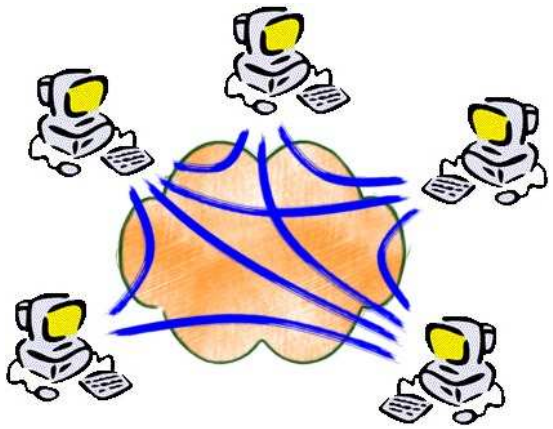
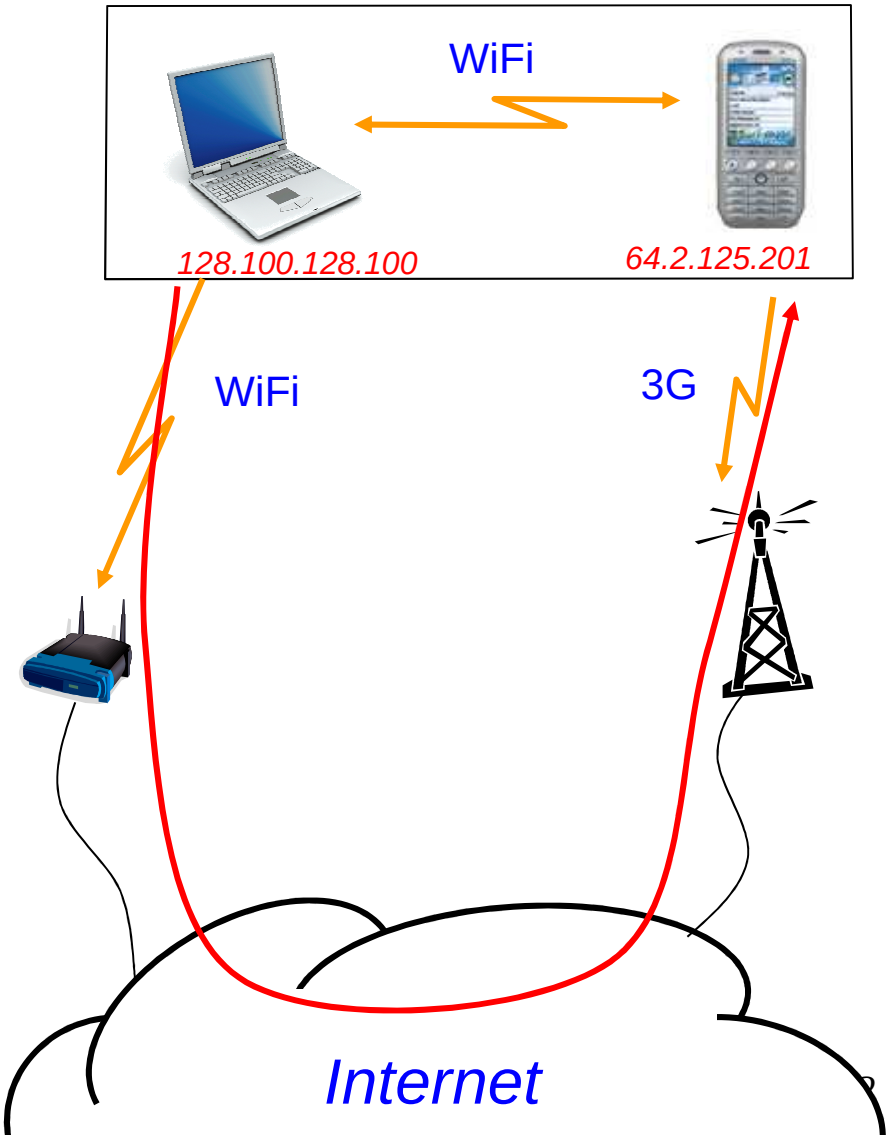
Dissemination of Address Bindings in Multi-substrate Overlay Networks

Jorg Liebeherr *Majid Valipour*

University of Toronto



Internet-centric Networking



Example:

Laptop 1: Macbook with 3G interface

Laptop 2: Thinkpad with Ethernet interface

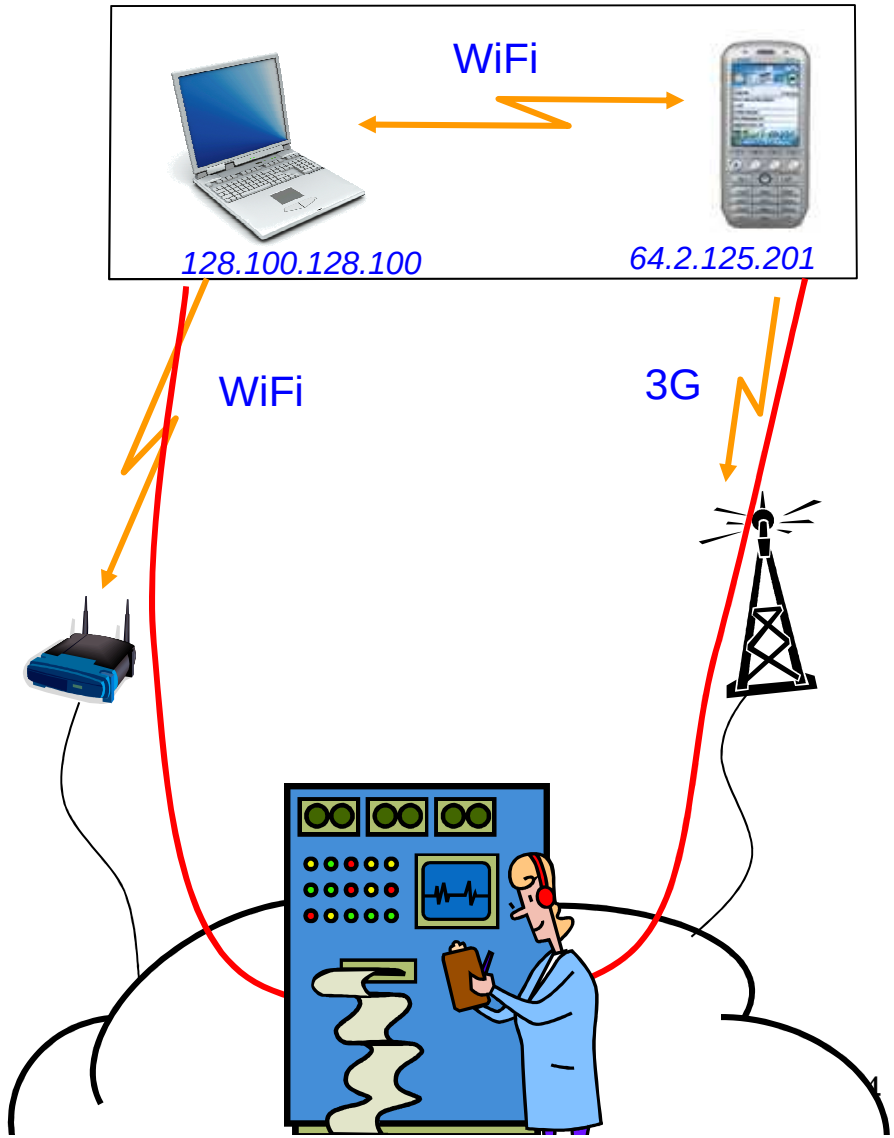
Distance <1m

jorg:~$ traceroute 130.149.220.38

```
traceroute to 130.149.220.38 (130.149.220.38), 64 hops max, 40 byte packets
 1  82.113.122.185 (82.113.122.185)  311.829 ms  339.046 ms  350.075 ms
 2  82.113.123.226 (82.113.123.226)  369.430 ms  339.371 ms  349.988 ms
 3  OSRMUN1-Vl212.net.de.o2.com (82.113.122.53)  339.959 ms  369.970 ms  369.174 m
 4  IARMUN1-Gi0-2-199.net.de.o2.com (82.113.122.2)  349.670 ms  335.778 ms  349.720
 5  IARMUN2-Gi-0-1.net.de.o2.com (82.113.122.58)  329.990 ms  359.114 ms  359.717 ms
 6  xmwc-mnch-de02-gigaet-2-26.nw.mediaways.net (195.71.164.225)  319.921 ms  338.7
 7  zr-fra1-ge0-2-0-5.x-win.dfn.de (188.1.231.93)  369.923 ms  389.062 ms  409.913 ms
 8  zr-pot1-te0-7-0-2.x-win.dfn.de (188.1.145.138)  419.876 ms  449.205 ms  439.918 ms
 9  xr-tub1-te2-3.x-win.dfn.de (188.1.144.222)  429.851 ms  409.338 ms  429.804 ms
10  kr-tu-berlin.x-win.dfn.de (188.1.33.82)  439.768 ms  439.172 ms  450.061 ms
11  t-labs.gate.TU-Berlin.DE (130.149.235.15)  449.921 ms  418.957 ms  409.705 ms
12  * * *
13  * * *
14  * * *
```

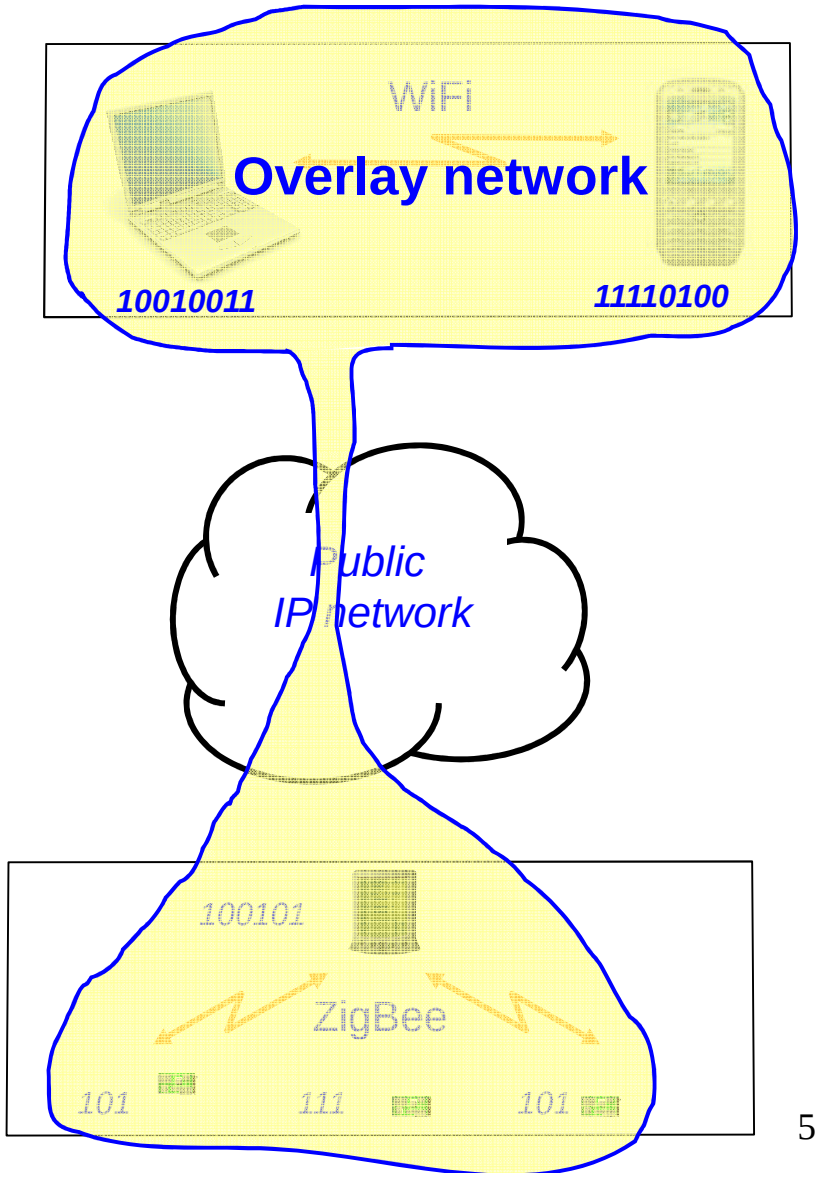
An Analogy

Internet today
appears like
Mainframe computing of 1970s



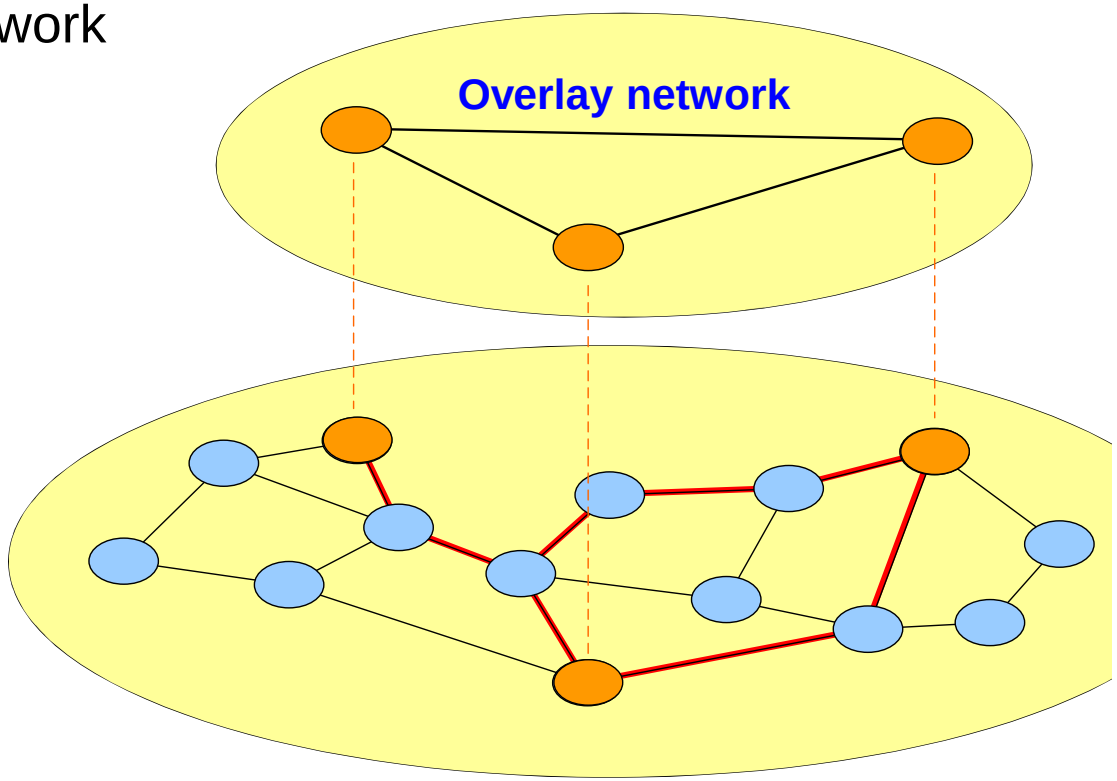
Overlay-based Networking

- Applications/devices self-organize as a network
- Application networks define their own address space
- Access infrastructure only if needed

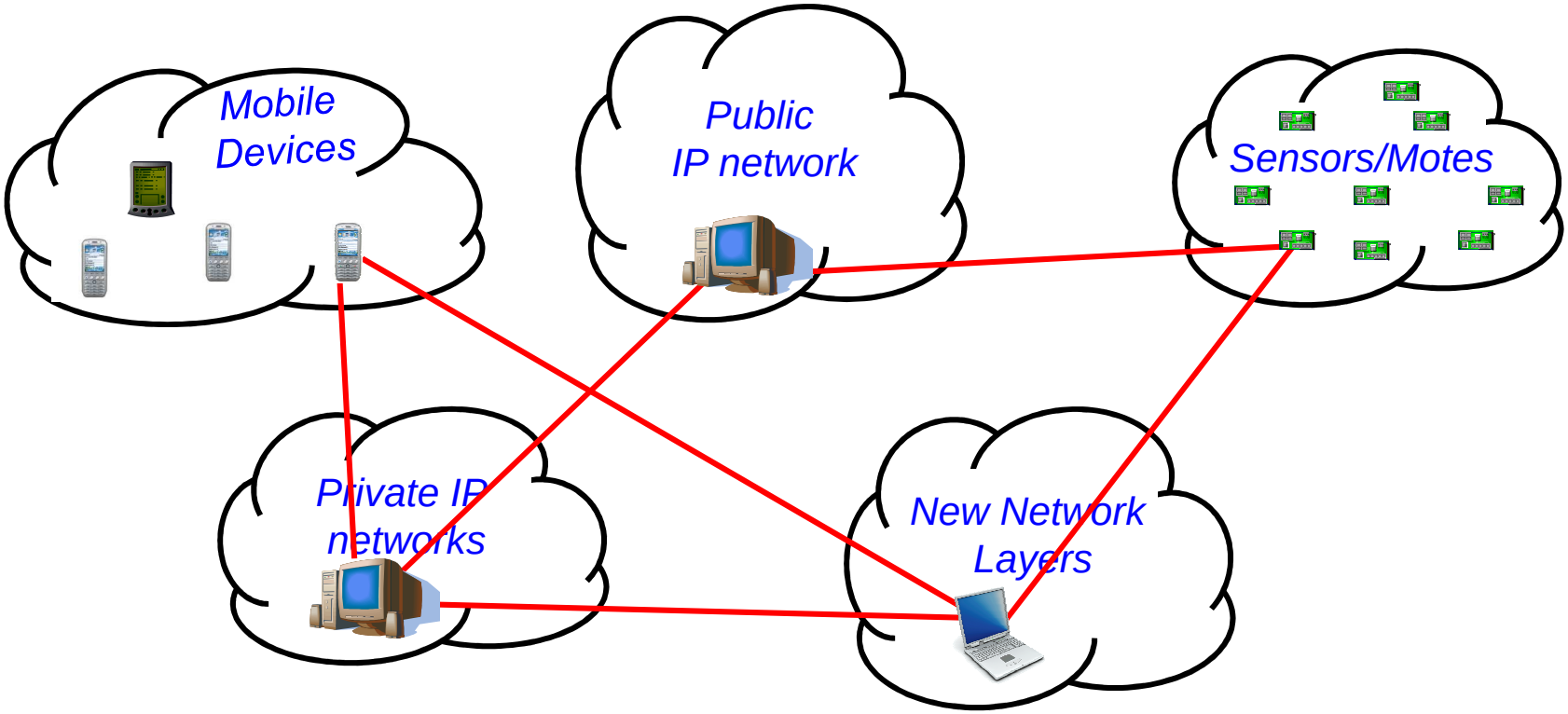


What is an overlay anyway?

- An overlay network is a virtual network of nodes and logical links built on top of an existing network
- A virtual link in the overlay corresponds to a path in the underlay

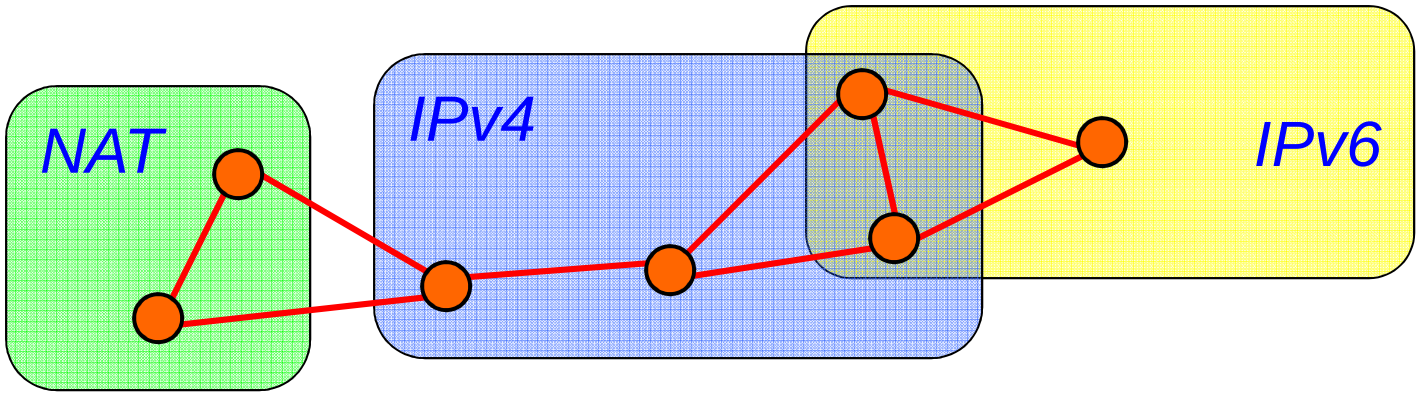


Multiple Substrate

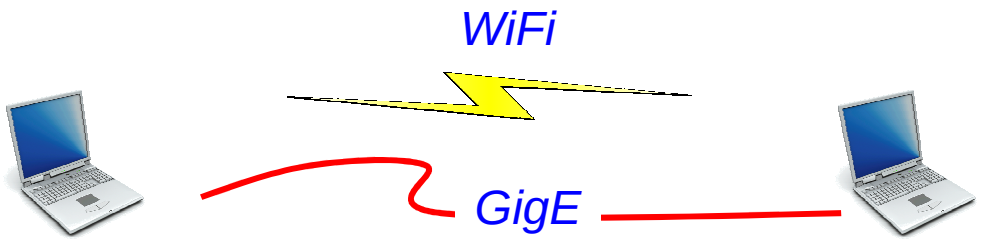


- Connection to a single infrastructure/address space not feasible or desirable
- **Objective:**
  Build self-organizing overlay network over any collection of substrates

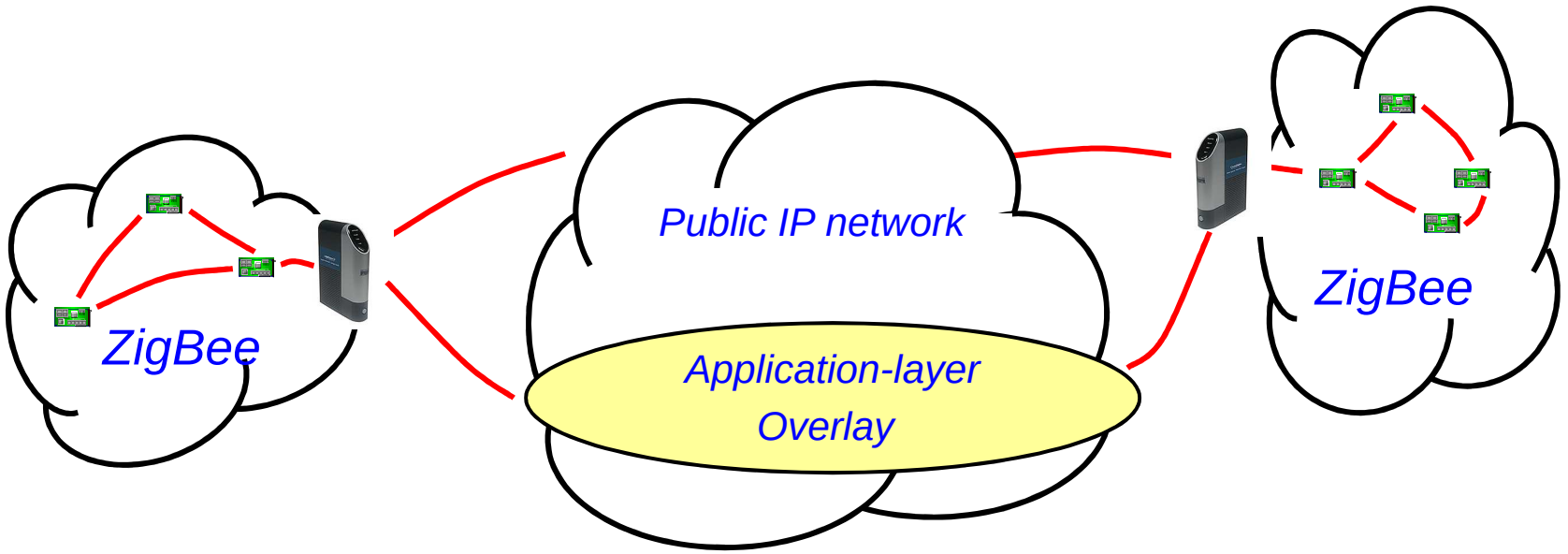
Multi-substrate Networks



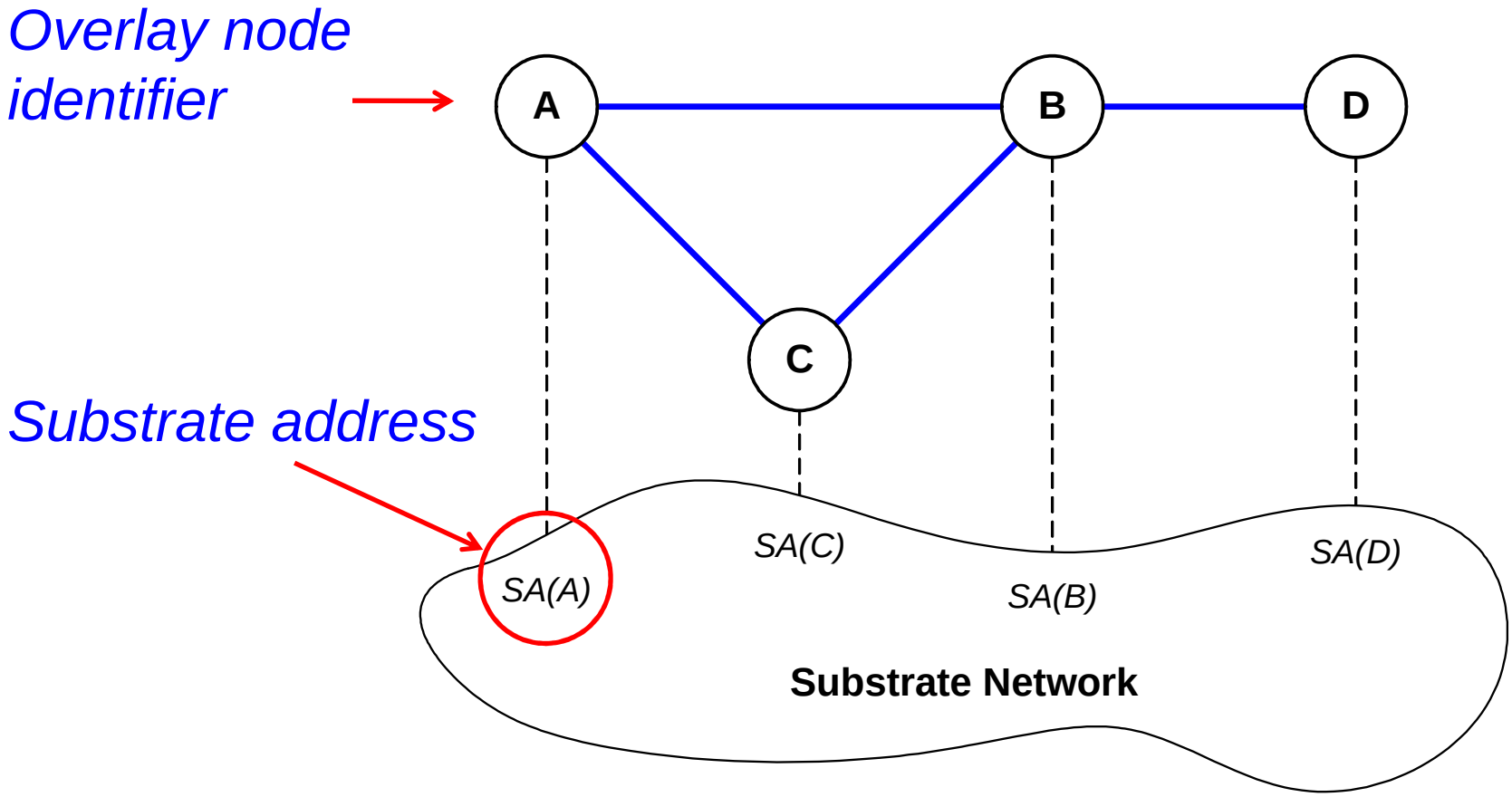
Multi-substrate Networks



Multi-substrate Networks

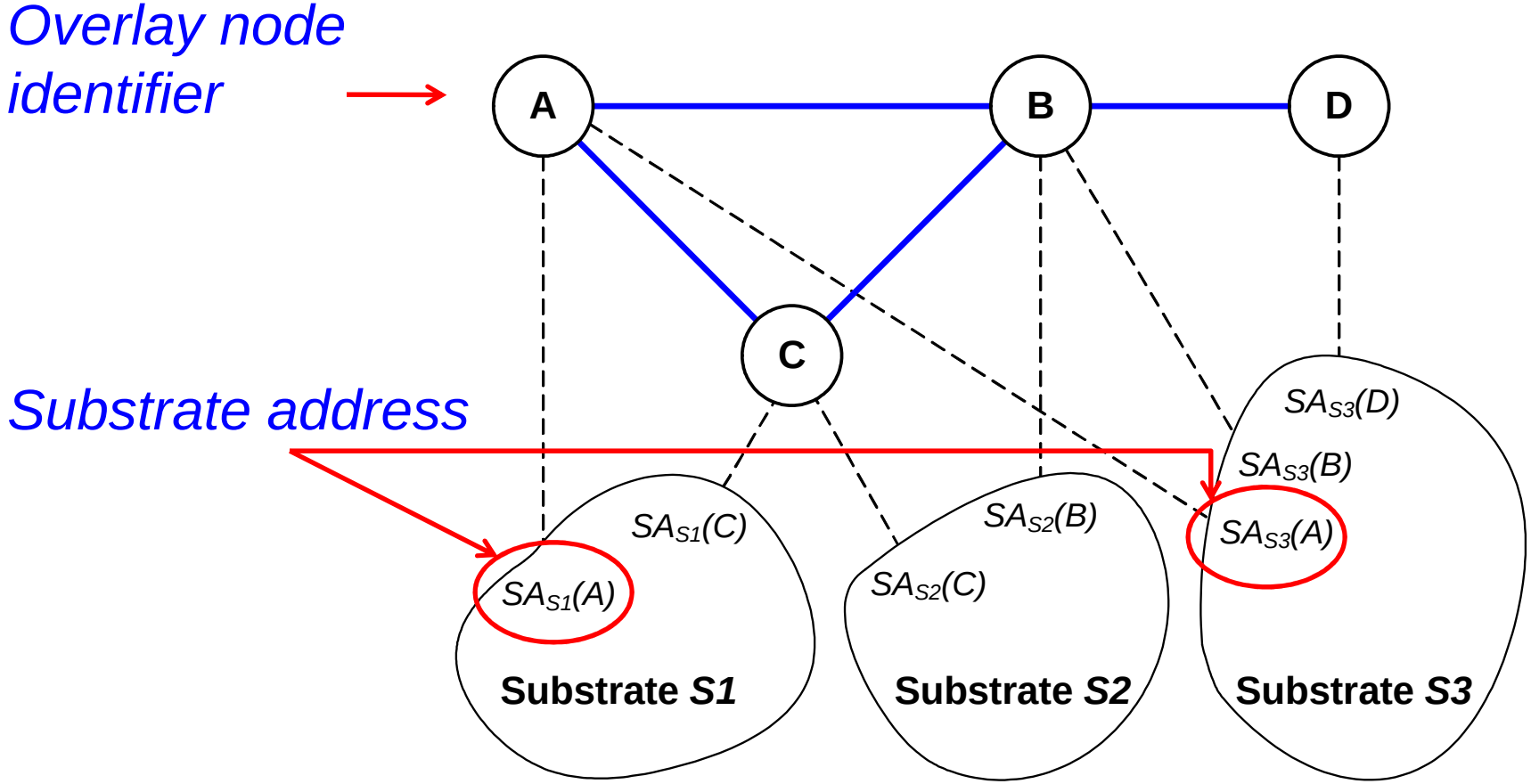


Address Bindings in Single-substrate Overlay



- Address binding: **[A; SA(A)]**

Address Bindings in Multi-substrate Overlay



- More complex address binding: **[A; $SA_{S1}(A)$, $SA_{S3}(A)$]**

Cross Substrate Advertising

- To send a message over a substrate network to a destination, a node must use the proper binding for that destination

Problem:

How to efficiently propagate information about address bindings?

Solution:

Protocol mechanisms for exchanging address bindings

→ **Cross Substrate Advertisement (CSA)**

This paper:

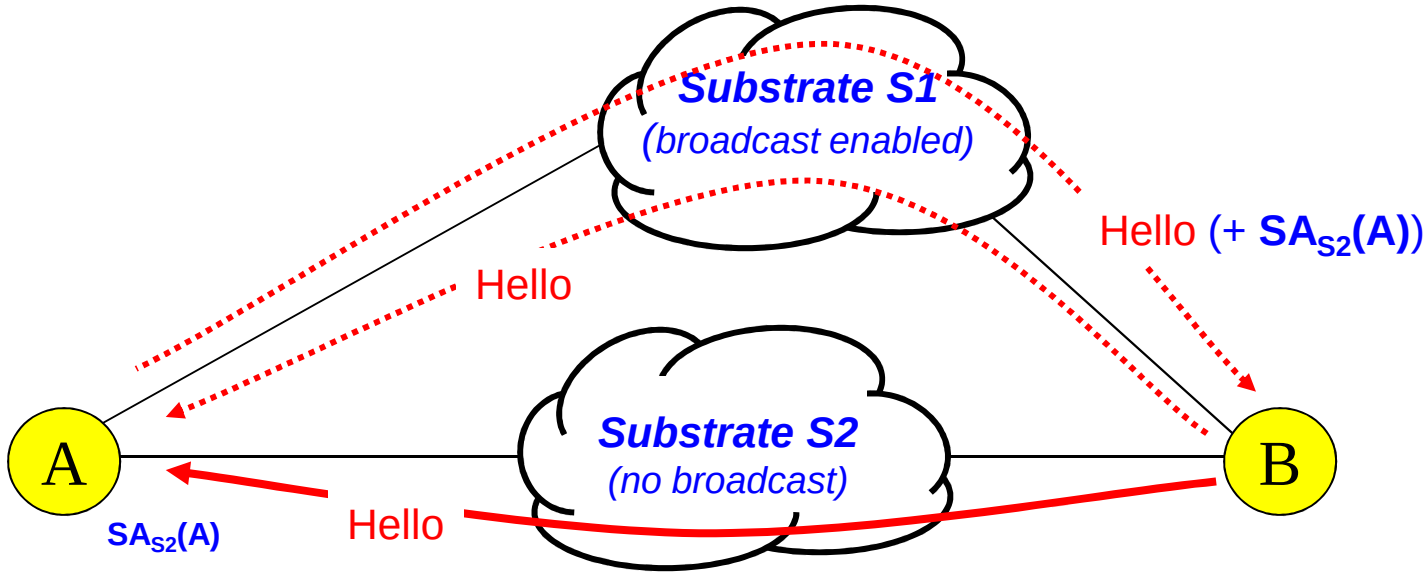
Design and evaluate protocol mechanisms for CSA

CSA Mechanism: Direct Exchange

Direct Exchange: Use directly connected substrates to exchange address bindings

Example:

- B prefers non-broadcast Substrate 2, but does not have needed address
- Use broadcast-enabled Substrate 1 to advertise address for Substrate 2

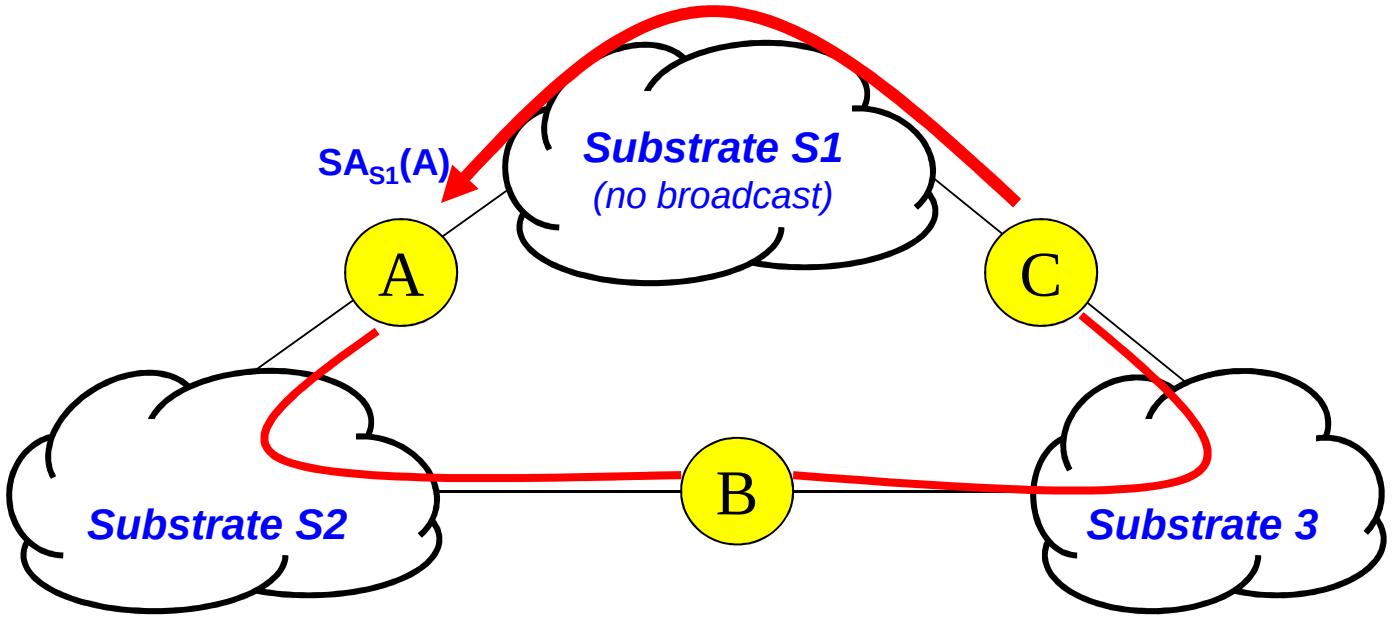


CSA Mechanism: Relayed Exchange

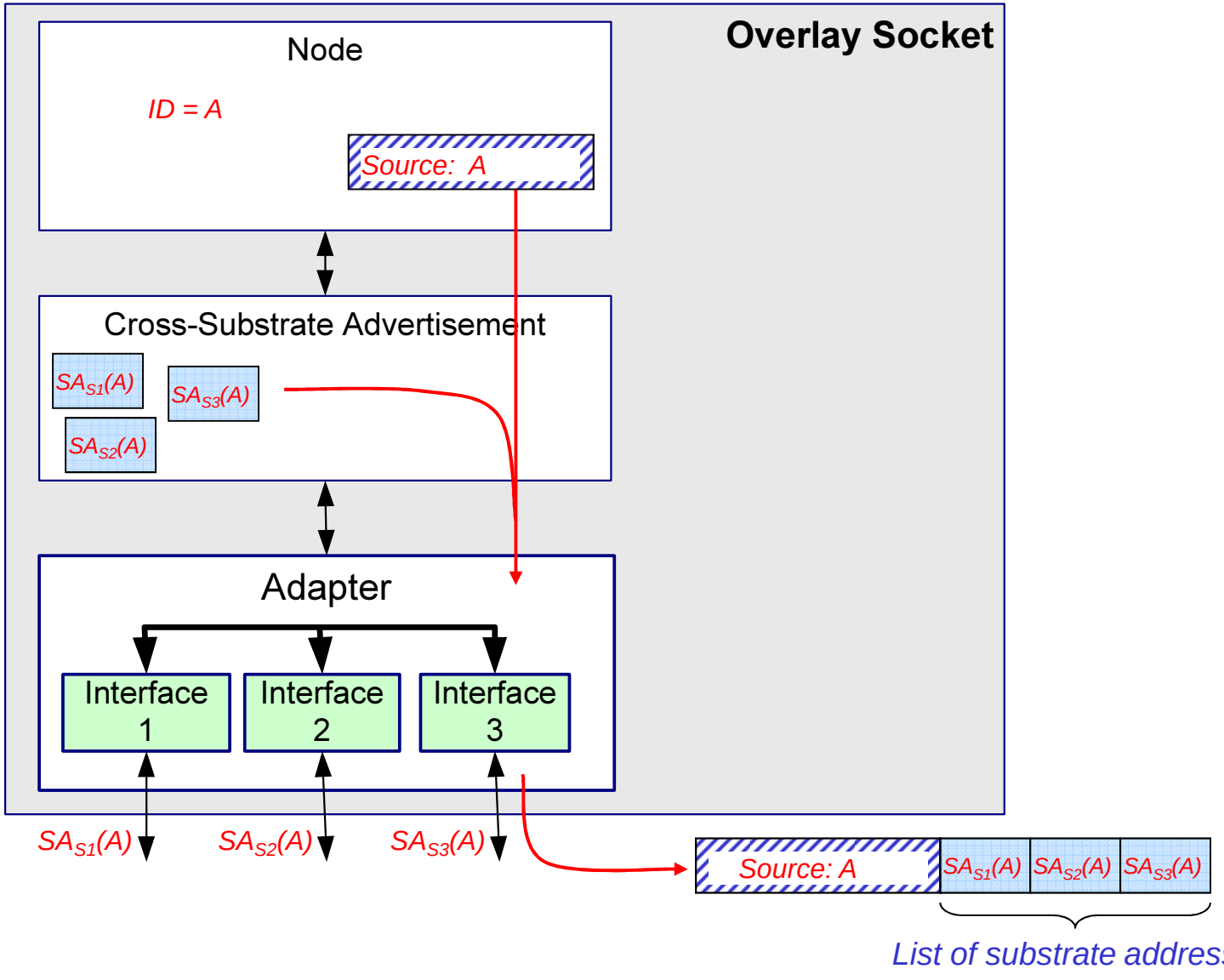
Relayed Exchange: Intermediate nodes forward address binding information

Example:

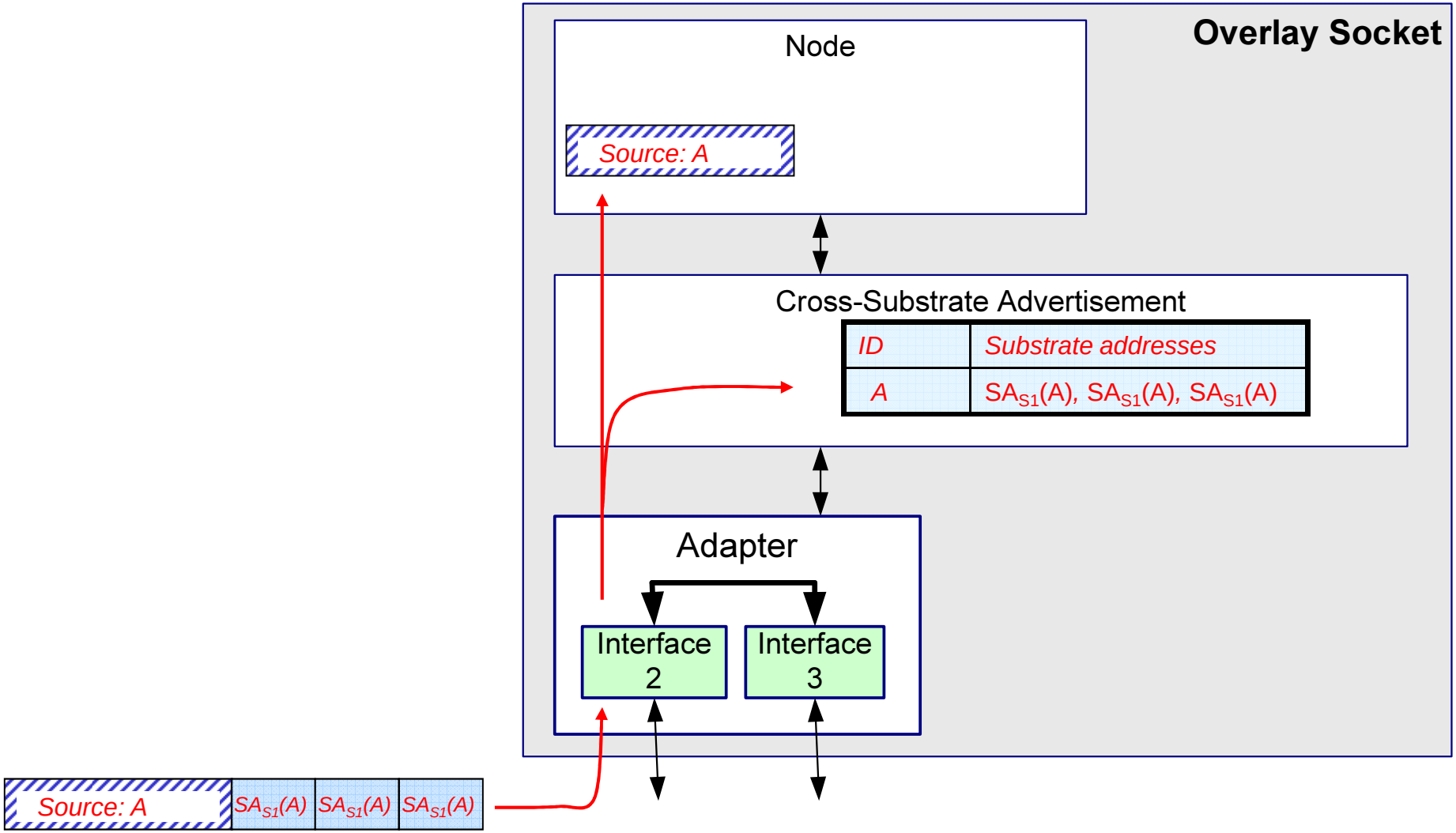
- C joins the overlay network on Substrate 3, but prefers to use Substrate 1
- A forwards its address bindings to B, which relays it to C



Cross Substrate Advertising *(simplified)* - Outgoing -



Cross Substrate Advertising *(simplified)*
- Incoming -



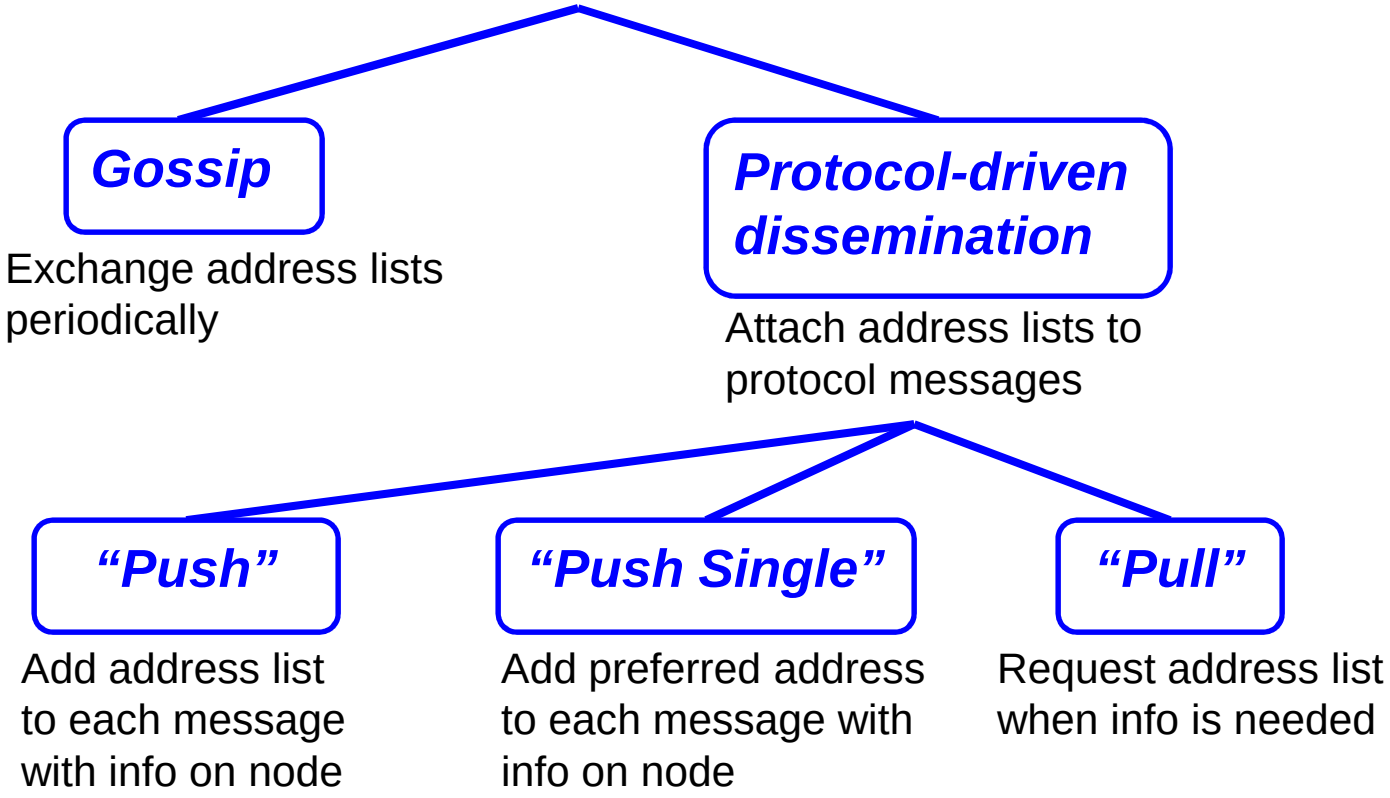
Implementation

- CSA Protocol realized as part of an open source overlay software system (www.hypercast.org)

- Implemented as a layer:
  - Mechanisms are independent of protocol that builds topology

- Considers preference for substrates
- CSA message types:
  - Request address list
  - Update

Evaluate Methods for Relayed Address Exchange



Experimental Evaluation

- Local *Emulab* Testbed
- 20 Linux nodes

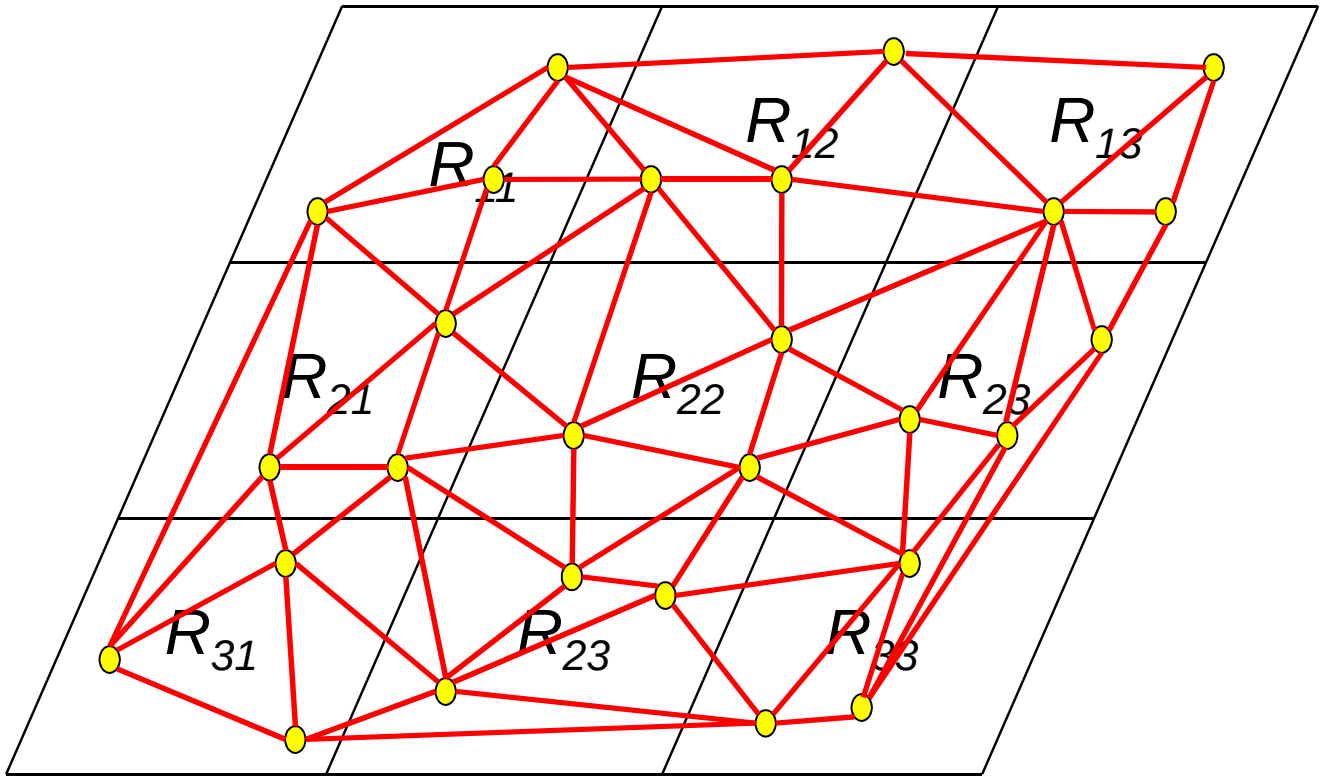
CPU	2xQuad-Core Xeon 5400 (2 Ghz)
RAM	4G DDR2
Interface	8x1Gbps (4 Intel card, 4 NetFPGA)

- Software:
  - *Hypercast with CSA*
  - *Delaunay Triangulation protocol*
  - *Multiple UDP/IP substrates*

Mapping of Nodes to Substrates

- K x K substrates → (K+1) x (K+1) regions
- Nodes distributed uniformly across regions

K=2



Performance metrics

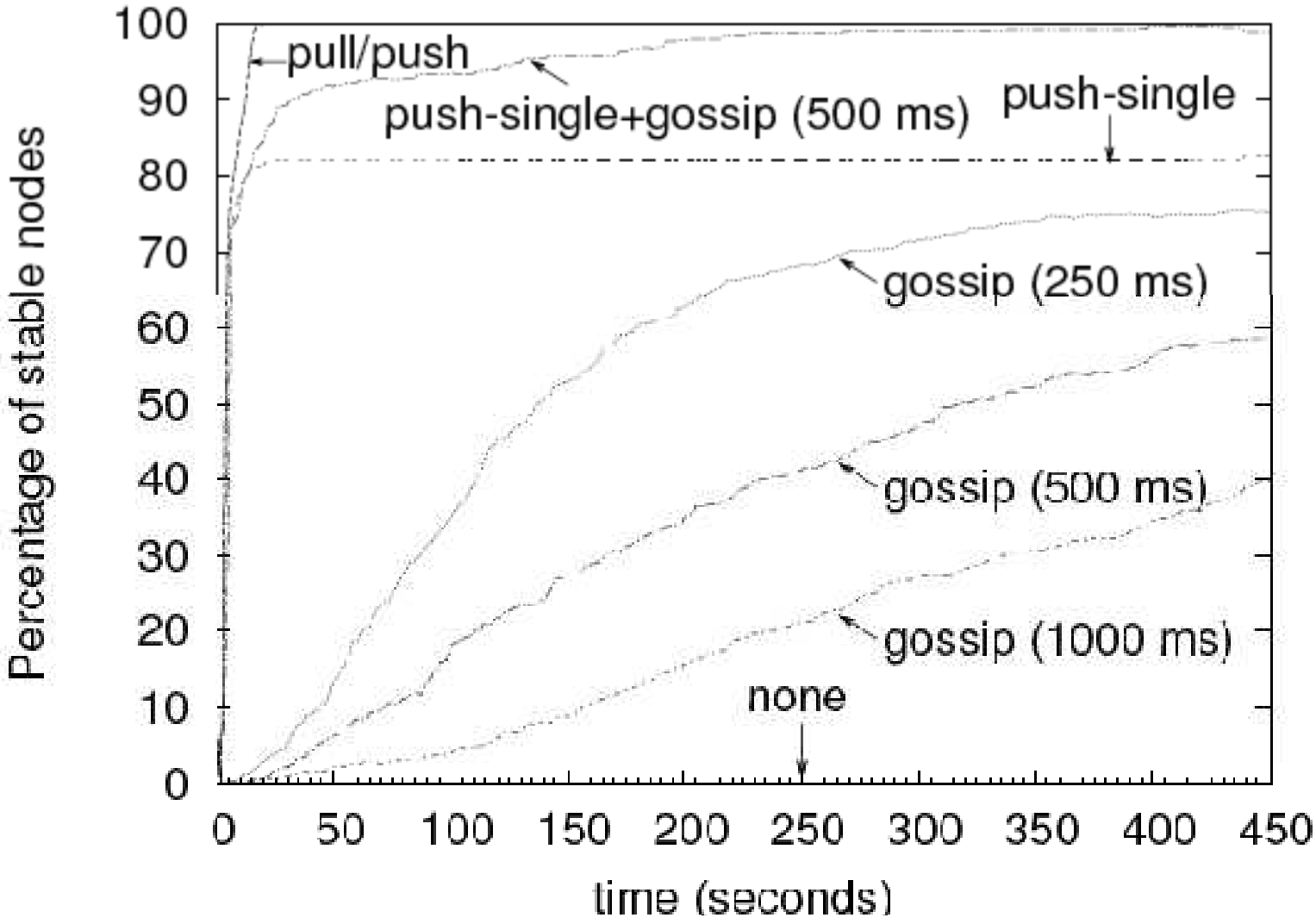
- **Stability:**
  Do nodes reach a stable state in overlay topology ?
  - % of nodes satisfying stability criterion for local neighborhood

- **Connectivity: Do nodes form a single overlay network?**
  - # of partitioned topologies

A single stable overlay topology has formed when
(1) 100% of nodes are stable; and
(2) there is one topology

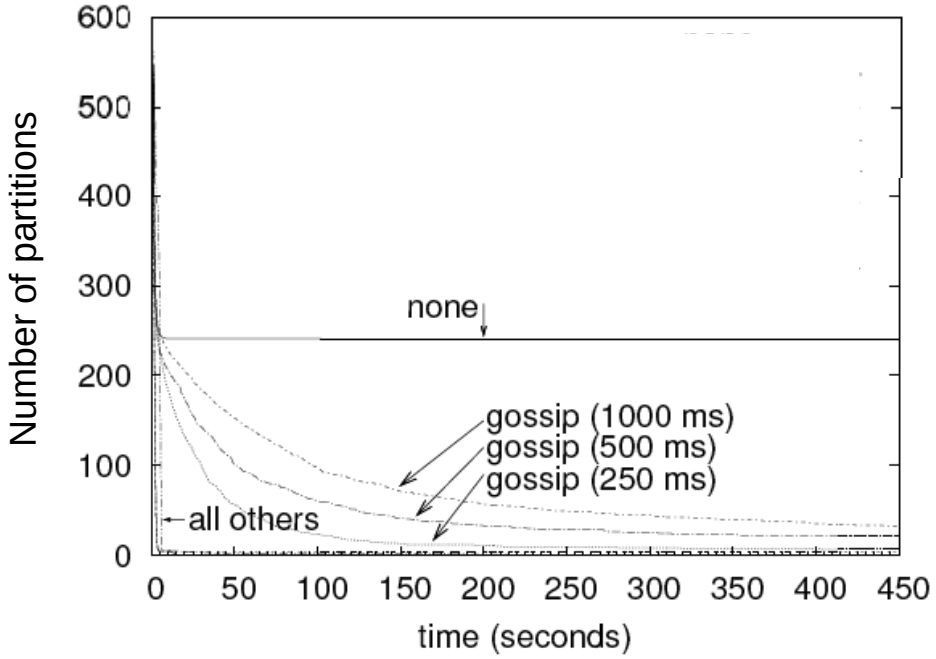
Stability

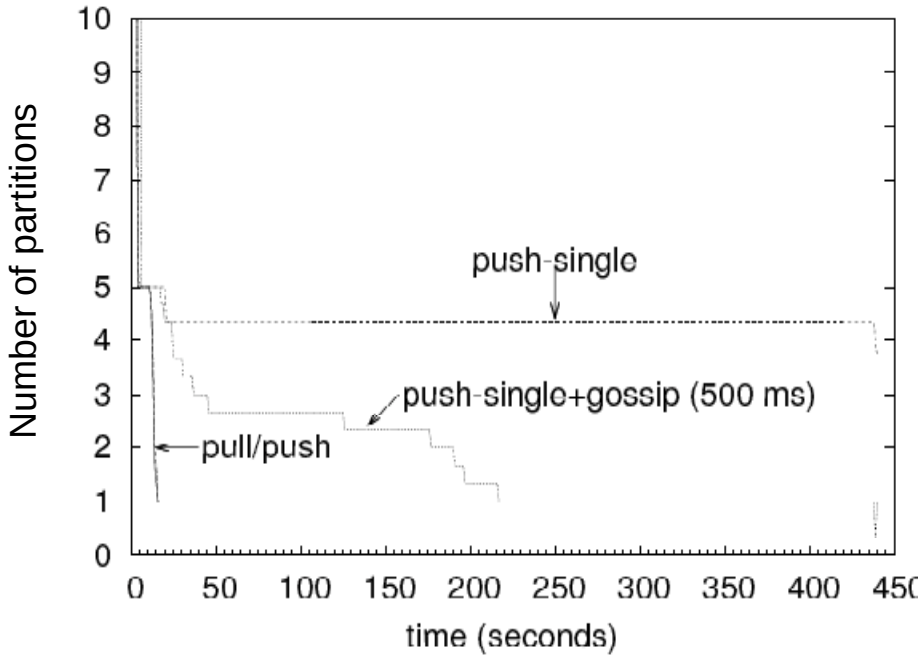
K=8 (64 substrates), 648 nodes



Connectivity

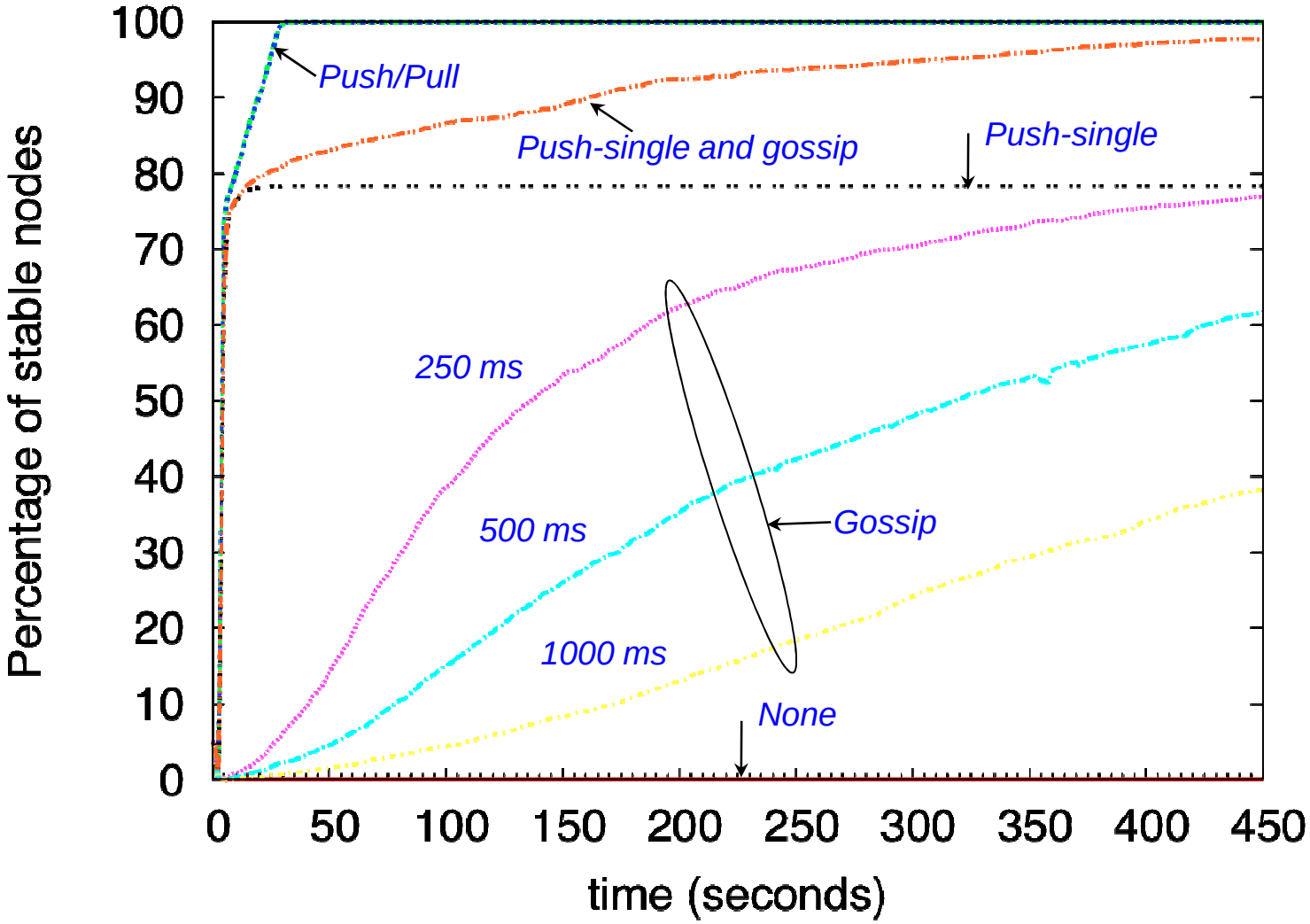
K=8 (64 substrates), 648 nodes





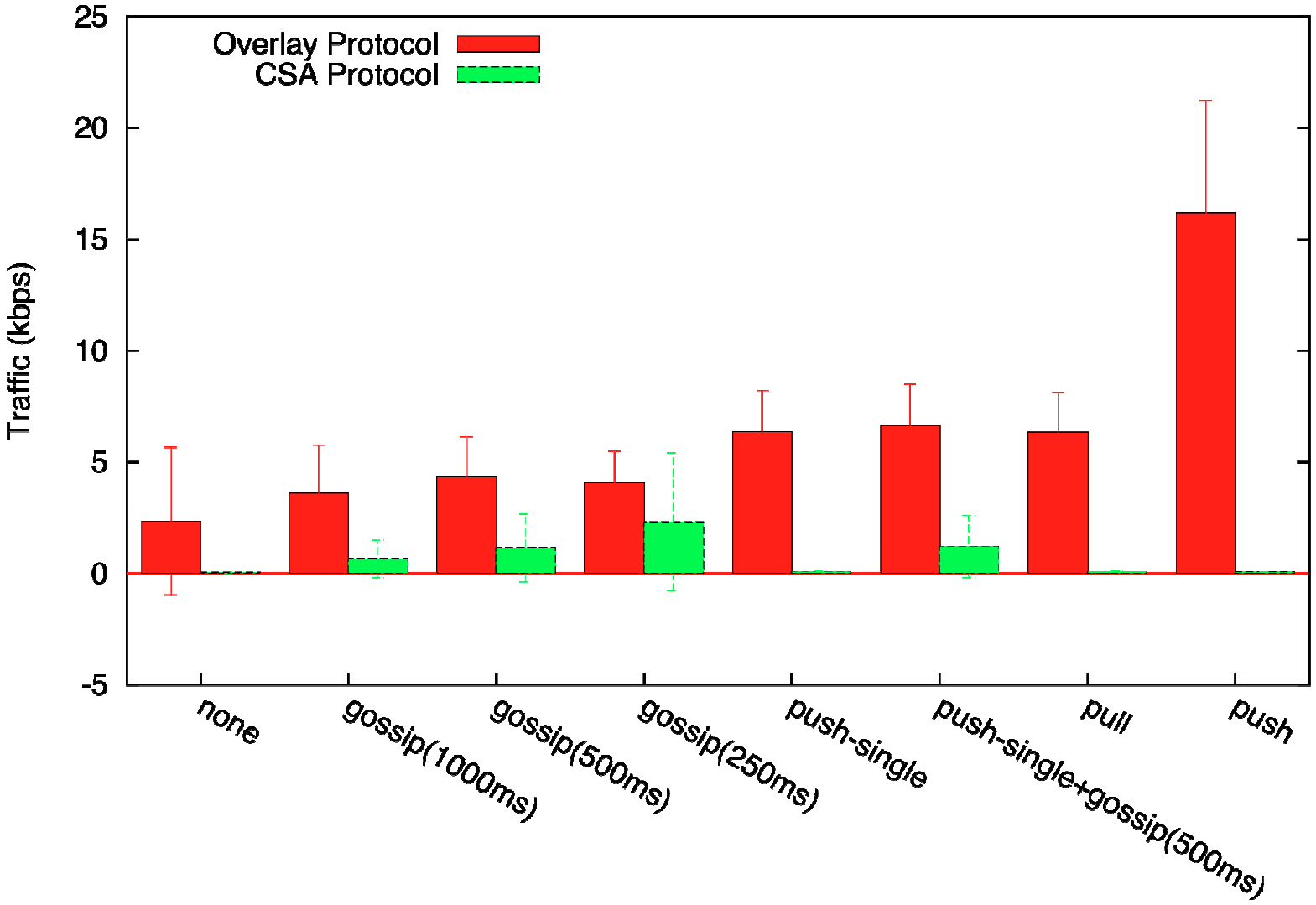
Stability

K=17 (289 substrates), 2592 nodes



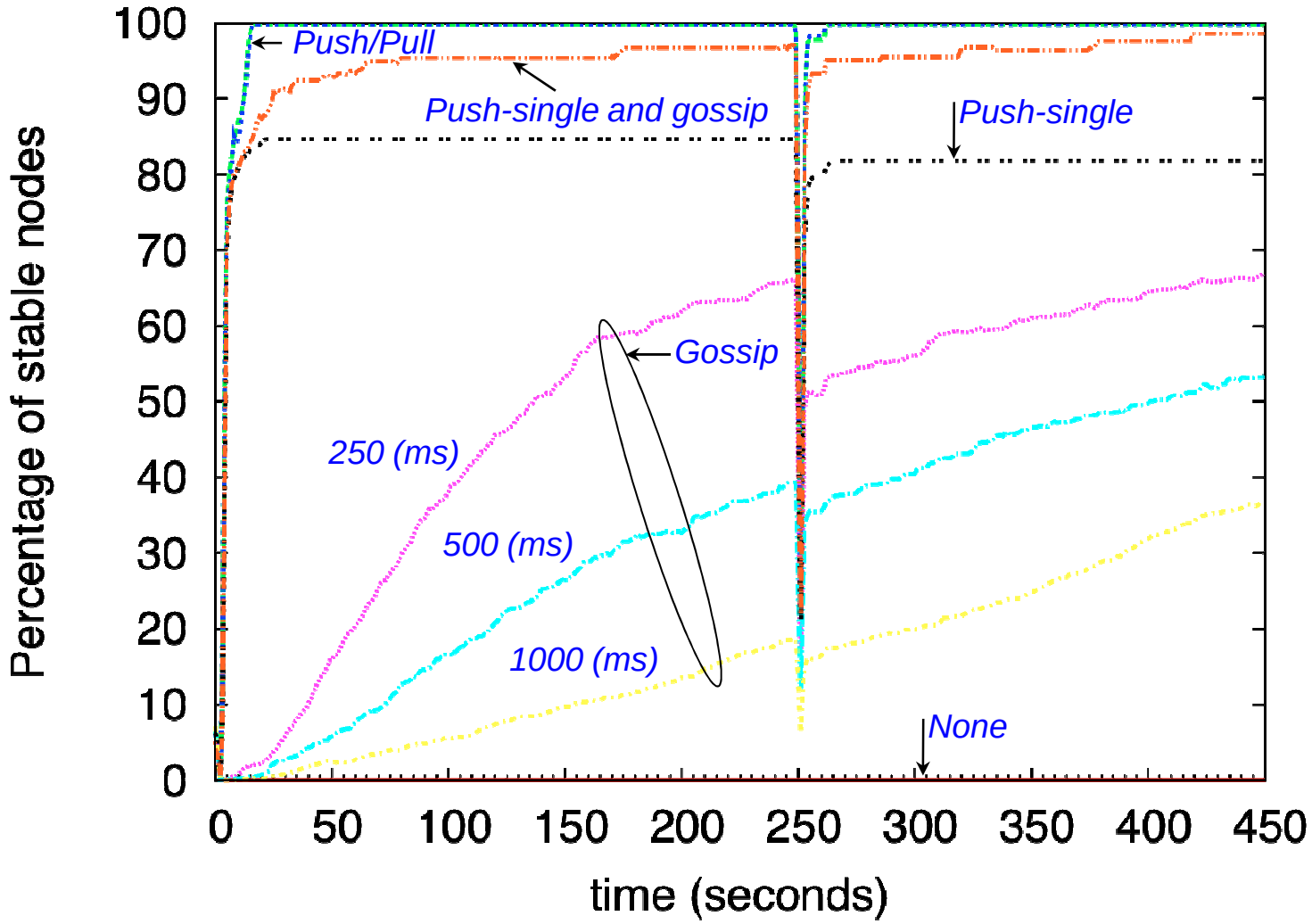
Protocol overhead

Received Traffic (average per node): K=17 (289 substrates), 2592 nodes



Stability under Churn

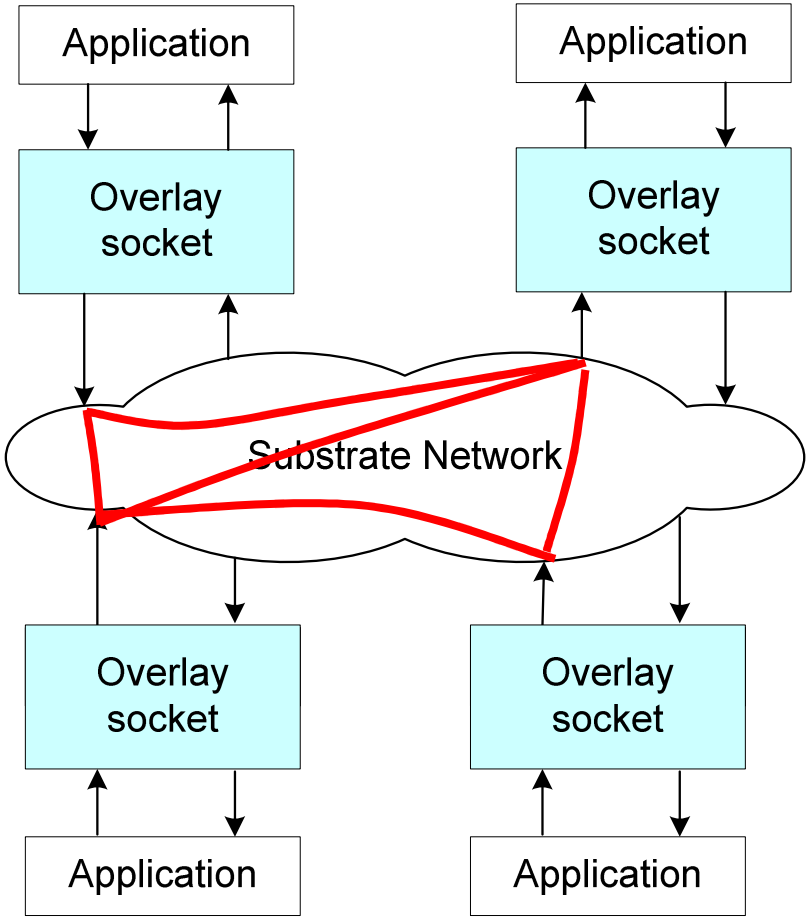
Percentage of stable nodes: K=8, 648 nodes **and 25% leave at t=250**



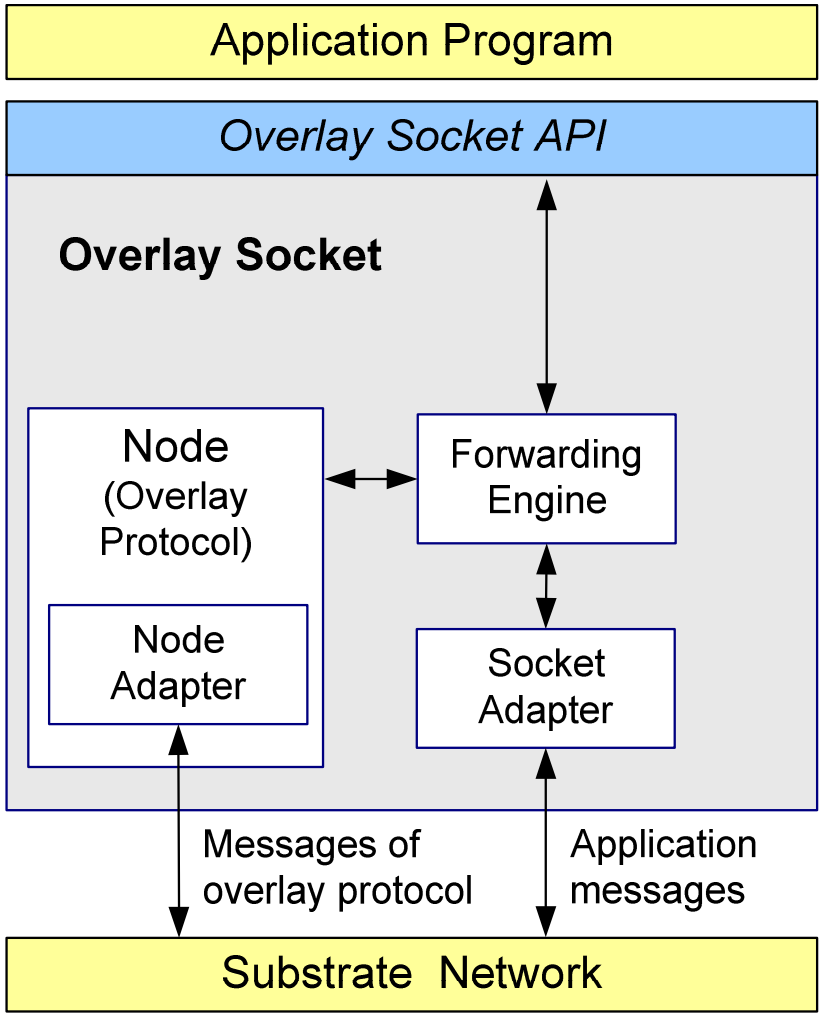
Summary

- Support for self-organizing overlay protocols with multiple substrate networks
- Developed methods for exchanging address bindings
  - Cross-Substrate Advertisement
- Experimental evaluation:
  - Effective in achieving connectivity even with large number of substrates
  - Trade-off of overhead vs. convergence
- CSA mechanisms crucial for self-organizing multi-substrate overlay networks

Overlay Sockets

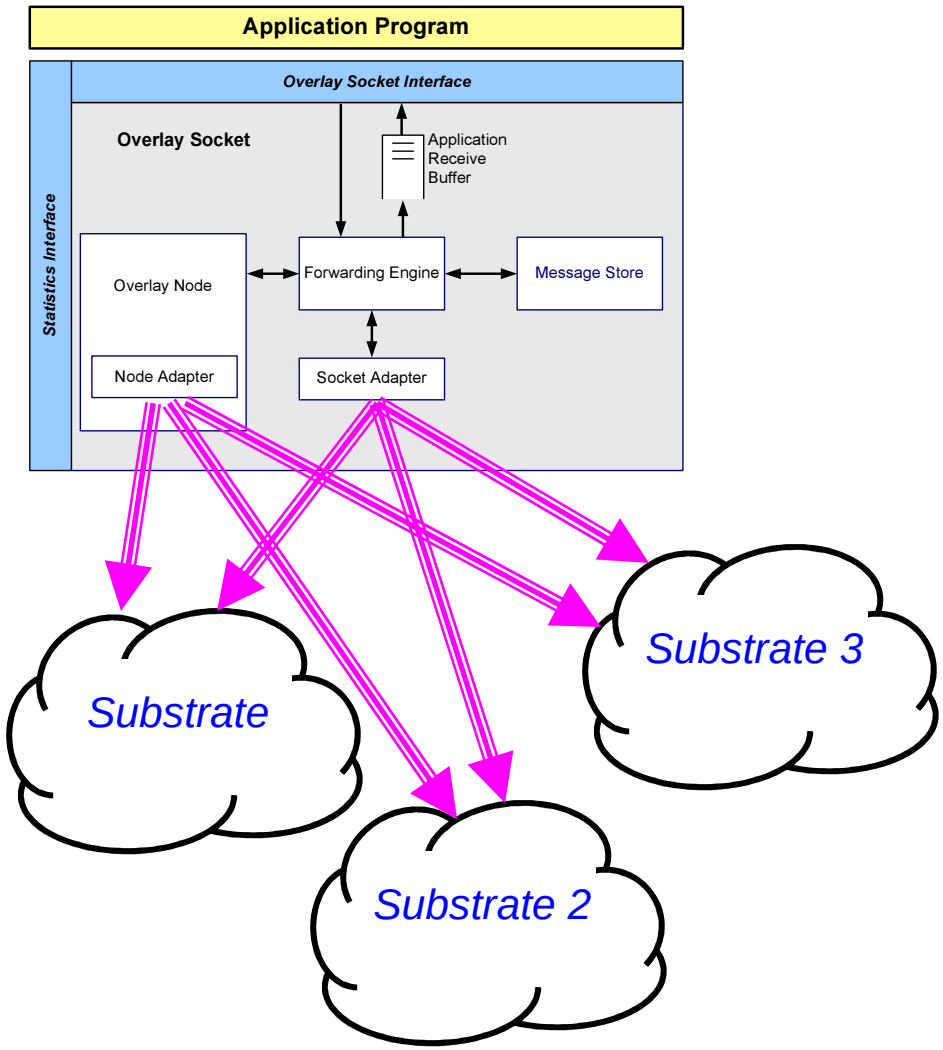


(b) Overlay Network
(Collection of overlay sockets)

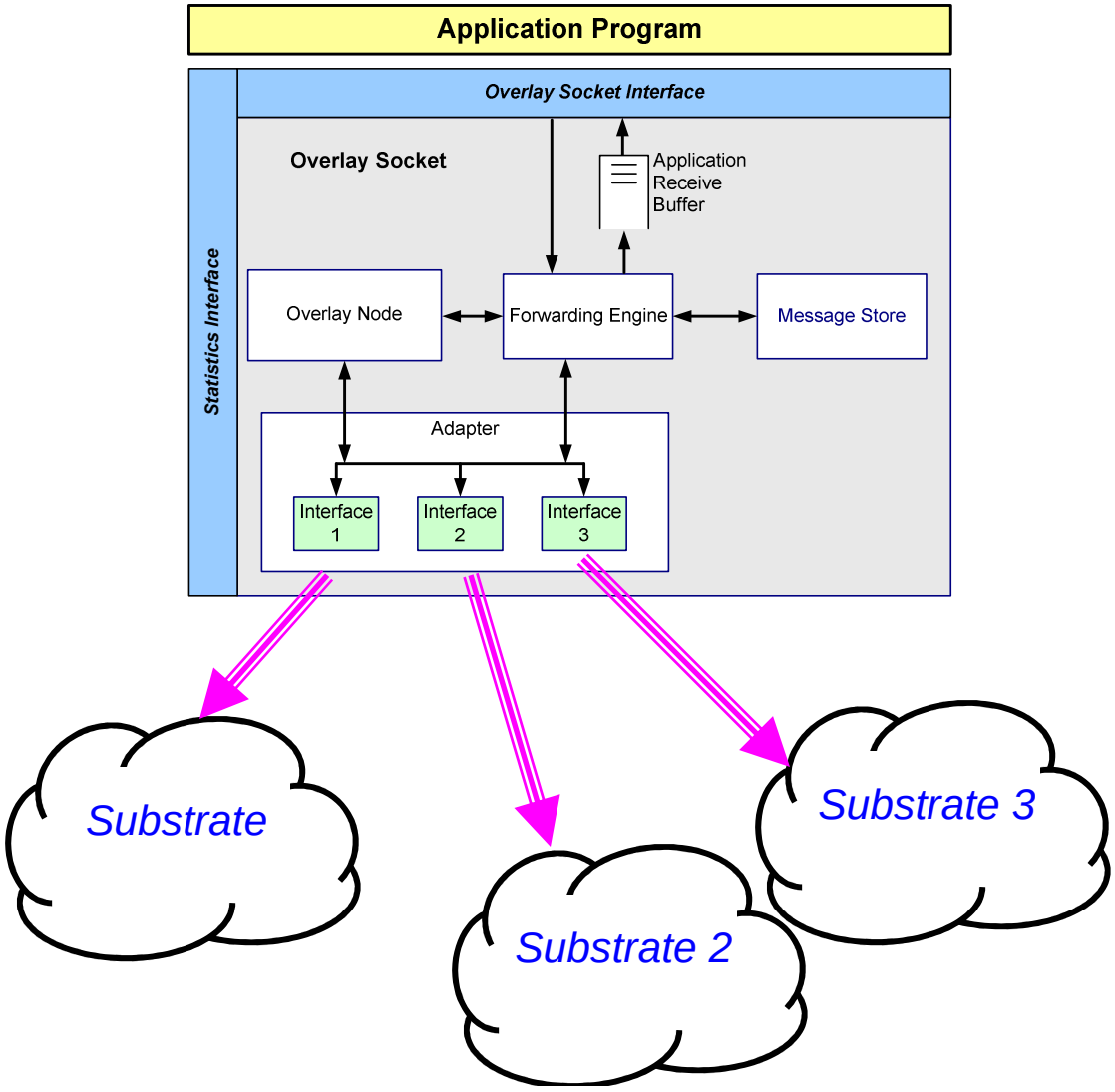


(a) Overlay socket

Problem to solve: Multiple substrate networks



Multi-substrate overlay socket



Adapter has one interface for each substrate.