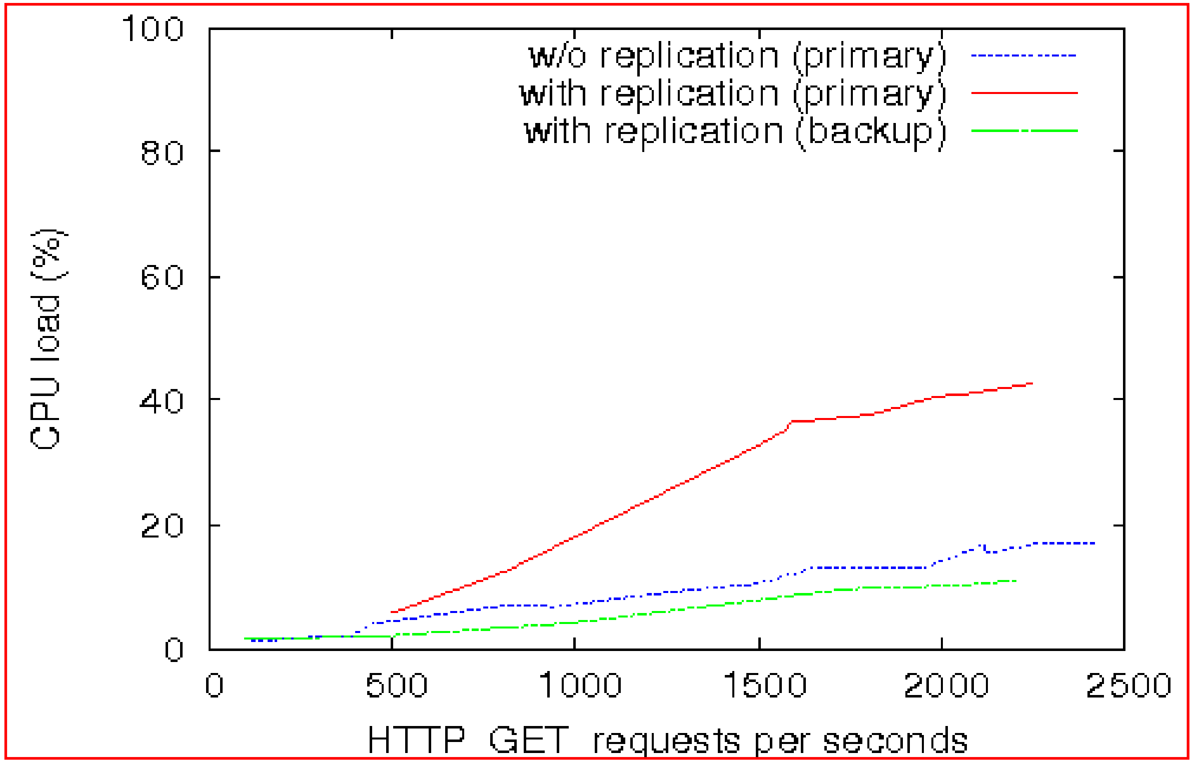
Evaluating FT-FW: Commit Time

- Commit time: Time required to inject the states stored in the external cache into the CTS. 25000 states in 250 milliseconds.



Evaluating FT-FW: Replication Cost

- CPU consumption during replication (1 TCP connection means 6 state changes): ~40% for 2500 HTTP GET connections per second.
- – Ping roundtrip: The solution introduces a delay of 5 milliseconds (negligible)

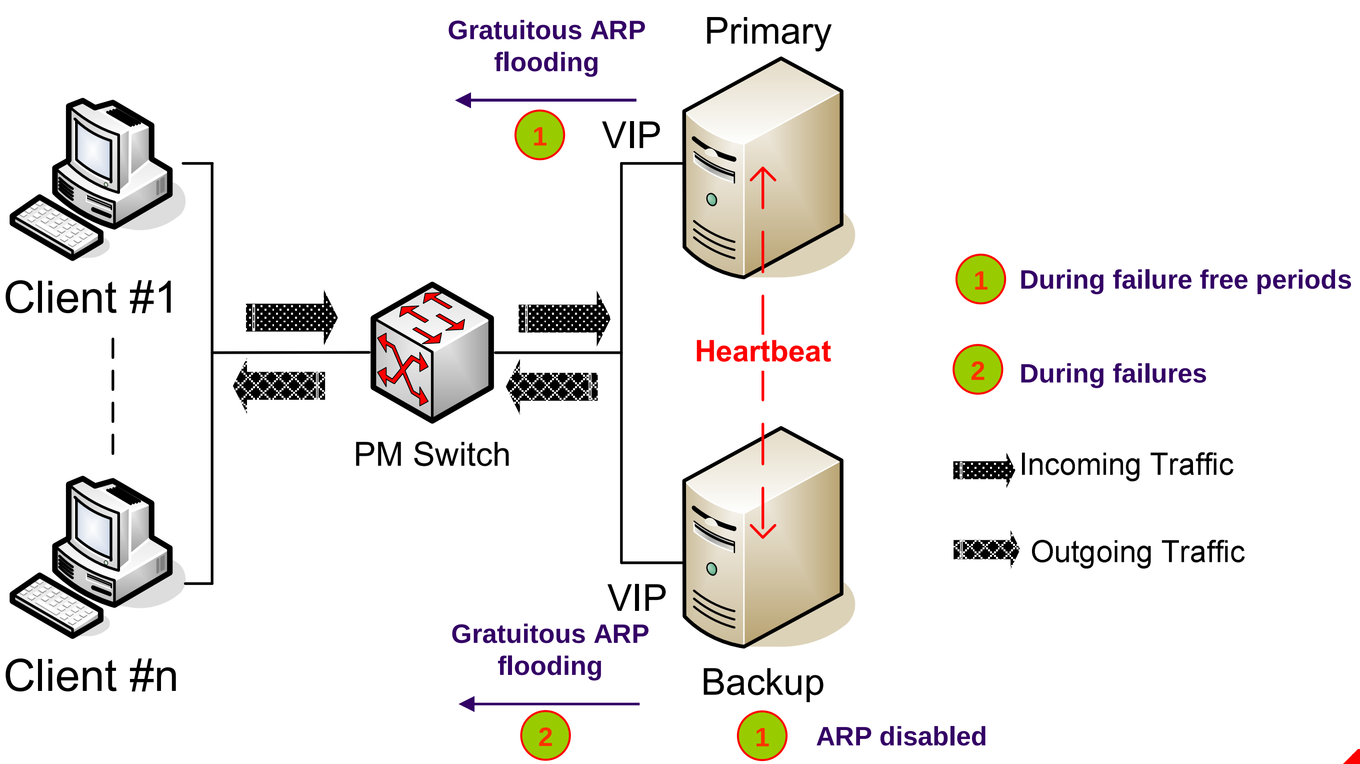


Discussion

- **Because of its asynchronous nature,**
  - Does not guarantee that firewall replicas are one-copy equivalences as in **database** replication schemas:
    - Two-phase and three-phase commit protocols are heavyweight since they would introduce an unaffordable delay.
- **Solution:**
  - Flexible recovery: Recover states from precedent states (reduce security)
  - Initial stages of a flow, the clients usually transparently retry after a short timeout if no response is received.
  - When Checkpointing states doesn't apply
    - Use the active replication concept

The concept of Active Replication

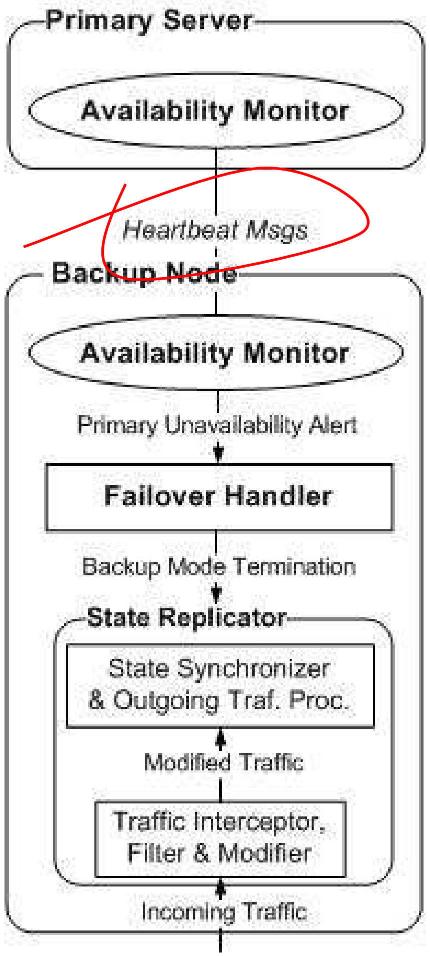
- "Active Replication" (AR) applied to any "stateful" device
  - Enables the failover of a session while avoiding the interruption of the ONGOING ones
  - Provides the transparent replication of the **kernel** states (the Netfilter states, the flow states, etc.) + the **application level** states (the load balancer states, etc.)



D13 - 29/02/2008

AR Architecture Components

- Full client/server transparency / Applies to any stateful device / No overhead is incurred to the end-to-end communications during failure free periods / Good performance during failures



AR Implementation Details (1/2)

- **The active replication based component for state replication**
  - Packet non intrusive interception and processing using {BPF/pcap/libnet} based user space packages
    - ICMP/UDP/TCP support
  - User and kernel space module for TCP state replication
    - Netlink capabilities
    - Structure of a ct_sync msg

Timestamp	
Source IP @	
Destination IP @	
Source Port	Destination Port
Sequence Number	

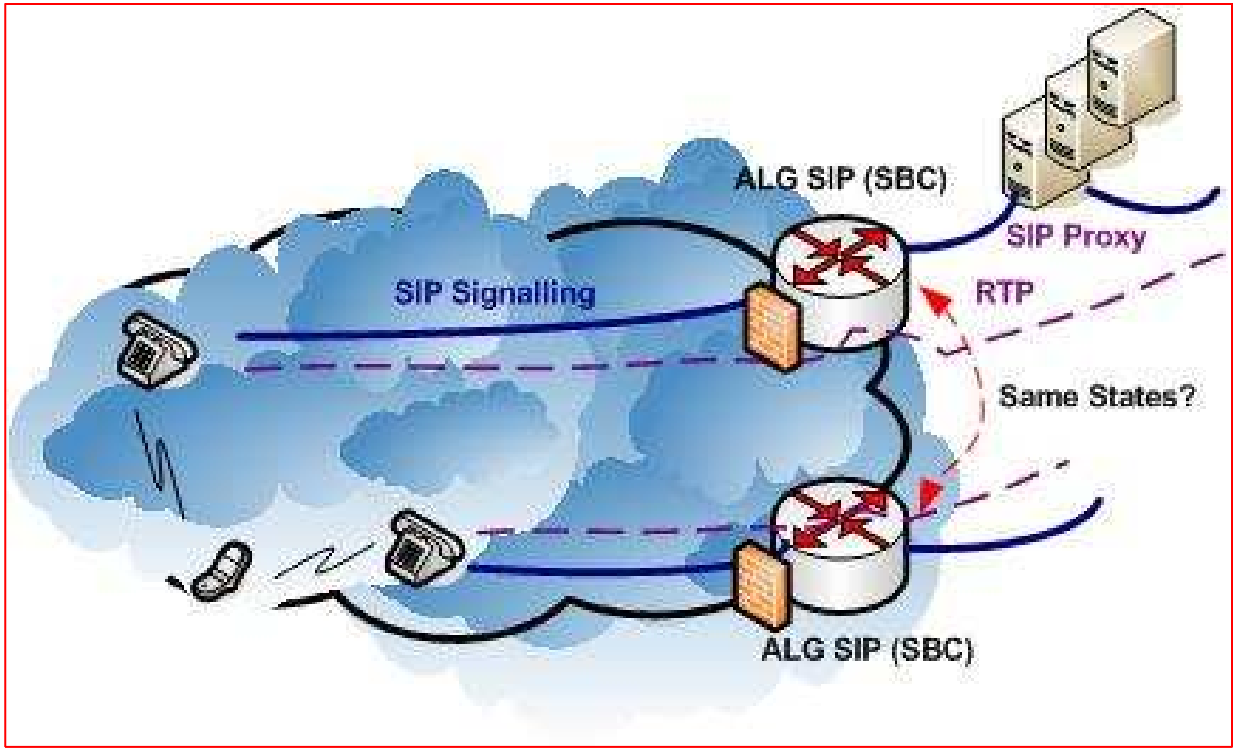
  - State consistency
    - Netfilter based rules triggered at the backup startup

AR Implementation Details (2/2)

- **The failure detection component**
  - Concept & Properties
    - Detects failures as soon as they occur
    - One error-free instance of the service is running at once
      - *STONIGHT & Mon*
  - Implementation
    - HB based: Light customization to handle time at micro-sec granularity
- **The failure recovery component**
  - Availability through Network level takeover
    - GRATUITOUS ARP resource takeover
  - Reliability through state replication
  - Disabling of the AR process
    - Interception, filtering, ARP module customizations, etc.
  - Ongoing connection freezing

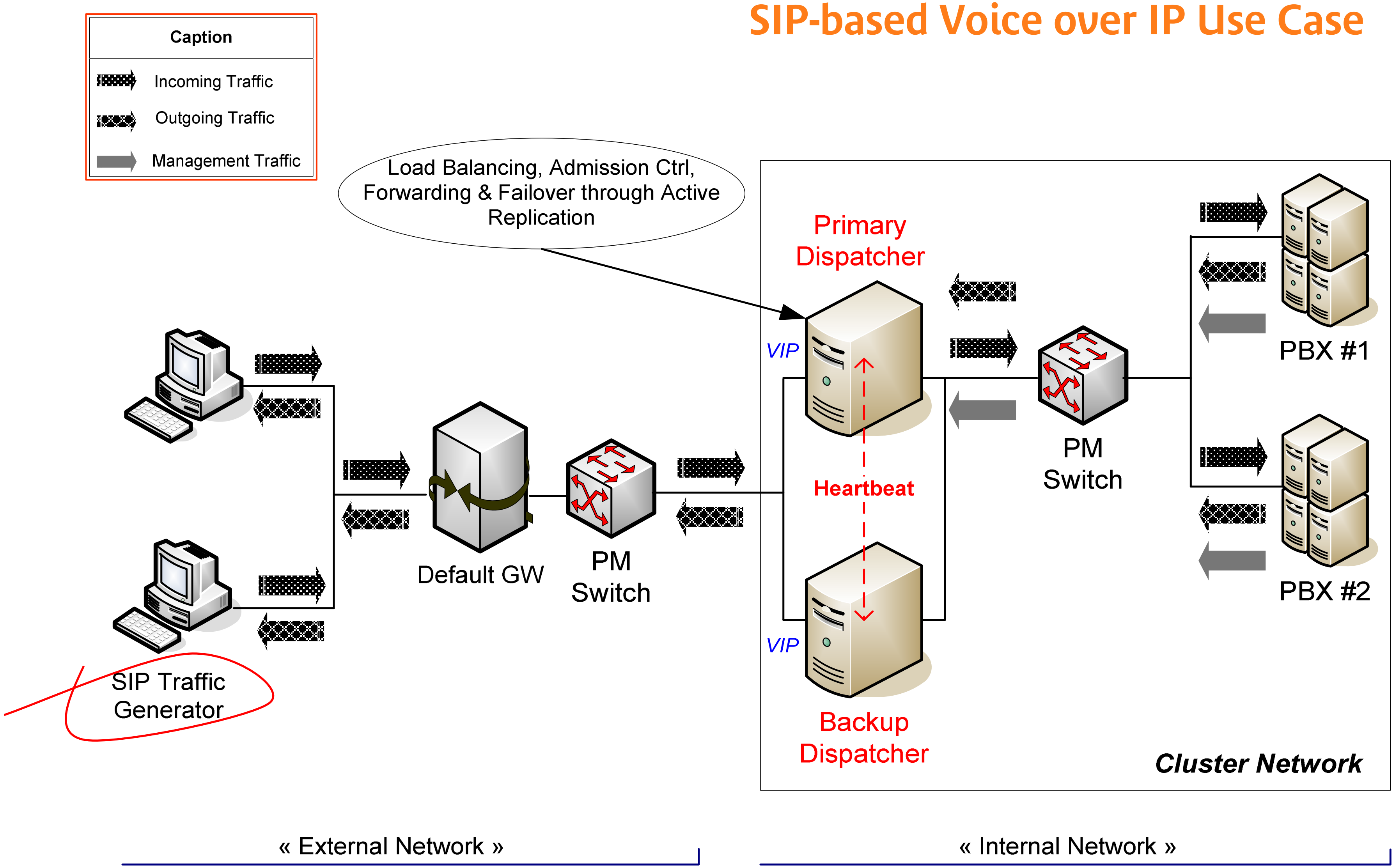
Experiments

The SIP-based VOIP Use case



The Experimental Setting

SIP-based Voice over IP Use Case



Performance Evaluation

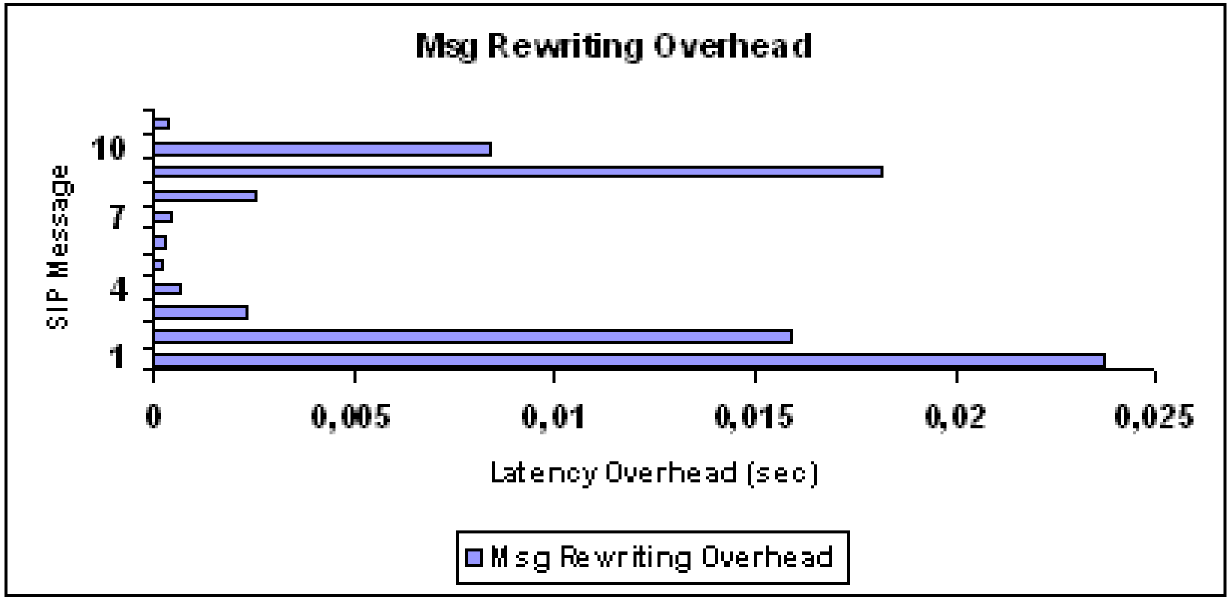
State replication consistency

Generated Traffic	Traffic interception and rewriting at the Backup	Traffic rewriting cost at the Backup	Backup outgoing traffic generation
ICMP	Correct frames (Data + Headers)	Less than the ms	Correct replies (Data + Headers)
UDP	Correct frames (Data + Headers)	Less than the ms	Correct replies (Data + Headers)
TCP	Correct frames (Data + Headers)	Less than the ms	Correct replies (Data + Headers)

Performance Evaluation

Latency incurred to the INCOMING & OUTGOING signaling traffic

- **Definition**
  - The time from when a legitimate frame is intercepted by the backup node to the time at which a copy of the same frame is delivered to the backup's kernel.



Performance Evaluation

Other Failure-Free Periods Performed Tests

- **Failure free periods AR cost**
  - Low CPU & memory usage overhead
  - Service level consistency
    - Only one instance of the cluster entry is talking at a time

Performance Evaluation

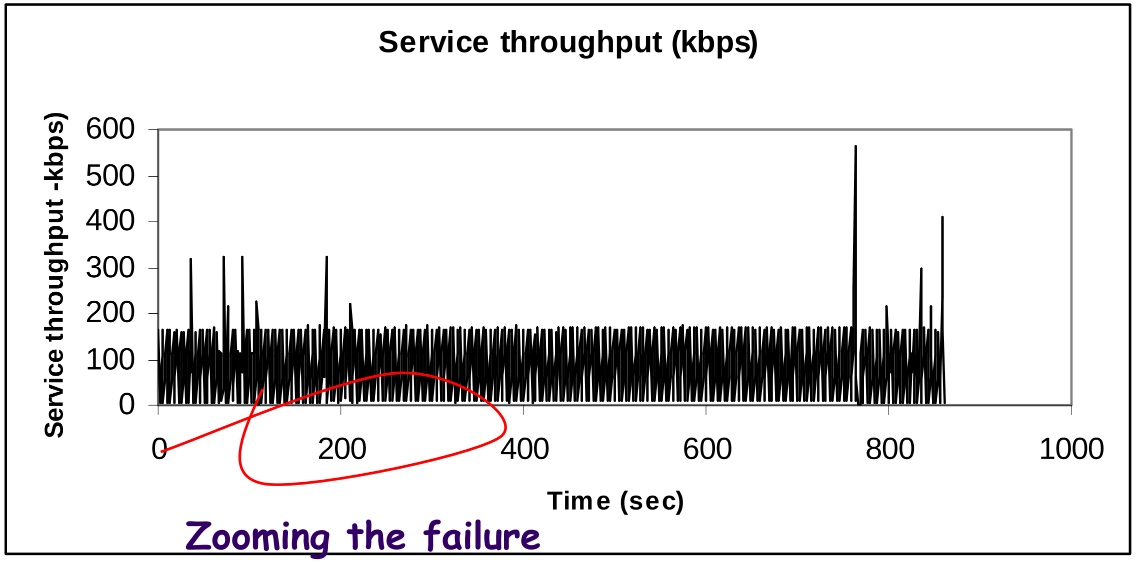
Failure Recovery Cost

- **Failure recovery effectiveness**
- **Failure recovery latency**
  - *Definition*
    - $latency_{failure\_recovery} = latency_{Fault\_Detection} + latency_{Takeover}$

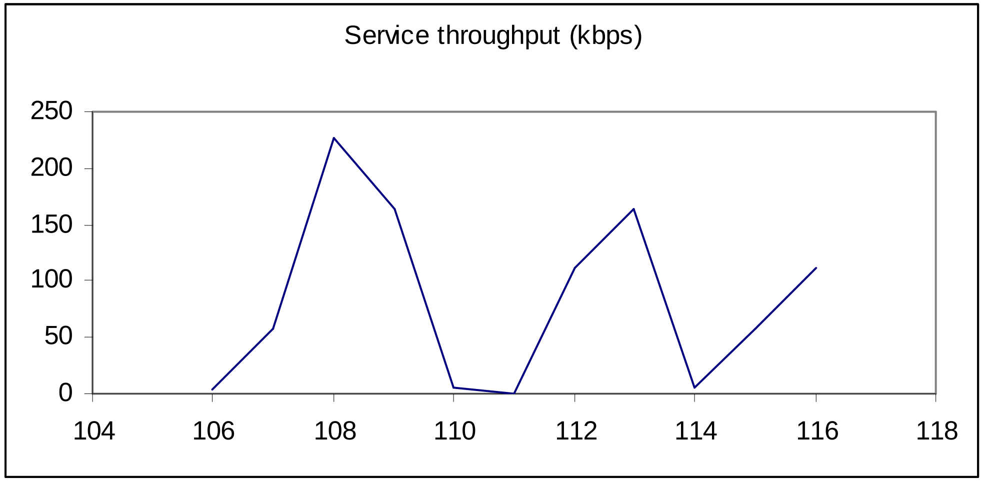
FDI (sec)	Average Failure Detection Latency (micro-sec)	Average Takeover Latency (micro-sec)	Average Recovery Time (micro-sec)
1	0 708 375	0 546 748	1 255 123
3	0 825 868	0 546 748	1 372 616
5	1 430 499	0 546 748	1 977 247

Performance Evaluation

Failure Recovery Cost



Zooming the failure injection time



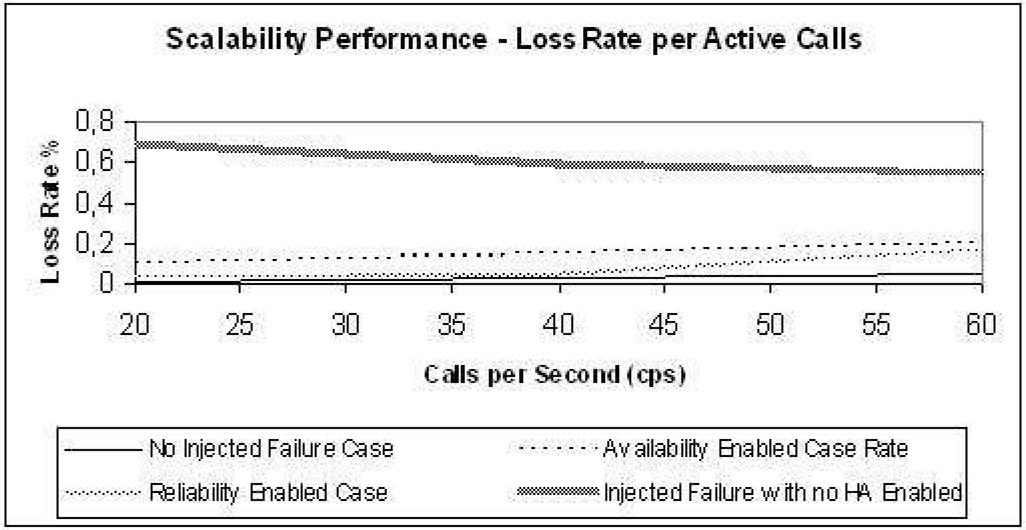
Performance Evaluation

Failure Recovery Cost

Scalability measures

- The loss ratio is intuitively defined as the following (cps):

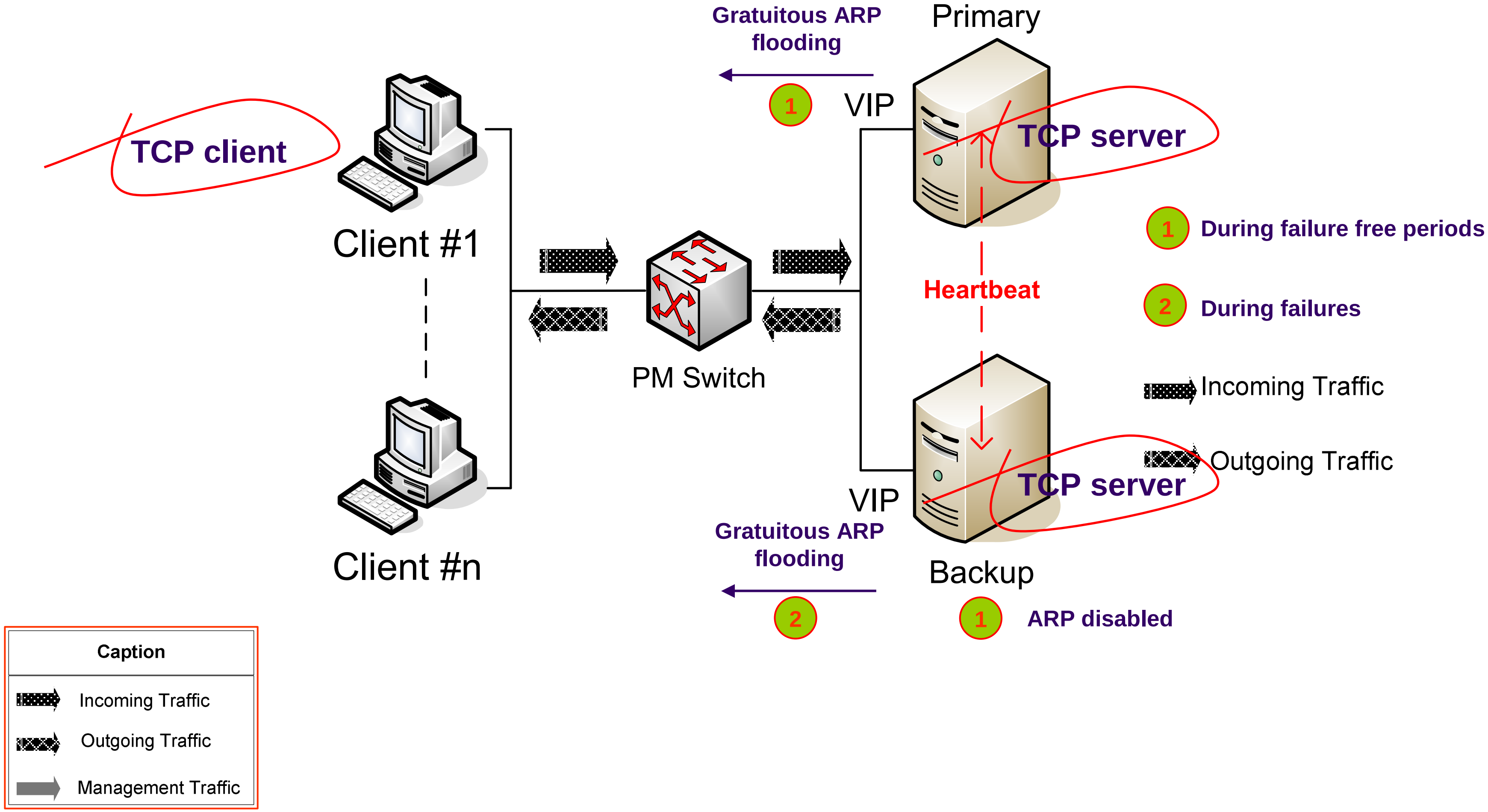
$$Loss\_Ratio = \frac{Average\_Nber\_Failed\_Sessons}{Total\_Nber\_Active\_Sessions}$$



INRIA

The Experimental Setting

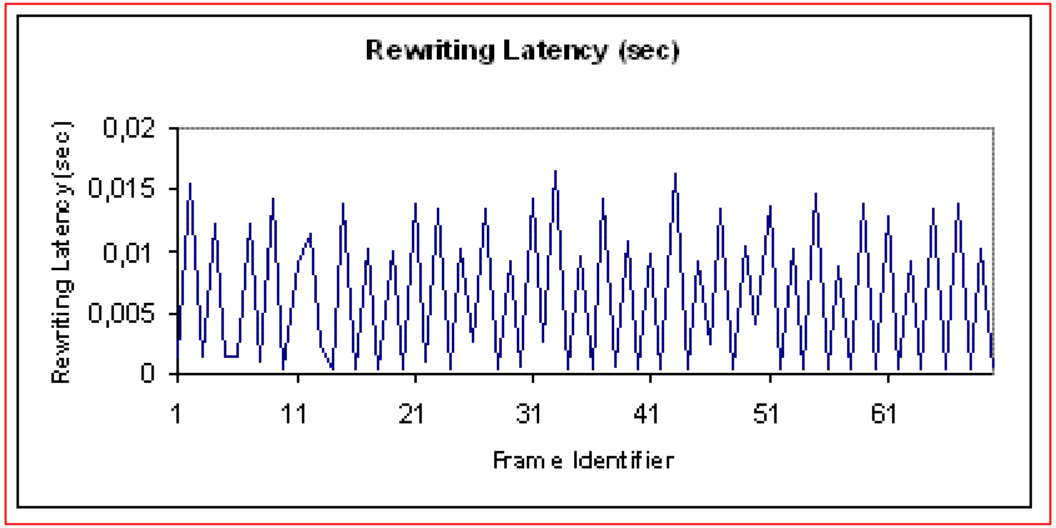
Simple TCP-based Conversation Use Case



Performance Evaluation

Failure Free AR Cost

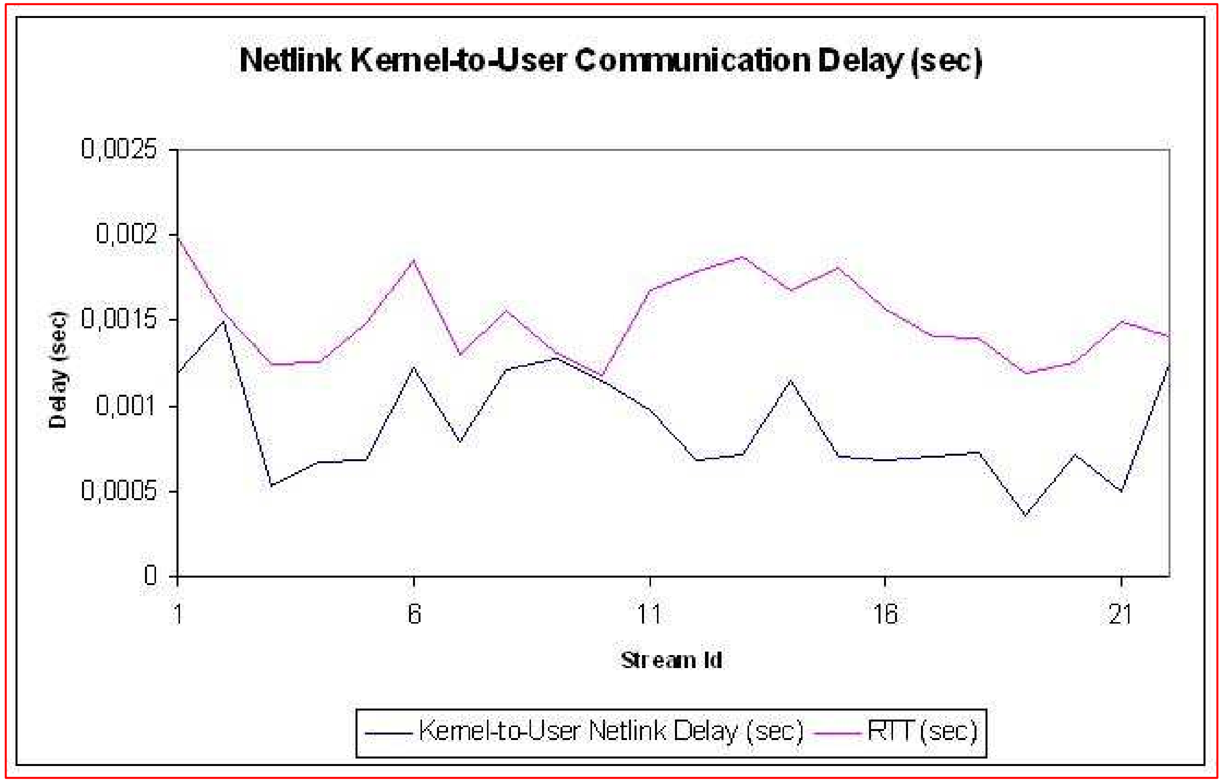
- Pl. refer to slide 12 for the definition



Performance Evaluation

Failure Free AR Cost

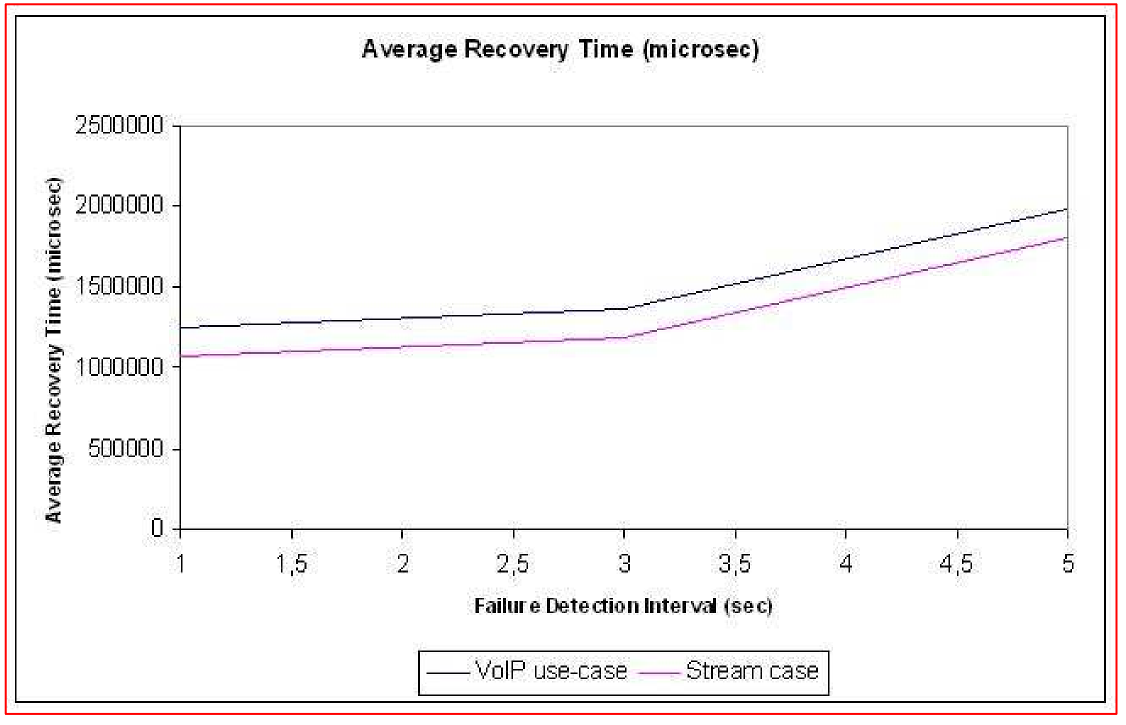
- **Kernel-to-User Netlink communication delay**
  - The time required by the Netlink based kernel module to send back the acknowledgement to the user space application, after successfully updating the state of the fake TCP connection.



Performance Evaluation

Failure Recovery Cost

- Pl. refer to slide 14 for the definition



Conclusion

- **We presented a "true" client/server transparent service high availability through active replication**
  - Recovers the ongoing and the new offered sessions
- **Provides**
  - Replication of kernel level states (transport layer states, etc.) & application level states for stateful devices
- **Assumes for now**
  - "Deterministic enough" applications
  - A "lightweight" Linux kernel 2.6.x dependency
- **Room for improvement**

Possible extensions

- **Active/Active redundancy**
  - VRRP
- **More AR compliant applications**
  - AR + checkpointing (for less deterministic applications)
- **More large scale tests**

Thanks

Any Questions?

Contact: narjess.ayari@orange-ftgroup.com