Balancing Gossip Exchanges in Networks with Firewalls

J. Leitão, R. van Renesse and L. Rodrigues

IPTPS 2010
April 27, 2010

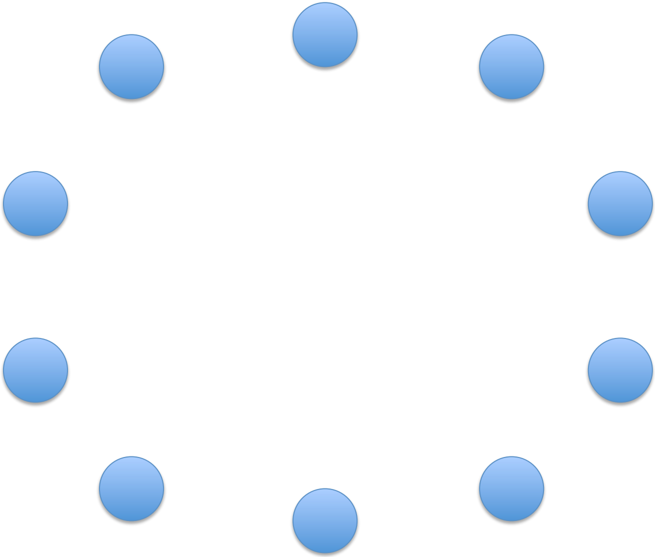
Outline

Introduction

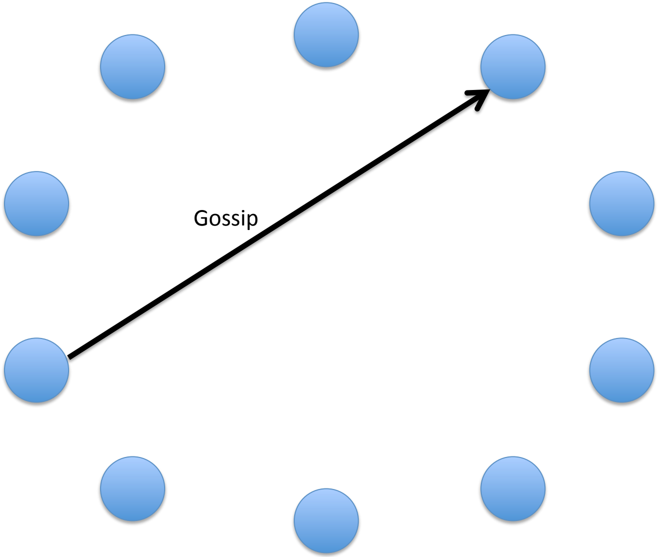
Scope

- Gossip protocols:
    - Very flexible.
    - Easy to implement.
    - Scalable.

Introduction
Gossip protocols

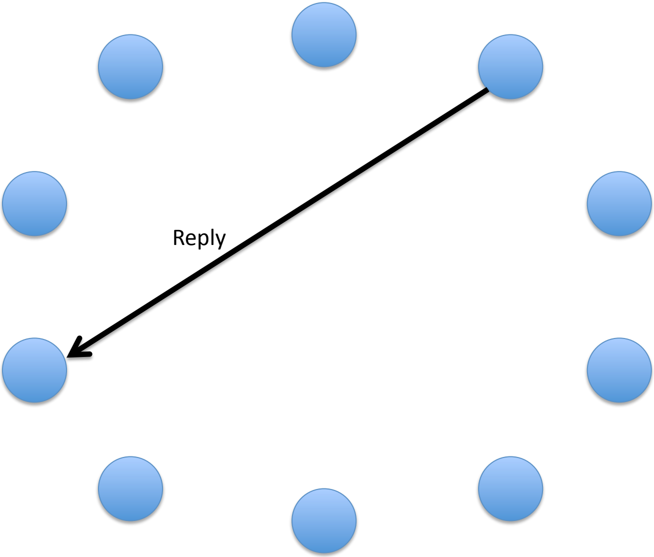


Introduction
Gossip protocols



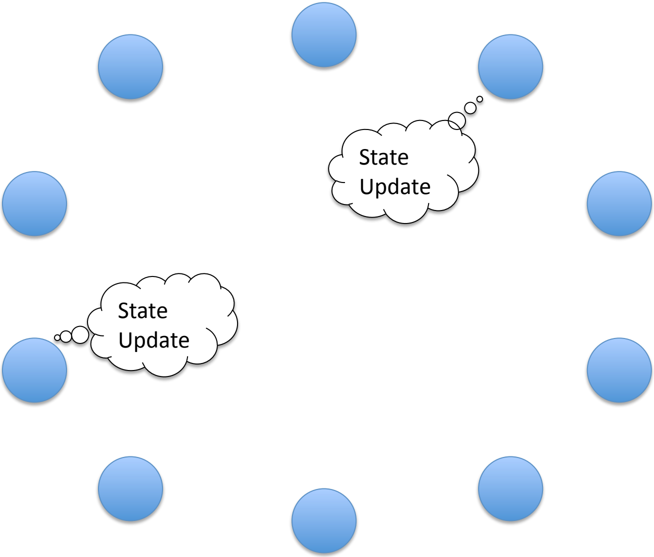
Introduction

Gossip protocols

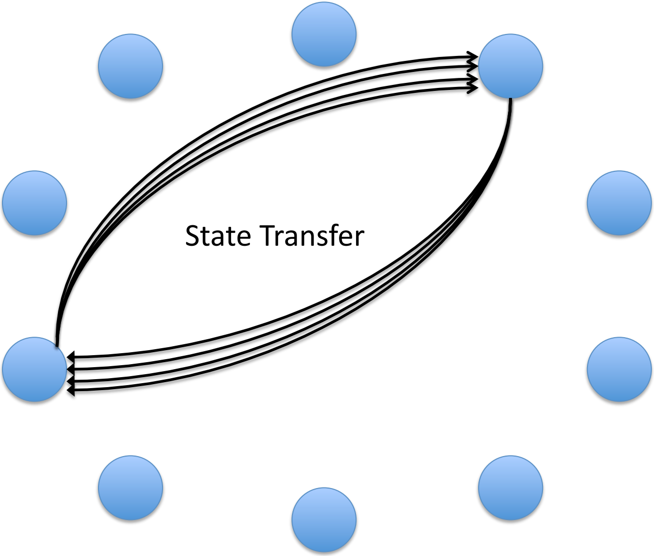


Introduction

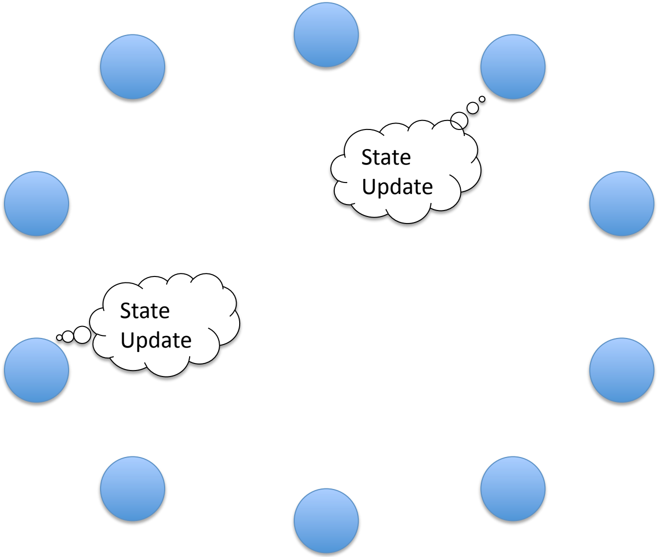
Gossip protocols



Introduction
Gossip protocols



Introduction
Gossip protocols



Introduction

Gossip Protocols.

Inherent load-balancing properties

Every participant will engage in a similar number of gossip exchanges.

Load Balancing...

Only true if considering a **“flat” topology**.

Introduction

Gossip Protocols.

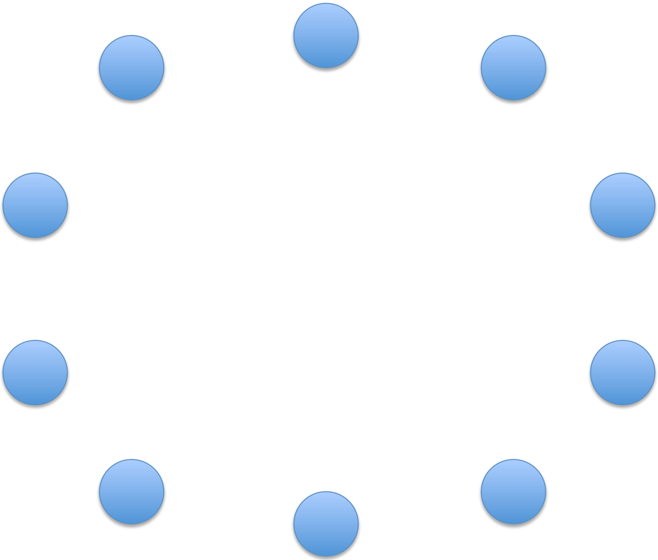
Inherent load-balancing properties

Every participant will engage in a similar number of gossip exchanges.

Load Balancing...

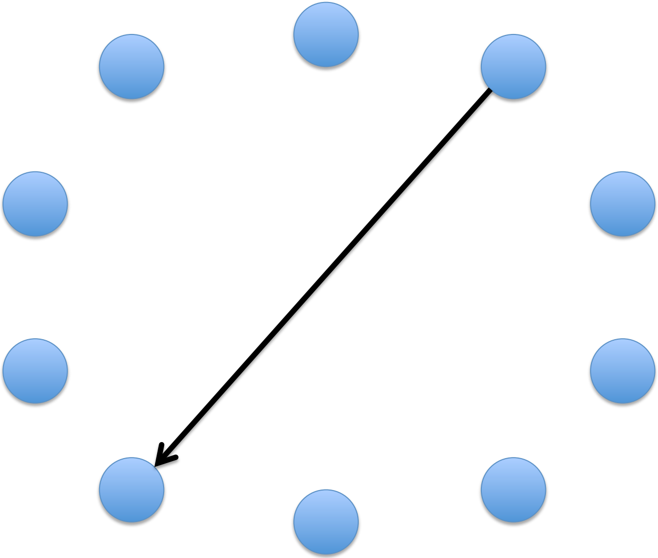
Only true if considering a **“flat” topology**.

Introduction
Introducing Firewalls and NAT boxes



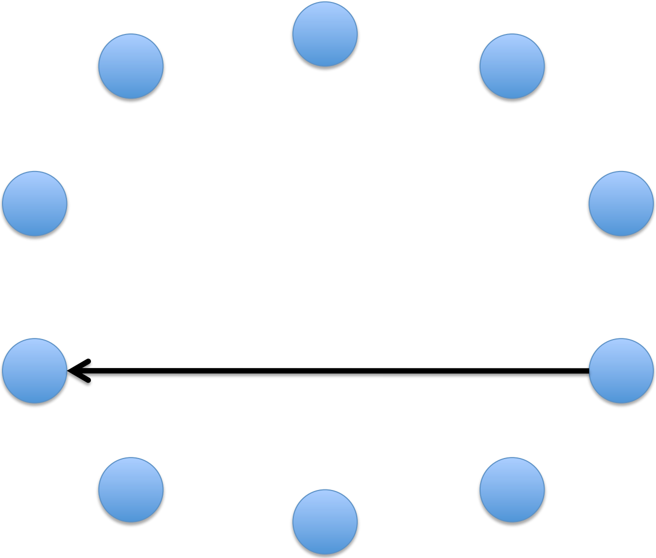
Introduction

Introducing Firewalls and NAT boxes



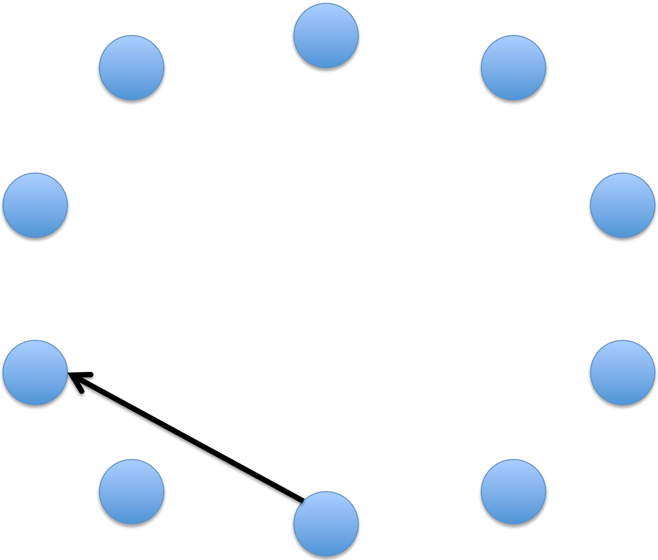
Introduction

Introducing Firewalls and NAT boxes



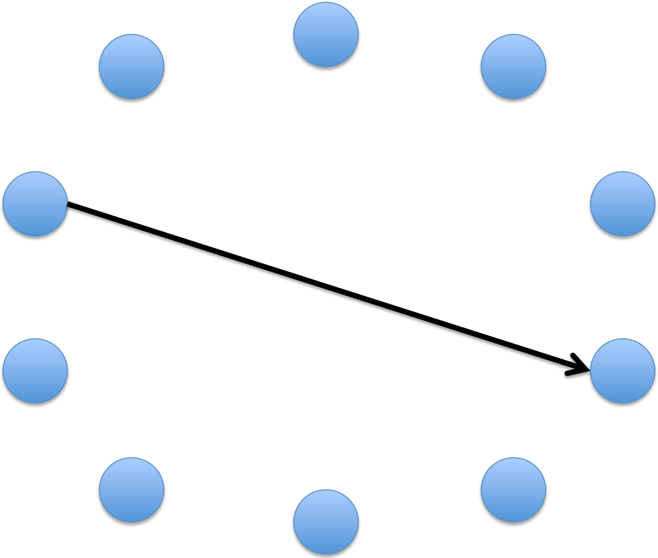
Introduction

Introducing Firewalls and NAT boxes

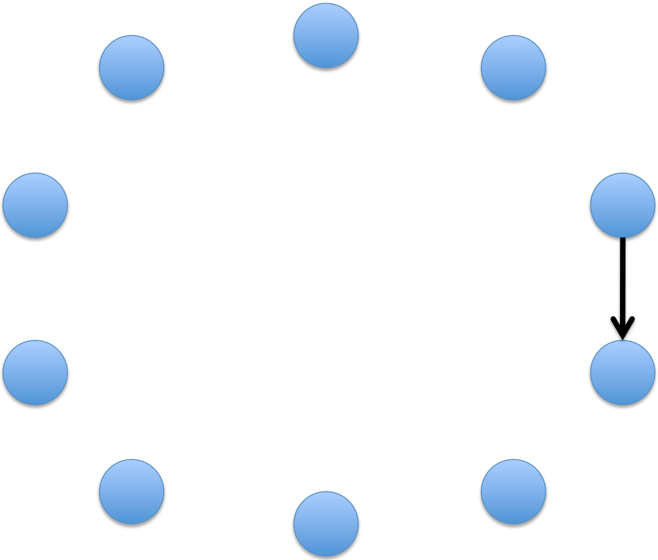


Introduction

Introducing Firewalls and NAT boxes

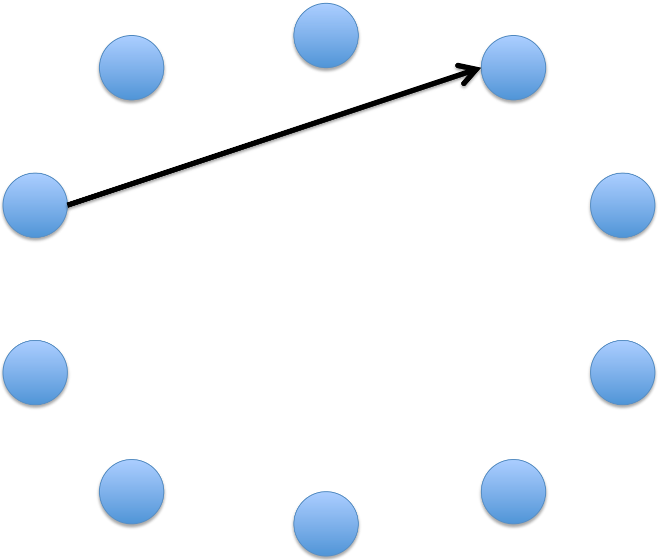


Introduction
Introducing Firewalls and NAT boxes

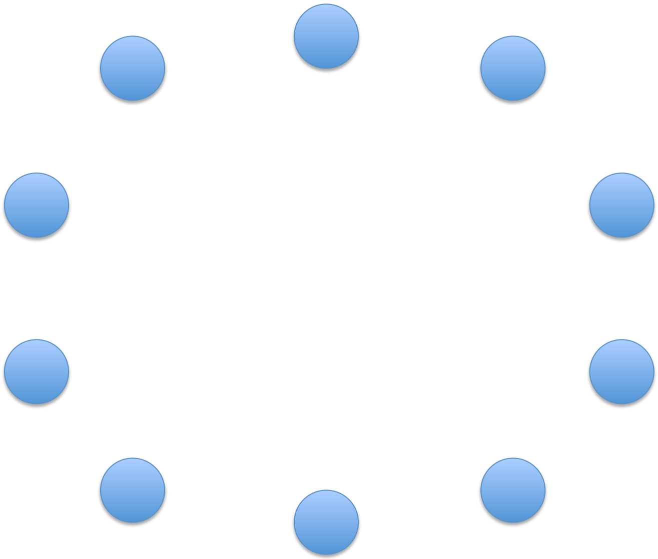


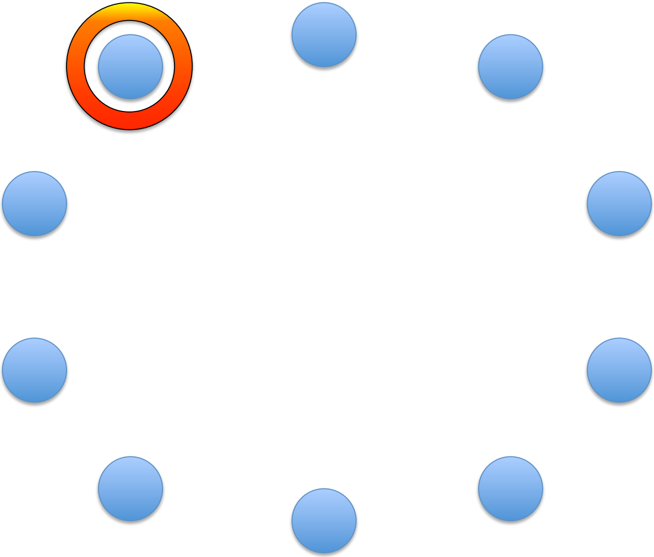
Introduction

Introducing Firewalls and NAT boxes



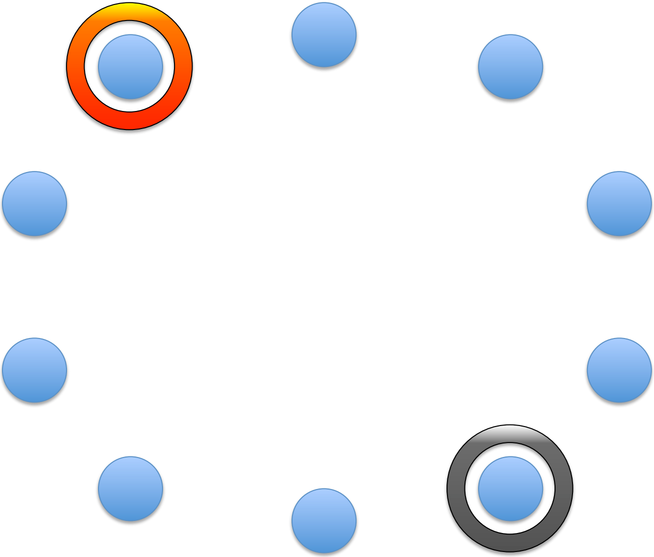
Introduction
Introducing Firewalls and NAT boxes





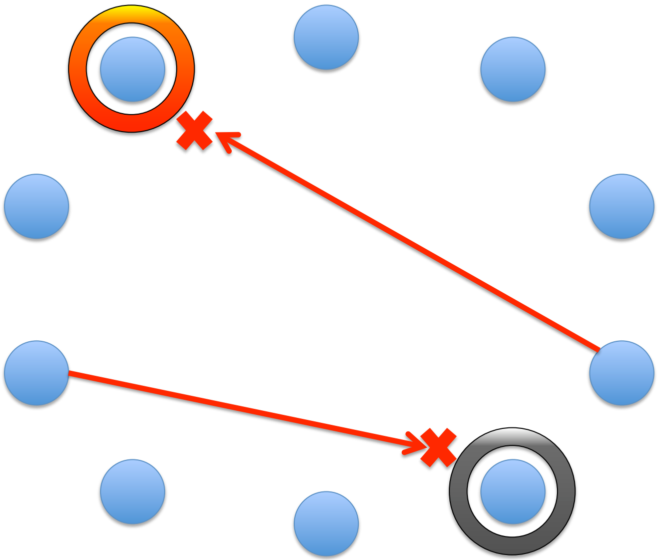
Introduction

Introducing Firewalls and NAT boxes



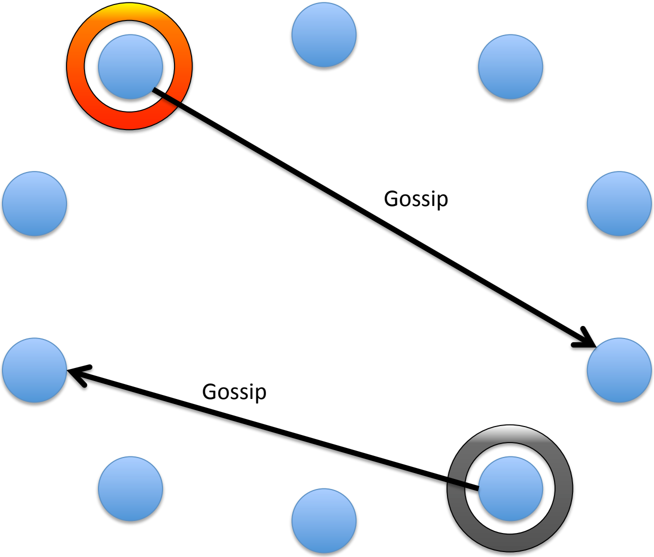
Introduction

Introducing Firewalls and NAT boxes



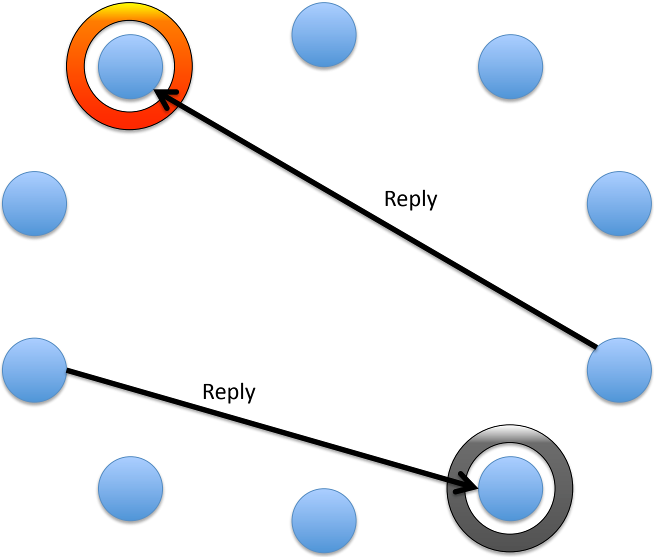
Introduction

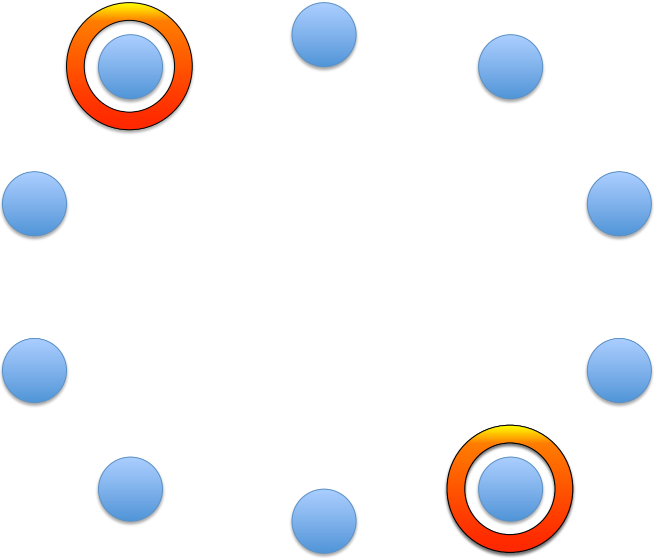
Introducing Firewalls and NAT boxes



Introduction

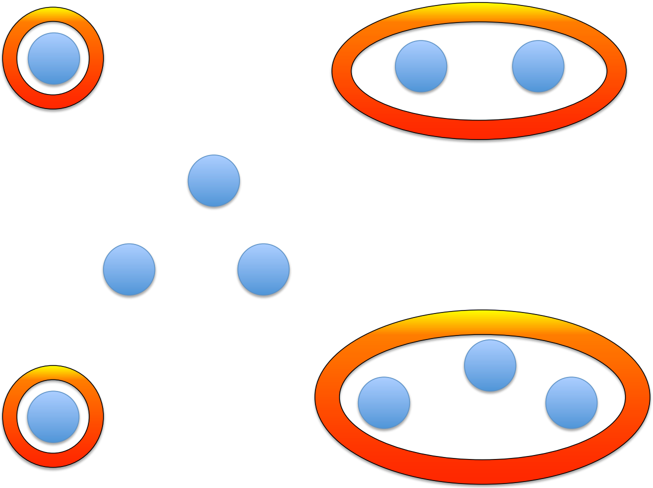
Introducing Firewalls and NAT boxes



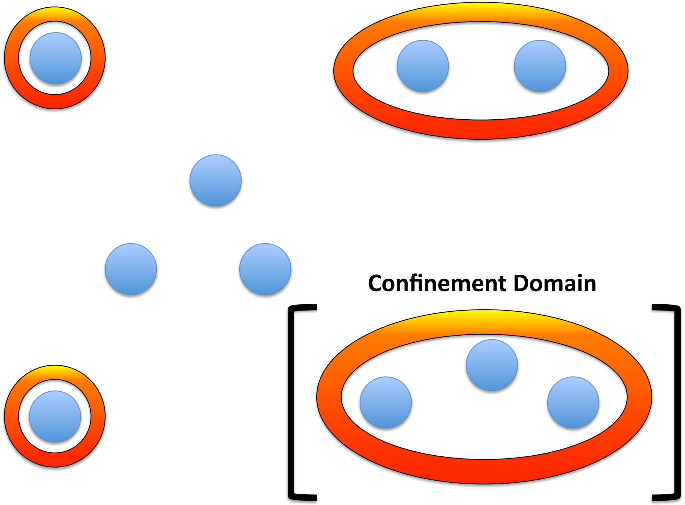


Introduction

Confinement Domain & Unconfined Nodes

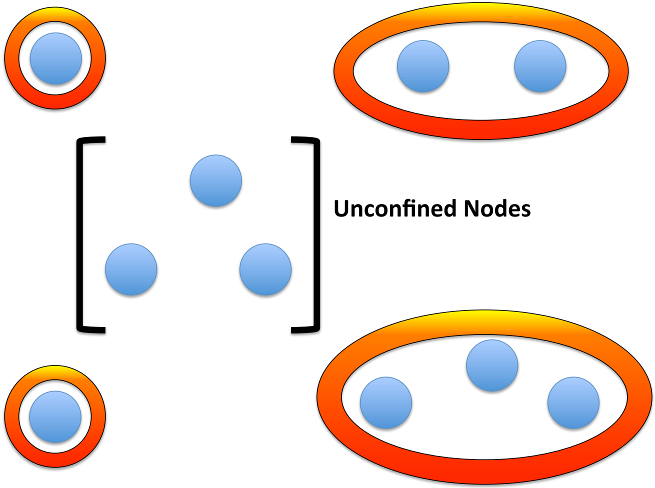


Introduction
Confinement Domain & Unconfined Nodes

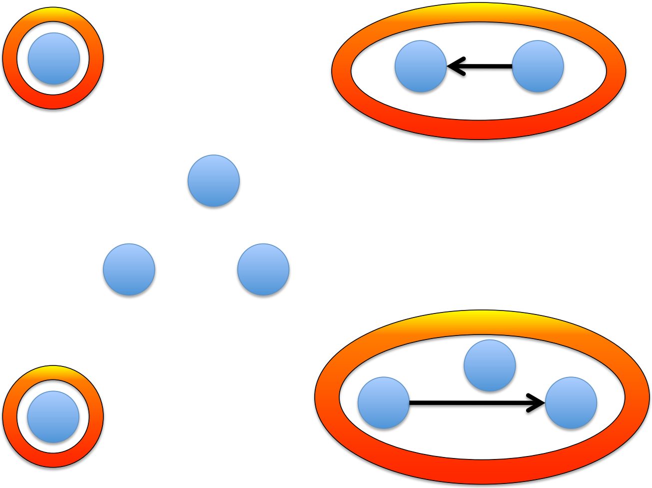


Introduction

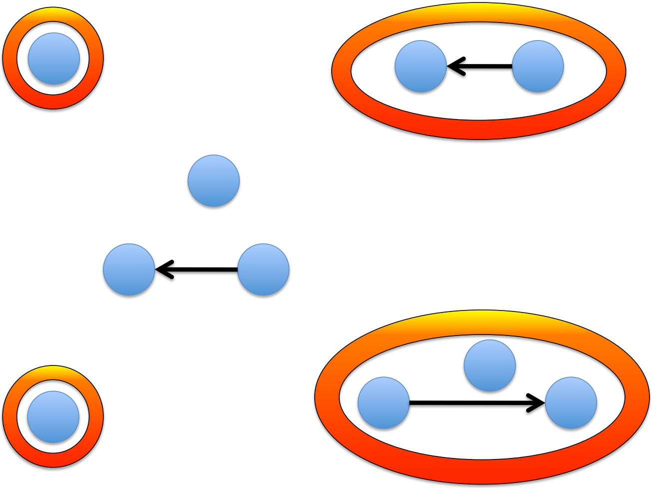
Confinement Domain & Unconfined Nodes



Introduction
Confinement Domain & Unconfined Nodes

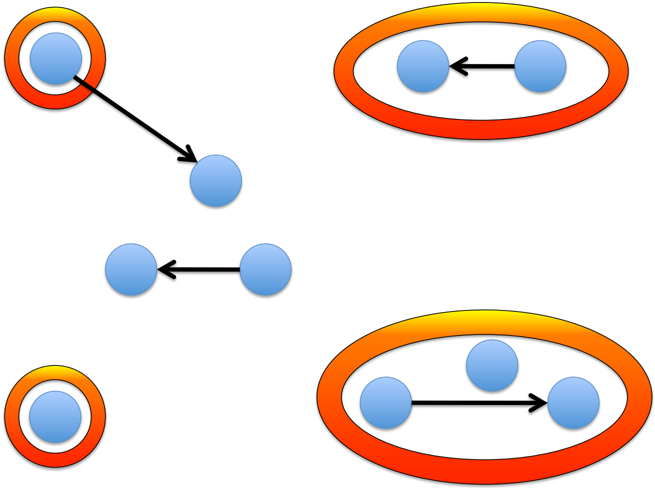


Introduction
Confinement Domain & Unconfined Nodes



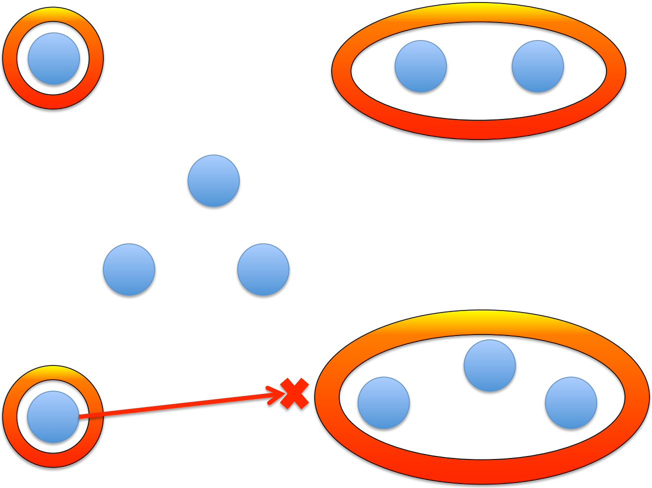
Introduction

Confinement Domain & Unconfined Nodes

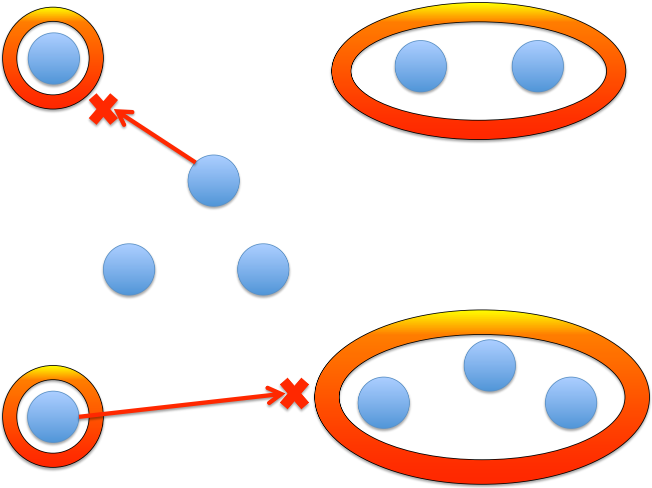


Introduction

Confinement Domain & Unconfined Nodes



Introduction
Confinement Domain & Unconfined Nodes



This can unbalance the system behavior:

- Unconfined nodes can participate in a much higher number of gossip exchanges.
- Specially when only a small fraction of nodes are unconfined.

This unbalance is undesirable:

- State reconciliation can require significant CPU Resources:
  - Techniques to reduce the use of bandwidth.
  - Encryption/decryption and signature/verification of messages.
  - Serialization/deserialization of objects.

This can unbalance the system behavior:

- Unconfined nodes can participate in a much higher number of gossip exchanges.
- Specially when only a small fraction of nodes are unconfined.

This unbalance is undesirable:

- State reconciliation can require significant CPU Resources:
  - Techniques to reduce the use of bandwidth.
  - Encryption/decryption and signature/verification of messages.
  - Serialization/deserialization of objects.

Introduction

Contribution

In this paper:

- We present a new approach to balance gossip exchanges in networks with firewalls.
  - only requires local information.
  - no coordination overhead.
  - nodes are not required to know if they are unconfined or confined.

In this paper:

- We present a new approach to balance gossip exchanges in networks with firewalls.
  - only requires local information.
  - no coordination overhead.
  - nodes are not required to know if they are unconfined or confined.

Outline

We follow 2 observations.

Observation 1:

Two nodes in distinct confinement domains can only exchange information through an unconfined node.

Observation 2:

In a balanced system on average:
For each gossip exchange initiated by a node (on average) that node participates in a gossip exchange initiated by another peer.

We follow 2 observations.

Observation 1:

Two nodes in distinct confinement domains can only exchange information through an unconfined node.

Observation 2:

In a balanced system on average:
For each gossip exchange initiated by a node (on average) that node participates in a gossip exchange initiated by another peer.

We follow 2 observations.

Observation 1:

Two nodes in distinct confinement domains can only exchange information through an unconfined node.

Observation 2:

In a balanced system on average:
For each gossip exchange initiated by a node (on average) that node participates in a gossip exchange initiated by another peer.

Balancing Gossip

Intuition

- Each node maintains:
  - A quota value (initially with a value of 1).
  - A single-entry cache for connections created by other nodes.

Every node in the system executes the same protocol.

- Each node maintains:
  - A quota value (initially with a value of 1).
  - A single-entry cache for connections created by other nodes.

Every node in the system executes the same protocol.

- The quota limits the number of gossip exchanges initiated by other peers that a node can accept.
- Nodes increase their quota when they initiate a gossip exchange.
- The connection cache keeps alive the last connection used by another peer to initiate a gossip exchange.
- When a node receives a gossip request and does not have a quota value above zero it forwards the request through the cached connection.

Balancing Gossip
Intuition

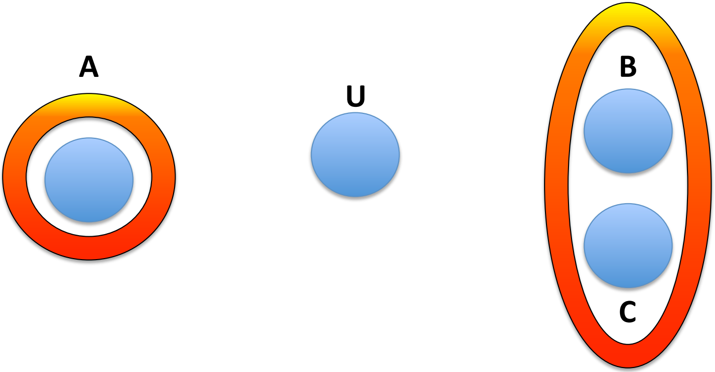
- The quota limits the number of gossip exchanges initiated by other peers that a node can accept.
- Nodes increase their quota when they initiate a gossip exchange.
- The connection cache keeps alive the last connection used by another peer to initiate a gossip exchange.
- When a node receives a gossip request and does not have a quota value above zero it forwards the request through the cached connection.

Balancing Gossip
Intuition

- The quota limits the number of gossip exchanges initiated by other peers that a node can accept.
- Nodes increase their quota when they initiate a gossip exchange.
- The connection cache keeps alive the last connection used by another peer to initiate a gossip exchange.
- When a node receives a gossip request and does not have a quota value above zero it forwards the request through the cached connection.

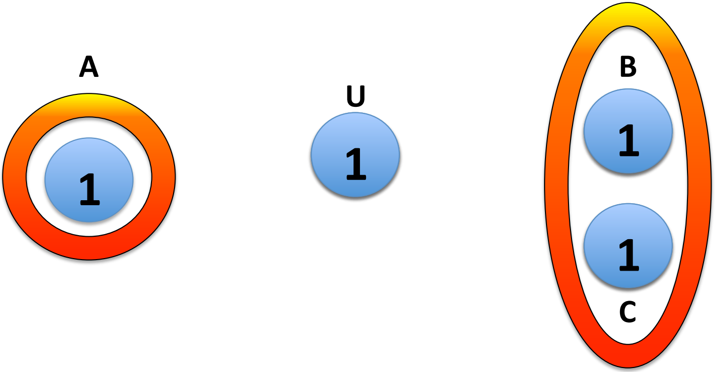
Balancing Gossip

Example

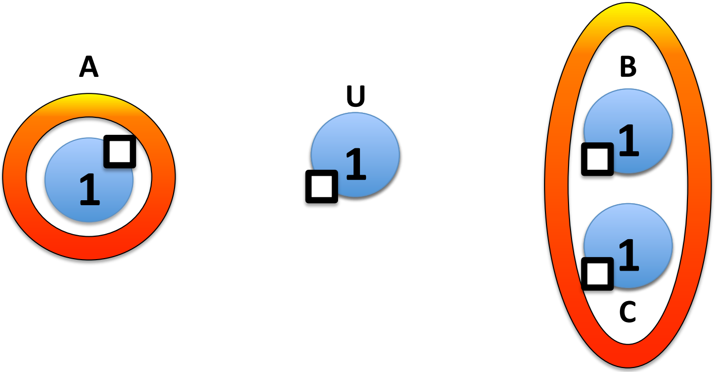


Balancing Gossip

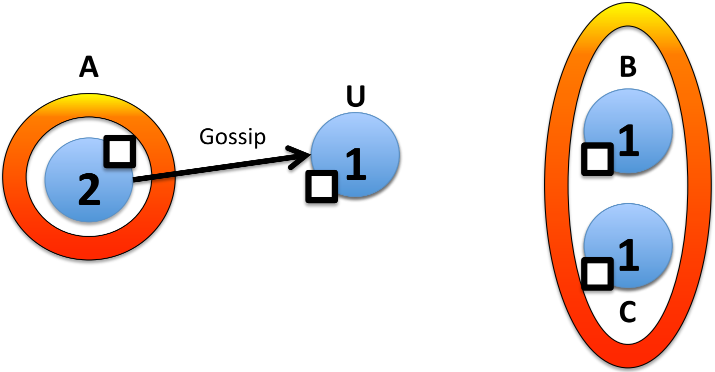
Example



Balancing Gossip
Example

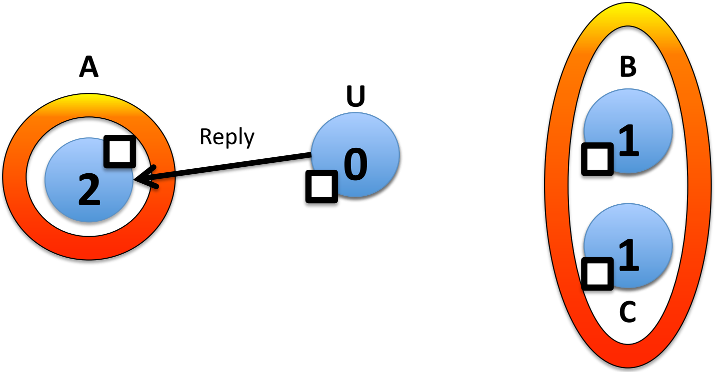


Balancing Gossip
Example



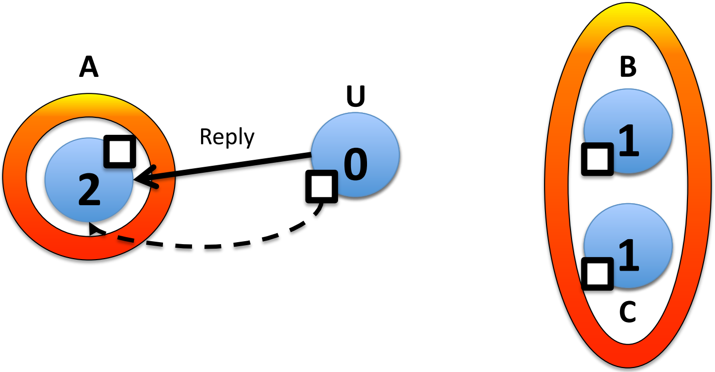
Balancing Gossip

Example

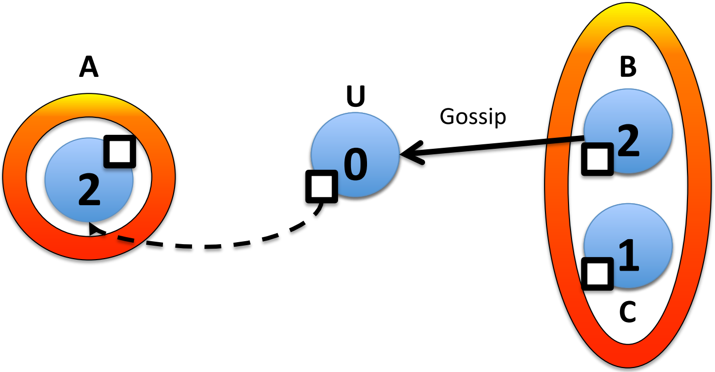


Balancing Gossip

Example

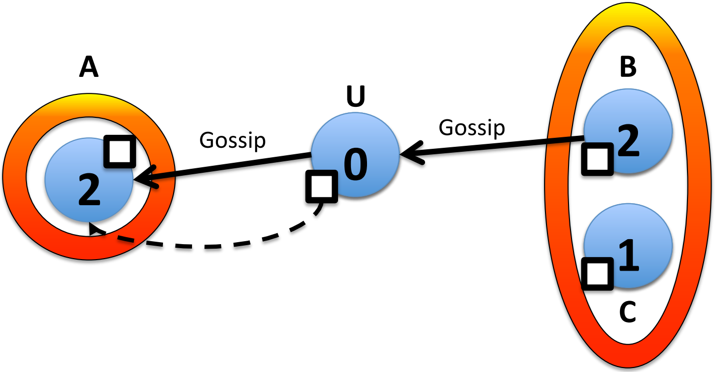


Balancing Gossip
Example

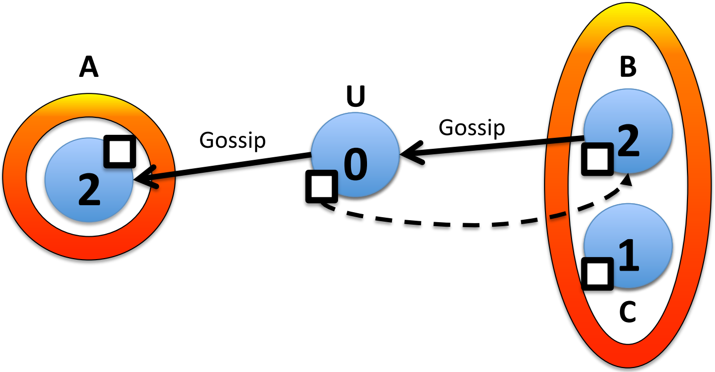


Balancing Gossip

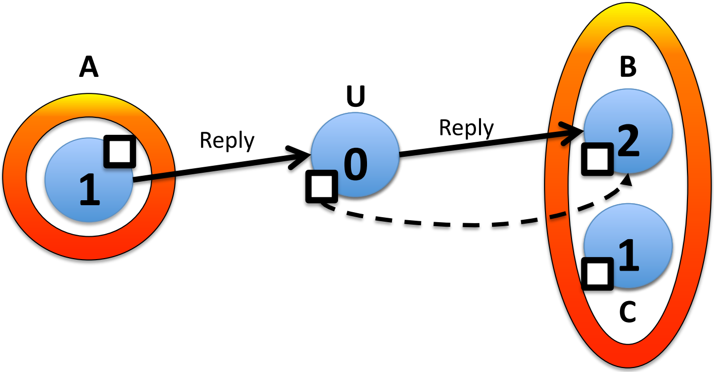
Example



Balancing Gossip
Example

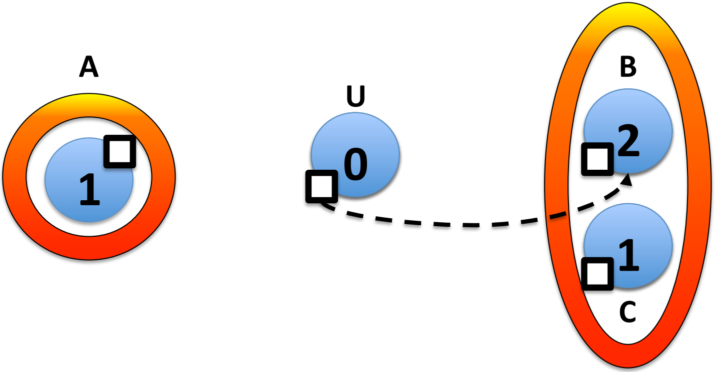


Balancing Gossip
Example



Balancing Gossip

Example



Balancing Gossip

Some additional aspects:

- A gossip requests are forwarded a limited number of times (TTL).
- If a node has an empty connection cache it engages in the gossip exchange.

Outline

Evaluation

Experimental Setting

- We conducted simulations in the Peersim simulator.
- System composed of 12.800 nodes.
- Distributed in a variable number of confinement domains:
- From 1 (flat topology) to 12.100 (star topology).
- Each communication step has a latency selected uniformly at random between 2 and 7.

Evaluation

Experimental Setting

- We conducted simulations in the Peersim simulator.
- System composed of 12.800 nodes.
- Distributed in a variable number of confinement domains:
- From 1 (flat topology) to 12.100 (star topology).
- Each communication step has a latency selected uniformly at random between 2 and 7.

Evaluation

Experimental Setting

- We conducted simulations in the Peersim simulator.
- System composed of 12.800 nodes.
- Distributed in a variable number of confinement domains:
- From 1 (flat topology) to 12.100 (star topology).
- Each communication step has a latency selected uniformly at random between 2 and 7.

Balancing Gossip Exchanges in Networks with Firewalls

J. Leitão, R. van Renesse and L. Rodrigues

Introduction

Balancing Gossip

Evaluation

Conclusions

Application

- Simple anti-entropy protocol.
  - All nodes have a state values initially set to 0.
  - A random node changes its state value to 1.
  - Nodes gossip their state value and update theirs with highest value.
- Each node initiates 500 gossip exchanges.
- If the system is balanced each node should participate in 1000 gossip exchanges.

Application

- Simple anti-entropy protocol.
  - All nodes have a state values initially set to 0.
  - A random node changes its state value to 1.
  - Nodes gossip their state value and update theirs with highest value.
- Each node initiates 500 gossip exchanges.
- If the system is balanced each node should participate in 1000 gossip exchanges.

Evaluation

Experimental Setting

- We evaluate our protocol using distinct TTL values:
  - TTL = 1 - Equivalent to regular gossip.
  - TTL = 2 - Each gossip request can be redirected one time.
  - TTL = 5.
  - TTL = 10.

Evaluation

Experimental Setting

- We evaluate our protocol using distinct TTL values:
  - TTL = 1 - Equivalent to regular gossip.
  - TTL = 2 - Each gossip request can be redirected one time.
  - TTL = 5.
  - TTL = 10.

Results show:

- Maximum latency (time until all nodes update their state value to 1).
- Maximum gossip exchanges performed by a single node.
- Maximum number of nessages forwarded by a single node.

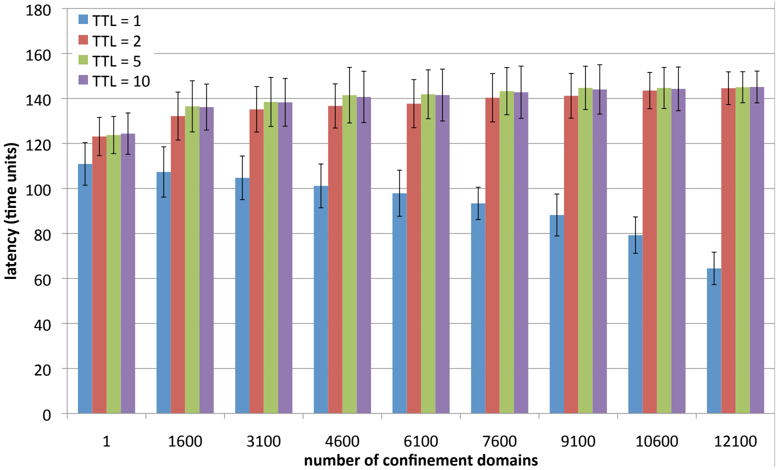
Results show:

- Maximum latency (time until all nodes update their state value to 1).
- Maximum gossip exchanges performed by a single node.
- Maximum number of nessages forwarded by a single node.

Results show:

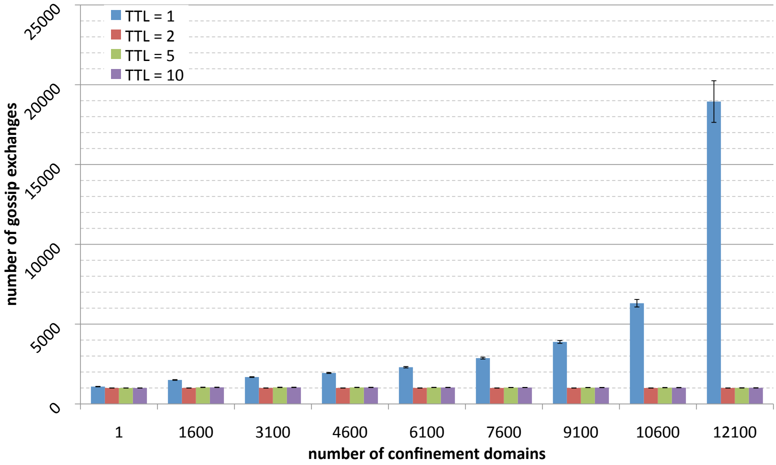
- Maximum latency (time until all nodes update their state value to 1).
- Maximum gossip exchanges performed by a single node.
- Maximum number of nessages forwarded by a single node.

Evaluation
Experimental Results: Maximum latency



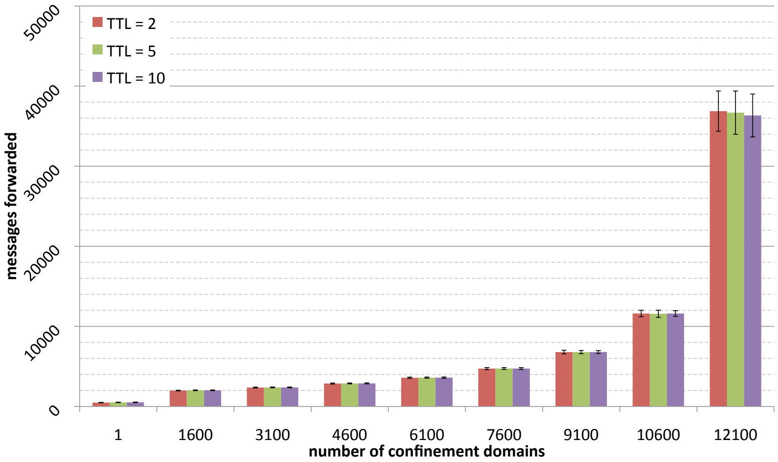
Evaluation

Experimental Results: Maximum gossip exchanges per node



Evaluation

Experimental Results: Maximum forwarded messages per node



Outline

Conclusions

- We have studied how to balance gossip exchanges in networks with firewalls.
- We proposed a new solution:
  - Effectively balances gossip exchanges.
  - Does not require nodes to know if they are confined or unconfined.
  - Has no coordination overhead.
- This technique can be easily implemented in current gossip-based mechanisms.

Conclusions

- We have studied how to balance gossip exchanges in networks with firewalls.
- We proposed a new solution:
  - Effectively balances gossip exchanges.
  - Does not require nodes to know if they are confined or unconfined.
  - Has no coordination overhead.
- This technique can be easily implemented in current gossip-based mechanisms.

Thanks.