

An Analytical Model of Information Dissemination for a Gossip-based Protocol

Rena Bakhshi, Daniela Gavidia
Wan Fokkink, Maarten van Steen

Department of Computer Science
Vrije Universiteit Amsterdam

Dutch Dependability Day
May 12, 2009

Outline

The Shuffle Protocol

Protocol Analysis

Experimental Validation

Properties Modelling

The Shuffle Protocol

- Large-scale networks
- Information dissemination
- Gossip-based protocol
- Simple push-pull protocol



How the shuffle protocol works?

Each node

- holds list of data items
- communicates with random peer
- exchanges items

The Shuffle Protocol

The algorithm

wait (some t time units)

$B := \text{selectPeer}();$

$s_A := \text{itemsToSend}(c_A);$

$\text{sendTo}(s_A, B);$

$s_B := \text{receive}();$

$c_A := \text{itemsToKeep}(c_A, s_B);$

(a) initiating node (push)

$s_A := \text{receive}();$

$s_B := \text{itemsToSend}(c_B);$

$\text{sendTo}(s_B, A);$

$c_B := \text{itemsToKeep}(c_B, s_A);$

(b) contacted node (pull)

Example

c_A

a

b

c

d

e

The Shuffle Protocol

The algorithm

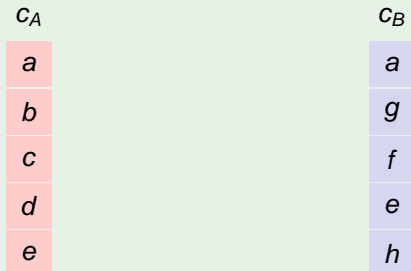
```
wait (some  $t$  time units)
 $B := \text{selectPeer}()$ ;
 $s_A := \text{itemsToSend}(c_A)$ ;
sendTo( $s_A, B$ );
 $s_B := \text{receive}()$ ;
 $c_A := \text{itemsToKeep}(c_A, s_B)$ ;
```

(a) initiating node (push)

```
 $s_A := \text{receive}()$ ;
 $s_B := \text{itemsToSend}(c_B)$ ;
sendTo( $s_B, A$ );
 $c_B := \text{itemsToKeep}(c_B, s_A)$ ;
```

(b) contacted node (pull)

Example



The Shuffle Protocol

The algorithm

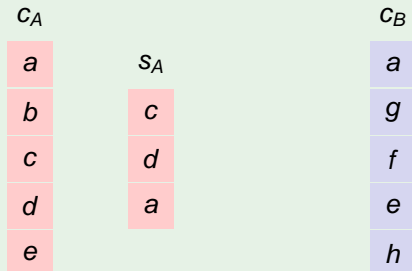
```
wait (some  $t$  time units)
 $B := \text{selectPeer}()$ ;
 $s_A := \text{itemsToSend}(c_A)$ ;
 $\text{sendTo}(s_A, B)$ ;
 $s_B := \text{receive}()$ ;
 $c_A := \text{itemsToKeep}(c_A, s_B)$ ;
```

(a) initiating node (push)

```
 $s_A := \text{receive}()$ ;
 $s_B := \text{itemsToSend}(c_B)$ ;
 $\text{sendTo}(s_B, A)$ ;
 $c_B := \text{itemsToKeep}(c_B, s_A)$ ;
```

(b) contacted node (pull)

Example



The Shuffle Protocol

The algorithm

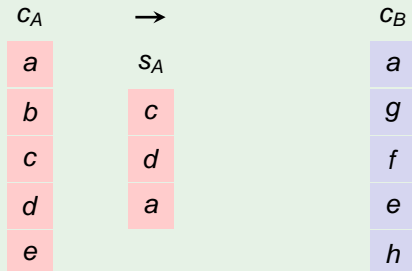
```
wait (some  $t$  time units)
 $B := \text{selectPeer}()$ ;
 $s_A := \text{itemsToSend}(c_A)$ ;
sendTo( $s_A, B$ );
 $s_B := \text{receive}()$ ;
 $c_A := \text{itemsToKeep}(c_A, s_B)$ ;
```

(a) initiating node (push)

```
 $s_A := \text{receive}()$ ;
 $s_B := \text{itemsToSend}(c_B)$ ;
sendTo( $s_B, A$ );
 $c_B := \text{itemsToKeep}(c_B, s_A)$ ;
```

(b) contacted node (pull)

Example



The Shuffle Protocol

The algorithm

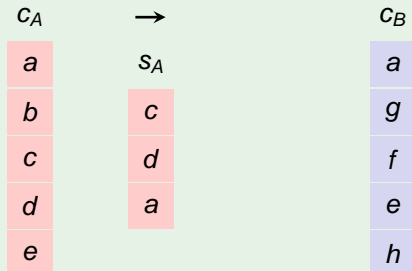
```
wait (some  $t$  time units)
 $B := \text{selectPeer}()$ ;
 $s_A := \text{itemsToSend}(c_A)$ ;
 $\text{sendTo}(s_A, B)$ ;
 $s_B := \text{receive}()$ ;
 $c_A := \text{itemsToKeep}(c_A, s_B)$ ;
```

(a) initiating node (push)

```
 $s_A := \text{receive}()$ ;
 $s_B := \text{itemsToSend}(c_B)$ ;
 $\text{sendTo}(s_B, A)$ ;
 $c_B := \text{itemsToKeep}(c_B, s_A)$ ;
```

(b) contacted node (pull)

Example



The Shuffle Protocol

The algorithm

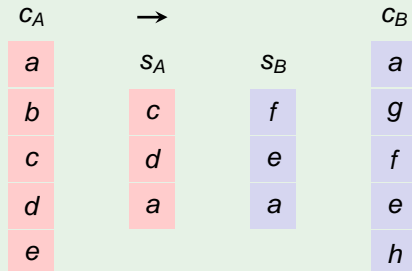
```
wait (some  $t$  time units)
 $B := \text{selectPeer}()$ ;
 $s_A := \text{itemsToSend}(c_A)$ ;
 $\text{sendTo}(s_A, B)$ ;
 $s_B := \text{receive}()$ ;
 $c_A := \text{itemsToKeep}(c_A, s_B)$ ;
```

(a) initiating node (push)

```
 $s_A := \text{receive}()$ ;
 $s_B := \text{itemsToSend}(c_B)$ ;
 $\text{sendTo}(s_B, A)$ ;
 $c_B := \text{itemsToKeep}(c_B, s_A)$ ;
```

(b) contacted node (pull)

Example



The Shuffle Protocol

The algorithm

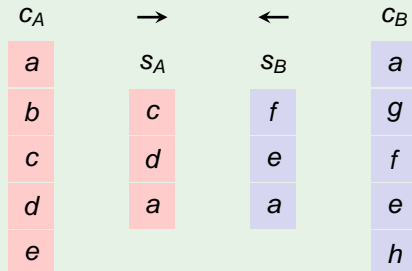
```
wait (some  $t$  time units)
 $B := \text{selectPeer}()$ ;
 $s_A := \text{itemsToSend}(c_A)$ ;
 $\text{sendTo}(s_A, B)$ ;
 $s_B := \text{receive}()$ ;
 $c_A := \text{itemsToKeep}(c_A, s_B)$ ;
```

(a) initiating node (push)

```
 $s_A := \text{receive}()$ ;
 $s_B := \text{itemsToSend}(c_B)$ ;
 $\text{sendTo}(s_B, A)$ ;
 $c_B := \text{itemsToKeep}(c_B, s_A)$ ;
```

(b) contacted node (pull)

Example



The Shuffle Protocol

The algorithm

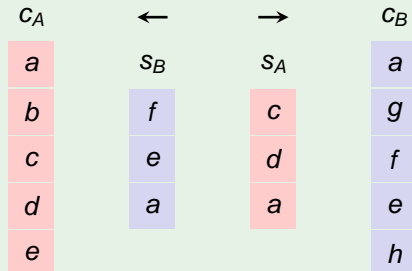
```
wait (some  $t$  time units)
 $B := \text{selectPeer}()$ ;
 $s_A := \text{itemsToSend}(c_A)$ ;
 $\text{sendTo}(s_A, B)$ ;
 $s_B := \text{receive}()$ ;
 $c_A := \text{itemsToKeep}(c_A, s_B)$ ;
```

(a) initiating node (push)

```
 $s_A := \text{receive}()$ ;
 $s_B := \text{itemsToSend}(c_B)$ ;
 $\text{sendTo}(s_B, A)$ ;
 $c_B := \text{itemsToKeep}(c_B, s_A)$ ;
```

(b) contacted node (pull)

Example



The Shuffle Protocol

The algorithm

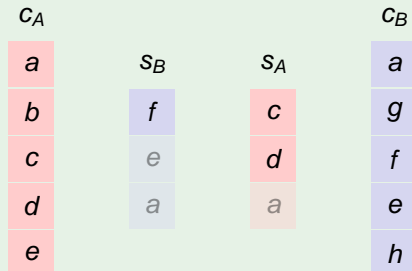
```
wait (some  $t$  time units)
 $B := \text{selectPeer}()$ ;
 $s_A := \text{itemsToSend}(c_A)$ ;
 $\text{sendTo}(s_A, B)$ ;
 $s_B := \text{receive}()$ ;
 $c_A := \text{itemsToKeep}(c_A, s_B)$ ;
```

(a) initiating node (push)

```
 $s_A := \text{receive}()$ ;
 $s_B := \text{itemsToSend}(c_B)$ ;
 $\text{sendTo}(s_B, A)$ ;
 $c_B := \text{itemsToKeep}(c_B, s_A)$ ;
```

(b) contacted node (pull)

Example



The Shuffle Protocol

The algorithm

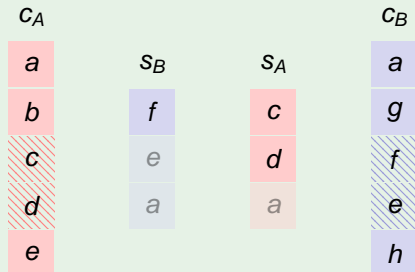
```
wait (some  $t$  time units)
 $B := \text{selectPeer}()$ ;
 $s_A := \text{itemsToSend}(c_A)$ ;
 $\text{sendTo}(s_A, B)$ ;
 $s_B := \text{receive}()$ ;
 $c_A := \text{itemsToKeep}(c_A, s_B)$ ;
```

(a) initiating node (push)

```
 $s_A := \text{receive}()$ ;
 $s_B := \text{itemsToSend}(c_B)$ ;
 $\text{sendTo}(s_B, A)$ ;
 $c_B := \text{itemsToKeep}(c_B, s_A)$ ;
```

(b) contacted node (pull)

Example



The Shuffle Protocol

The algorithm

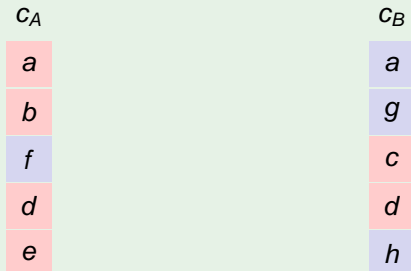
```
wait (some  $t$  time units)
 $B := \text{selectPeer}()$ ;
 $s_A := \text{itemsToSend}(c_A)$ ;
sendTo( $s_A, B$ );
 $s_B := \text{receive}()$ ;
 $c_A := \text{itemsToKeep}(c_A, s_B)$ ;
```

(a) initiating node (push)

```
 $s_A := \text{receive}()$ ;
 $s_B := \text{itemsToSend}(c_B)$ ;
sendTo( $s_B, A$ );
 $c_B := \text{itemsToKeep}(c_B, s_A)$ ;
```

(b) contacted node (pull)

Example



Protocol Analysis

Assumptions

- No failures
- Neighbourhood info provided
- Uniform distribution
- Insertion of items at once

System parameters:

n # different items in the network

c cache size

s exchange buffer size

Uniform Distribution:

- Shuffle is atomic operation
- No loss of items during exchange
- Path between any two nodes

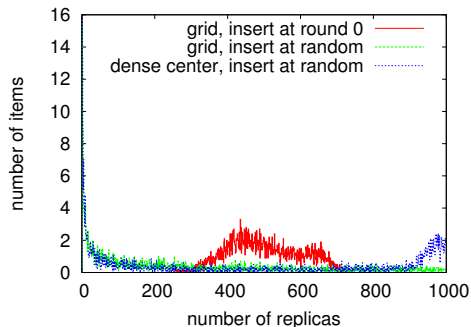
Experiments with different item insertion strategy

Assumption

Setup

- 2500 nodes, 500 items
- 2 topologies
- $c = 100$, $s = 50$

Round 100

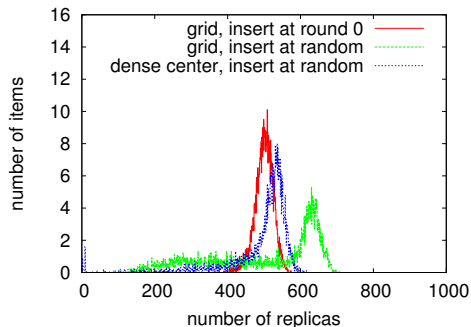


Assumption

Setup

- 2500 nodes, 500 items
- 2 topologies
- $c = 100$, $s = 50$

Round 200

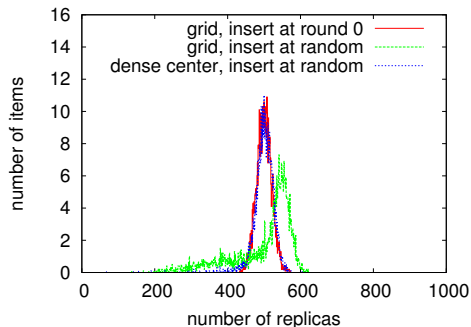


Assumption

Setup

- 2500 nodes, 500 items
- 2 topologies
- $c = 100$, $s = 50$

Round 250

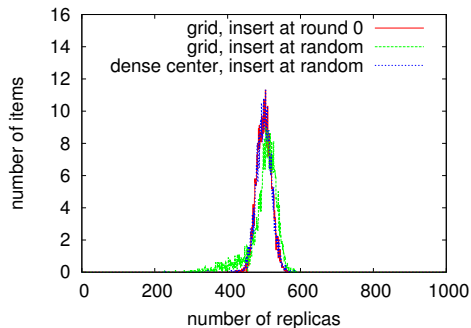


Assumption

Setup

- 2500 nodes, 500 items
- 2 topologies
- $c = 100$, $s = 50$

Round 300

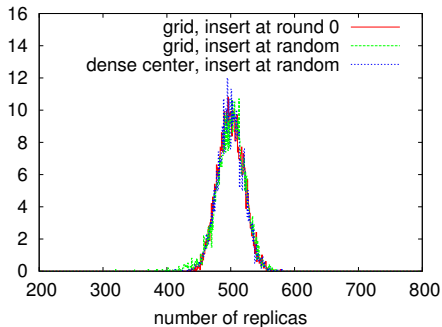


Assumption

Setup

- 2500 nodes, 500 items
- 2 topologies
- $c = 100$, $s = 50$

Round 350

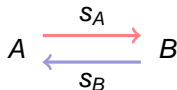


Protocol Analysis

1. The spread of a generic item

- State of a node:
 - 0 item d is not in cache
 - 1 node possesses item d

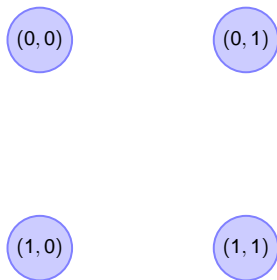
2. Pairwise node interaction



- Pairwise state update
 - $10 \rightarrow 01$
- State transition probabilities

3. Modelling protocol properties

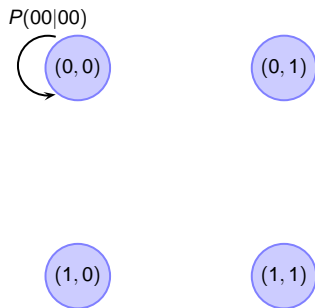
State Transitions



Transition probability matrix

Out-	In-state			
	00	01	10	11
00	1	0	0	0
01	0	$P(01 01)$	$P(01 10)$	$P(01 11)$
10	0	$P(10 01)$	$P(10 10)$	$P(10 11)$
11	0	$P(11 01)$	$P(11 10)$	$P(11 11)$

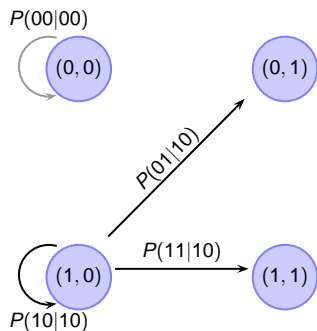
State Transitions



Transition probability matrix

Out-	In-state			
	00	01	10	11
00	1	0	0	0
01	0	$P(01 01)$	$P(01 10)$	$P(01 11)$
10	0	$P(10 01)$	$P(10 10)$	$P(10 11)$
11	0	$P(11 01)$	$P(11 10)$	$P(11 11)$

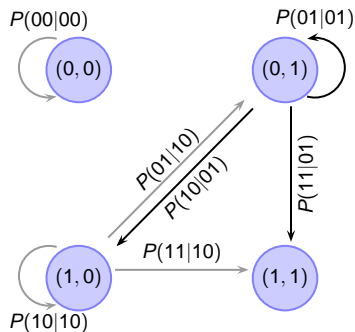
State Transitions



Transition probability matrix

Out-	In-state			
	00	01	10	11
00	1	0	0	0
01	0	$P(01 01)$	$P(01 10)$	$P(01 11)$
10	0	$P(10 01)$	$P(10 10)$	$P(10 11)$
11	0	$P(11 01)$	$P(11 10)$	$P(11 11)$

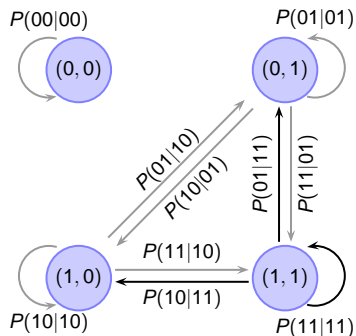
State Transitions



Transition probability matrix

Out-	In-state			
	00	01	10	11
00	1	0	0	0
01	0	$P(01 01)$	$P(01 10)$	$P(01 11)$
10	0	$P(10 01)$	$P(10 10)$	$P(10 11)$
11	0	$P(11 01)$	$P(11 10)$	$P(11 11)$

State Transitions



Transition probability matrix

Out-	In-state			
	00	01	10	11
00	1	0	0	0
01	0	$P(01 01)$	$P(01 10)$	$P(01 11)$
10	0	$P(10 01)$	$P(10 10)$	$P(10 11)$
11	0	$P(11 01)$	$P(11 10)$	$P(11 11)$

Transition Probabilities

- Due to symmetry, $P(a_2 b_2 | a_1 b_1) = P(b_2 a_2 | b_1 a_1)$;
- Thus, 5 transition probabilities to calculate;
- Depend on P_{select} and P_{drop} .

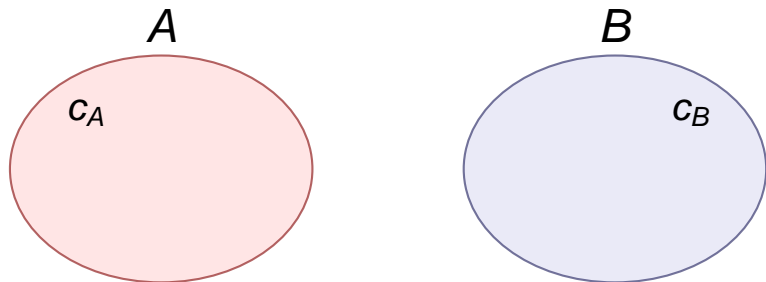
Select/drop an item

- P_{select} : a node selects an item during exchange
- P_{drop} : a node drops (selected) item after exchange

Example

- $P(10|10) = 1 - P_{select}$
- $P(01|10) = P_{select} \cdot P_{drop}$
- $P_{select} = \frac{S}{C}$
- $P_{drop} = ?$

Protocol Analysis



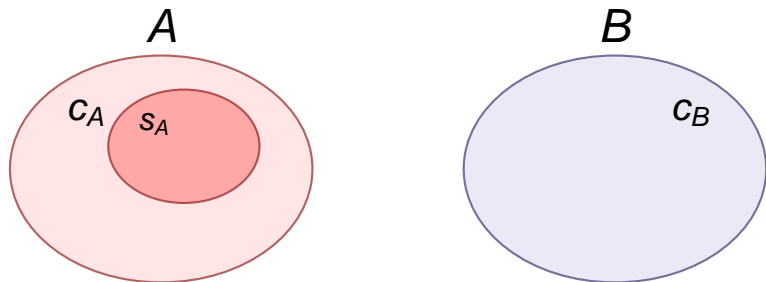
K items sent by A that are in B ($s_A \setminus c_B$);

\hat{S} items in sent by A , also sent by B ($s_A \setminus s_B$);

$s_A \setminus K$ new items that have to be kept by B ;

$s_B \setminus \hat{S}$ items in B that could be dropped to make space

Protocol Analysis



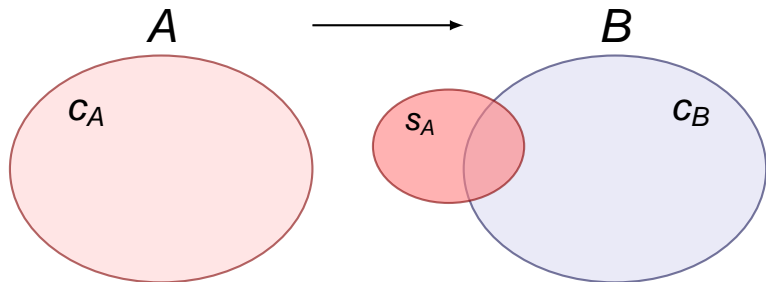
K items sent by A that are in B ($s_A \setminus c_B$);

\hat{S} items in sent by A, also sent by B ($s_A \setminus s_B$);

$s_A \setminus K$ new items that have to be kept by B;

$s_B \setminus \hat{S}$ items in B that could be dropped to make space

Protocol Analysis



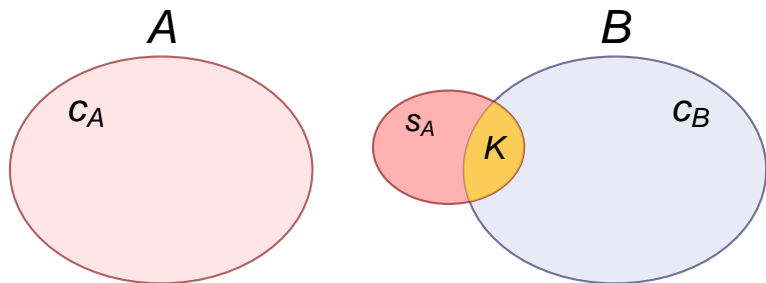
K items sent by A that are in B ($s_A \setminus c_B$);

\hat{S} items in sent by A, also sent by B ($s_A \setminus s_B$);

$s_A \setminus K$ new items that have to be kept by B;

$s_B \setminus \hat{S}$ items in B that could be dropped to make space

Protocol Analysis



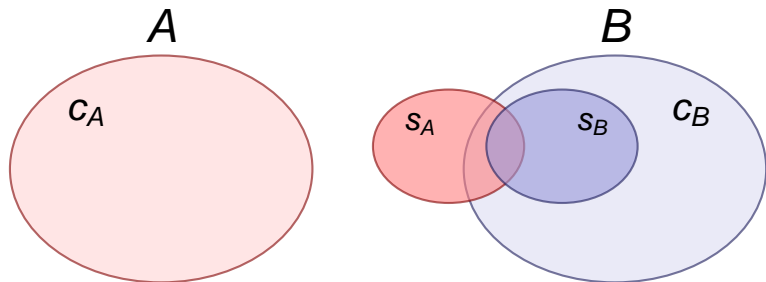
K items sent by A that are in B ($s_A \setminus c_B$);

\hat{S} items in sent by A, also sent by B ($s_A \setminus s_B$);

$s_A \setminus K$ new items that have to be kept by B;

$s_B \setminus \hat{S}$ items in B that could be dropped to make space

Protocol Analysis



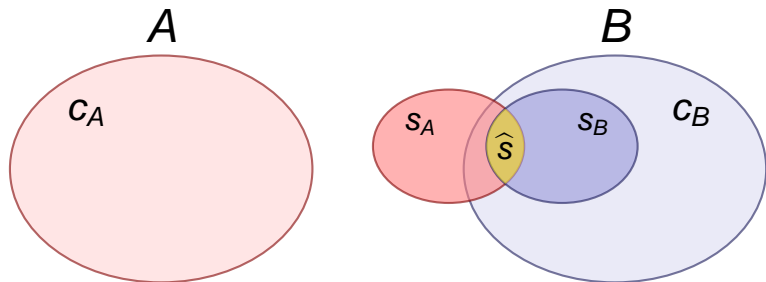
K items sent by A that are in B ($s_A \setminus c_B$);

\hat{S} items in sent by A, also sent by B ($s_A \setminus s_B$);

$s_A \setminus K$ new items that have to be kept by B;

$s_B \setminus \hat{S}$ items in B that could be dropped to make space

Protocol Analysis



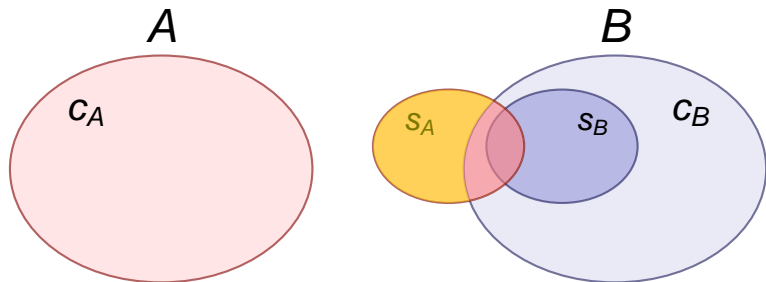
K items sent by A that are in B ($s_A \setminus c_B$);

\hat{S} items in sent by A, also sent by B ($s_A \setminus s_B$);

$s_A \setminus K$ new items that have to be kept by B;

$s_B \setminus \hat{S}$ items in B that could be dropped to make space

Protocol Analysis



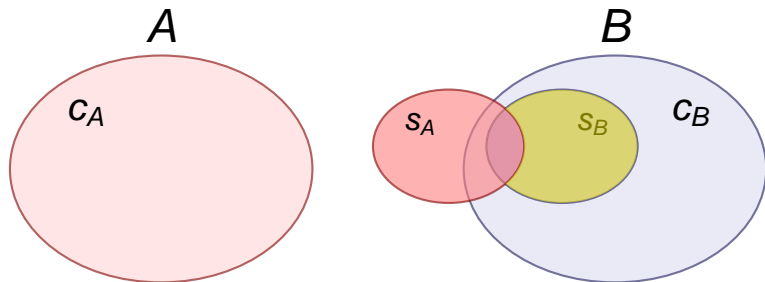
K items sent by A that are in B ($s_A \setminus c_B$);

\hat{S} items in sent by A , also sent by B ($s_A \setminus s_B$);

$s_A \setminus K$ new items that have to be kept by B ;

$s_B \setminus \hat{S}$ items in B that could be dropped to make space

Protocol Analysis



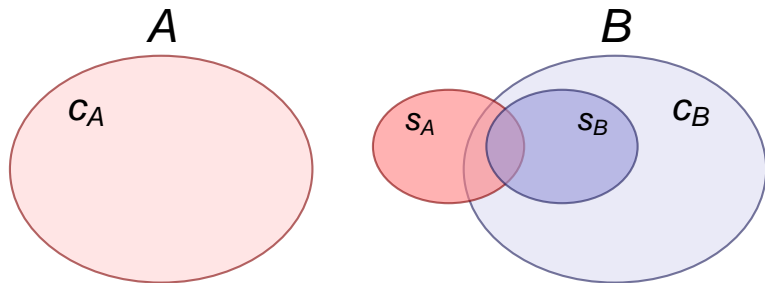
K items sent by A that are in B ($s_A \setminus c_B$);

\hat{S} items in sent by A , also sent by B ($s_A \setminus s_B$);

$s_A \setminus K$ new items that have to be kept by B ;

$s_B \setminus \hat{S}$ items in B that could be dropped to make space

Protocol Analysis



K items sent by A that are in B ($s_A \setminus c_B$);

\hat{S} items in sent by A, also sent by B ($s_A \setminus s_B$);

$s_A \setminus K$ new items that have to be kept by B;

$s_B \setminus \hat{S}$ items in B that could be dropped to make space

$$P_{drop} = \frac{s - k}{s - \hat{S}}$$

Protocol Analysis

- $P_{drop} = \frac{s - k}{s - \hat{s}}$
- But, we don't know values of k and \hat{s} .
- Estimated values using uniform distribution:

$$k \text{ is } s \cdot \frac{c}{n}$$

$$\hat{s} \text{ is } k \cdot \frac{S}{c}$$

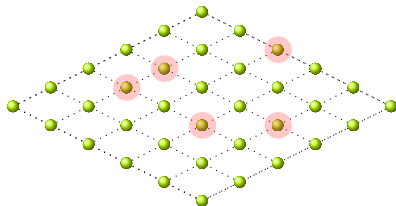
Thus, estimated P_{drop} is $\frac{n - c}{n - s}$.

Protocol Properties

Generic item d inserted in a network of gossiping nodes.

Replication

How many copies of item d there are at round r ?

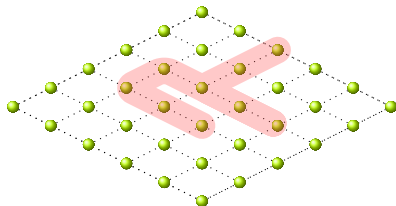


Protocol Properties

Generic item d inserted in a network of gossiping nodes.

Coverage

How many nodes have seen item d after r rounds?



Experimental validation

2500 nodes in a grid, 100 simulations each

Experiments with the protocol

- For each node: $c = 100$
- Items in the network: $n = 500$
- Start-up: 1000 rounds
- At round 1000: item d is inserted and tracked.

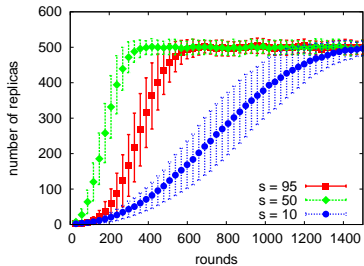
Experiments with the model

- State of a node: $0|1$
- System parameters: $n = 500, c = 100$
- State of one random node set to 1.

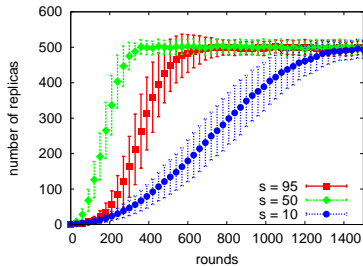
Two properties: replication and coverage

Replication

Protocol

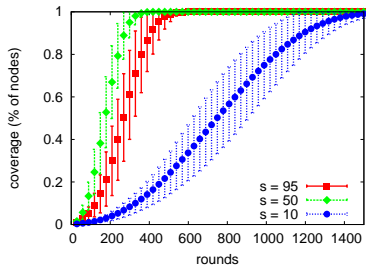


Model

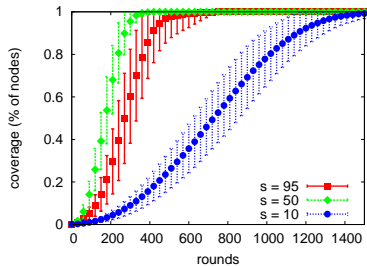


Coverage

Protocol



Model



Properties modelling

Assumptions:

- Full connectivity
- Each node initiates a gossip once per round

Let x – fraction of nodes possessing item d .

Replication

$$\frac{dx}{dt} = [P(11|10) + P(11|01)] \cdot (1 - x) \cdot x - [P(10|11) + P(01|11)] \cdot x \cdot x$$

Properties modelling

Assumptions:

- Full connectivity
- Each node initiates a gossip once per round

Let x – fraction of nodes possessing item d .

Replication

$$\frac{dx}{dt} = \underbrace{[P(11|10) + P(11|01)] \cdot (1-x) \cdot x}_{\text{duplication of } d} - [P(10|11) + P(01|11)] \cdot x \cdot x$$

Properties modelling

Assumptions:

- Full connectivity
- Each node initiates a gossip once per round

Let x – fraction of nodes possessing item d .

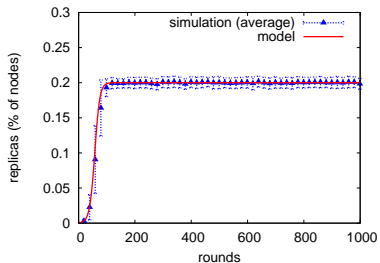
Replication

$$\frac{dx}{dt} = \underbrace{[P(11|10) + P(11|01)] \cdot (1-x) \cdot x}_{\text{duplication of } d} - \underbrace{[P(10|11) + P(01|11)] \cdot x \cdot x}_{\text{dropping } d}$$

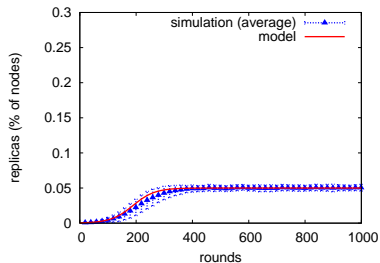
Replication

Results validation:

- $N = 2500$
- $c = 100$
- $s = 50$
- 100 independent runs



500 items



2000 items

Properties modelling

Assumptions:

- Full connectivity
- Each node initiates a gossip once per round
- On average two shuffles – active and passive

Let

- x – fraction of nodes possessing item d
- y – fraction of nodes have seen d

Coverage

$$\frac{dy}{dt} = \frac{1}{2} \cdot (P(1*|01) \cdot P(*1|*1) \cdot (1-y) \cdot x + P(*1|10) \cdot P(1*|1*) \cdot x \cdot (1-y))$$

Properties modelling

Assumptions:

- Full connectivity
- Each node initiates a gossip once per round
- On average two shuffles – active and passive

Let

- x – fraction of nodes possessing item d
- y – fraction of nodes have seen d

Coverage

$$\frac{dy}{dt} = \frac{1}{2} \cdot \underbrace{(P(1*|01) \cdot P(*1|*1))}_{\text{obtain } d} \cdot (1-y) \cdot x + \underbrace{P(*1|10) \cdot P(1*|1*)}_{\text{obtain } d} \cdot x \cdot (1-y)$$

Properties modelling

Assumptions:

- Full connectivity
- Each node initiates a gossip once per round
- On average two shuffles – active and passive

Let

- x – fraction of nodes possessing item d
- y – fraction of nodes have seen d

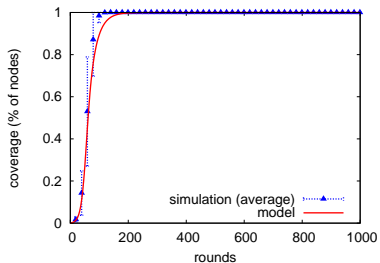
Coverage

$$\frac{dy}{dt} = \frac{1}{2} \cdot \underbrace{P(1*|01)}_{\text{obtain } d} \cdot \underbrace{P(*1|*1)}_{\text{keep } d} \cdot (1-y) \cdot x + \underbrace{P(*1|10)}_{\text{obtain } d} \cdot \underbrace{P(1*|1*)}_{\text{keep } d} \cdot x \cdot (1-y)$$

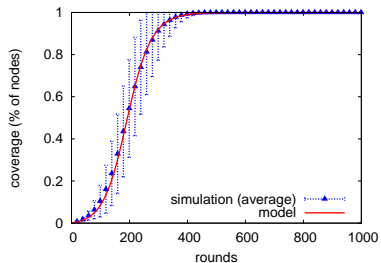
Coverage

Results validation:

- $N = 2500$
- $c = 100$
- $s = 50$
- 100 independent runs



500 items



2000 items

Conclusion

Covered by this talk:

- the shuffle protocol;
- assumptions;
- analysis of spread of generic item;
- P_{select} and P_{drop} ;
- experimental validation;
- modelling of properties.

Other things in the paper:

- precise value of P_{drop} ;
- correction factor of estimated P_{drop} ;
- optimization of exchange buffer;
- coverage model with arbitrary # shuffles per node

And finally!

- “ An Analytical Model of Information Dissemination for a Gossip-based Protocol”
- To appear in Computer Networks, 2009.

Thank you for your attention!

Questions?