Gossip
Maintaining Latest Information
Beyond Channel Bounds

ALFA, June 16th, 2015
Distributed Systems

Finite set of processes
Communicating via reliable FIFO message passing
multiple channels between processes
Remote control light

c_1!b → off

off → c_2!b

c_2!b → c_2!a

c_2!a → on

on → c_1!a

c_1!a → c_1!b

c_1!b

c_1?b → c_1?a

c_1?a

c_3!b

q_1

c_3!a

(c_3!b)

q_2

p

q_2

c_4

r
Obey the latest order

Message Sequence Charts
ITU Standard
Gossip

• Cooperate so that every process maintains latest information about every other process
• When receiving a message, a process needs to identify which is more recent:
  • the information it has,
  • the information transmitted by the sender
How to maintain the latest information?
How to maintain the latest information?
Why maintain the latest information using only finite set of messages?
How to maintain the latest information using only finite set of messages?

need to reuse tags

No natural ordering between the tags
How to maintain the latest information using only finite set of messages?

We use some secondary knowledge.

No natural ordering between the tags.
Is it even possible?

Synchronous communication [Zielonka87]

At least in some cases?

Bounded channels [Mukund et al.03]

Beyond Bounded channels?

How to maintain the latest information using only finite set of messages?
How to maintain the latest information using only finite set of messages?

When is a color not needed any more?

I can reuse a color when I know that the tagged message has been received.

and I know that everyone knows that the tagged message has been received.

colors are not freed in the order they were used.

k-Bounded channels permit finite time-stamping.

requires k colors, necessary, but not sufficient.

Secondary knowledge requires $k^2$ colors, showing a bound, and using a round-robin does not work.

Challenges
How to maintain the latest information using only finite set of messages?

* k-Bounded channels permit finite time-stamping
* Are channel bounds necessary for finite time-stamping?
* Equivalent writes
* Not simply stuttering
* Important writes
* Are existential channel bounds necessary?
How to maintain the latest information using only finite set of messages?

Are existential channel bounds necessary?

Equivalent writes

Important writes
We need some bound: Primary information

- Pending writes
- Equivalent writes
- Primary writes
We need some bound: Primary information

We solve the gossip problem for primary bounded equivalent writes.
How do we maintain the primary?

Keeping primary alone is not enough

Need secondary knowledge

What is secondary knowledge?
Secondary = Primary of Primary
Secondary = Primary of Primary

When can I reuse a color?

When it is not in the secondary

~ \(k^2\) colors

~ size of secondary

Can we maintain secondary?

YES, WE CAN!
Gossip: more precisely

- Message passing automaton (MPA or CFM)

Gossip = (Locs, (Trans_p)_{p \in \text{Procs}})

Run: \( \rho : \text{Events} \to \text{Locs} \)
Known and Latest

**Known**: $\text{Locs} \rightarrow 2^{\text{Procs}}$

**Latest**: $\text{Locs}^2 \rightarrow 2^{\text{Procs}}$

$\text{Known}(l'') = \{p_2, p_3, \ldots, p_6\}$

$\text{Latest}(l, l') = \{p_3, p_5, p_6\}$
Colors and time-stamps

\[ \chi(g) = \min(\mathbb{N} \setminus \chi(\text{Sec}(\downarrow g) \cap \text{Send}(d))) \]

\[ h(g) = (d, \chi(g)) \]

\[ K^1(g) = \{ h(e) \mid e \in \text{Prim}(\downarrow g) \} \]
Locations of Gossip

\[ K^2(g) = (\text{pid}(g), d, c, K^1(g), (K^1(e))_{e \in \text{Prim}(g)}) \]

\[ \ell = (p, d, c, P, (S_\gamma)_{\gamma \in P}) \]
Locations of Gossip

\[ K^2(g) = (\text{pid}(g), d, c, K^1(g), (K^1(e))_{e \in K^1(g)}) \]

\[ \ell = (p, d, c, P, (S_\gamma)_{\gamma \in P}) \]
Known

\[ \ell = (p, d, c, P, (S_\gamma)_{\gamma \in P}) \]

\[ \text{Known}(\ell) = \{p\} \cup \text{pid}(P) \]

\[ \text{pid}(\downarrow g) = \{\text{pid}(g)\} \cup \text{pid}(\text{Prim}(\downarrow g)) \]
Maintaining $K^2$

write event case 1

Equivalent writes: no changes
Maintaining $K^2$

**Write event case 2**

New channel: requires an available color

\[
\ell = (p, d, c, P, (S_\gamma)_{\gamma \in P})
\]
\[
\ell' = (p, d', c', P' = P \cup \{(d', c')\}, (S'_\gamma)_{\gamma \in P'})
\]
Maintaining $K^2$

$\ell = (p, d, c, P, (S_\gamma)_{\gamma \in P})$
$\ell' = (p', d', c', P', (S'_{\gamma})_{\gamma \in P'})$

$f < c$ iff

$(d', c') \in P \land \exists (d'', c'') \in P \setminus P' \ (d'' \neq d' \land W(d'') = W(d'))$
Maintaining $K^2$

**Read event case 1**

\[ \ell = (p, d, c, P, (S_\gamma)_{\gamma \in P}) \]
\[ \ell' = (p', d', c', P', (S'_\gamma)_{\gamma \in P'}) \]

\[ f < e \text{ iff } (d', c') \in P \land \exists (d'', c'') \in P \setminus P' (d'' \neq d' \land W(d'') = W(d')) \]
Maintaining $K^2$

*(read event case 1)*

Latest($\ell, \ell'$) = Known($\ell$)

\[ \ell = (p, d, c, P, (S_\gamma)_{\gamma \in P}) \]
\[ \ell' = (p', d', c', P', (S'_\gamma)_{\gamma \in P'}) \]

\[ \ell'' = (p, \bot, \bot, P'', (S''_\gamma)_{\gamma \in P''}) \]
\[ P'' = (P \setminus (P' \cap d')) \cup \{(d', c')\} \]

\[ f < e \text{ iff } \]
\[ (d', c') \in P \land \exists (d'', c'') \in P \setminus P' \bigg( d'' \neq d' \land \mathcal{W}(d'') = \mathcal{W}(d') \bigg) \]
Maintaining $K^2$

read event case 2

$\ell = (p, d, c, P, (S_\gamma)_{\gamma \in P})$

$\ell' = (p, d', c', P', (S'_\gamma)_{\gamma \in P'})$

$e < f$ implies $(d, c) \in P'$
Maintaining $K^2$

\[ \ell = (p, d, c, P, (S_\gamma)_{\gamma \in P}) \]
\[ \ell' = (p, d', c', P', (S'_\gamma)_{\gamma \in P'}) \]

$e < f$ implies $(d, c) \in P'$

not iff
Maintaining $K^2$

Latest($\ell, \ell''$) = \{p\}

$\ell = (p, d, c, P, (S_\gamma)_{\gamma \in P})$
$\ell' = (p, d', c', P', (S'_\gamma)_{\gamma \in P'})$
$\ell'' = (p, \perp, \perp, P'', (S''_\gamma)_{\gamma \in P''})$
$P'' = (P' \setminus (P' \cap d')) \cup \{(d', c')\}$

e < f implies (d, c) $\in$ P'

not case 1 and (d, c) $\in$ P' implies ...

not iff
Maintaining $K^2$

not (case 1 or case 2) implies $e \parallel f$

$$\text{Prim}(\downarrow g) = (\text{Prim}(\downarrow e) \cap \text{Prim}(\downarrow f)) \cup (\text{Prim}(\downarrow e) \setminus \downarrow f) \cup (\text{Prim}(\downarrow f) \setminus \downarrow e)$$

$$\text{Prim}(\downarrow e) \setminus \bigcup_{e' \in \text{Prim}(\downarrow e) \cap \text{Prim}(\downarrow f)} \text{Prim}(\downarrow e')$$
Maintaining $K^2$

- Read event case 3
- $h$ injective on $\text{Sec}(\downarrow e \cup \downarrow f)$

$\text{not (case 1 or case 2) implies } e \parallel f$

$$\text{Prim}(\downarrow g) = (\text{Prim}(\downarrow e) \cap \text{Prim}(\downarrow f)) \cup (\text{Prim}(\downarrow e) \setminus \downarrow f) \cup (\text{Prim}(\downarrow f) \setminus \downarrow e)$$

$$P''' = (P \cap P') \cup \left(P \setminus \bigcup_{\gamma \in P \cap P'} S_{\gamma}\right) \cup \left(P' \setminus \bigcup_{\gamma \in P \cap P'} S'_{\gamma}\right)$$

$$\text{Prim}(\downarrow e) \setminus \bigcup_{e' \in \text{Prim}(\downarrow e) \cap \text{Prim}(\downarrow f)} \text{Prim}(\downarrow e')$$
Maintaining $K^2$

**Read event case 3**

$h$ injective on $\text{Sec}(\downarrow e \cup \downarrow f)$

$\text{Latest}(\ell, \ell'')$ can be computed

\[
\ell = (p, d, c, P, (S_{\gamma})_{\gamma \in P}) \\
\ell' = (p, d', c', P', (S'_{\gamma})_{\gamma \in P'}) \\
P'' = (P'''' \setminus (P' \cap d')) \cup \{(d', c')\}
\]

not (case 1 or case 2) implies $e \parallel f$

\[
P'''' = (P \cap P') \cup (P \setminus \bigcup_{\gamma \in P \cap P'} S_{\gamma}) \cup (P' \setminus \bigcup_{\gamma \in P \cap P'} S'_{\gamma})
\]
How to maintain the latest information using only finite set of messages?

Is it even possible?

Synchronous communication [Zielonka87]

At least in some cases?

Bounded channels [Mukund et al.03]

Primary bounded