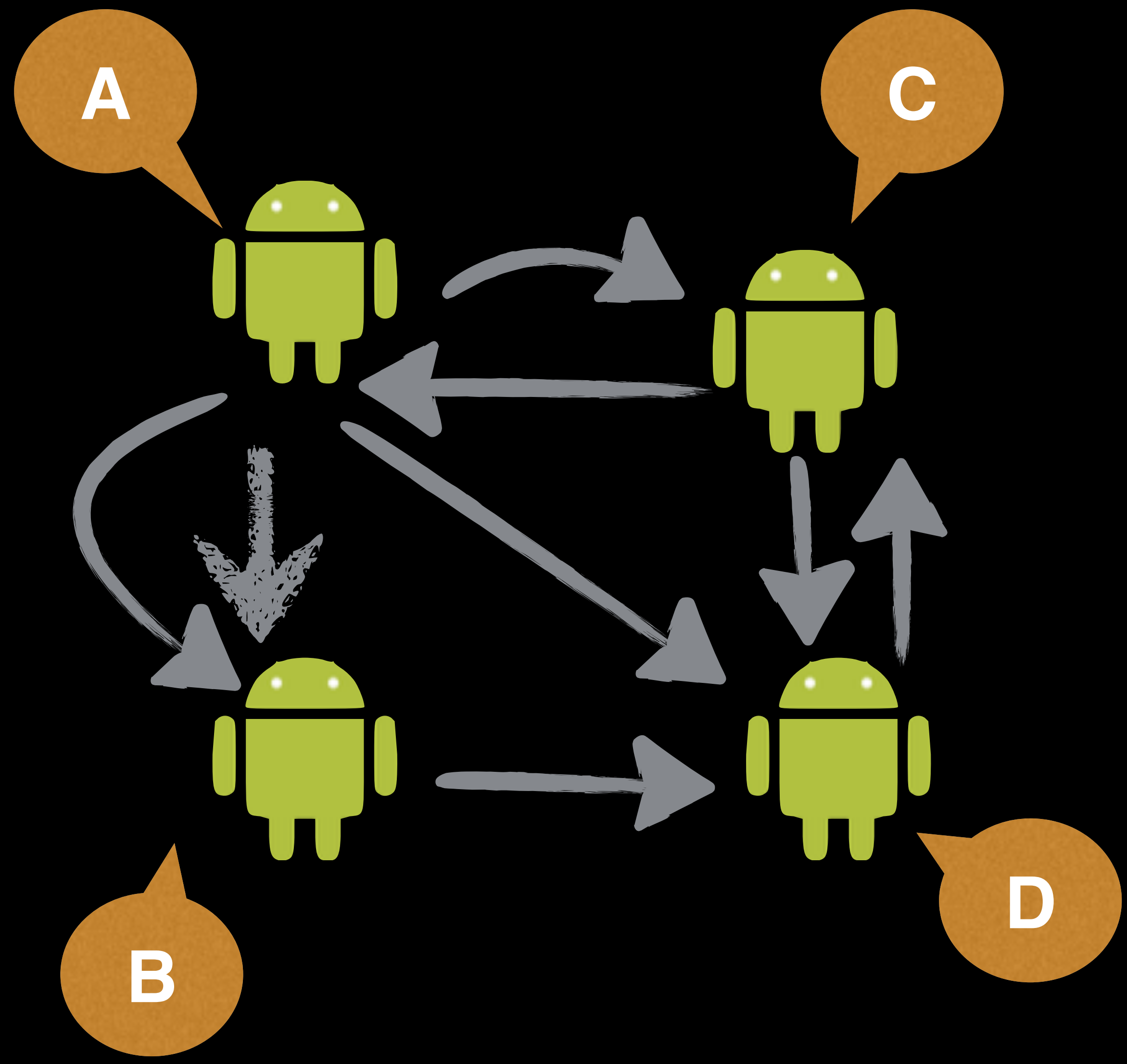
C. Aiswarya, Paul Gastin, K. Narayan Kumar

Gossip
Maintaining Latest Information Beyond Channel Bounds

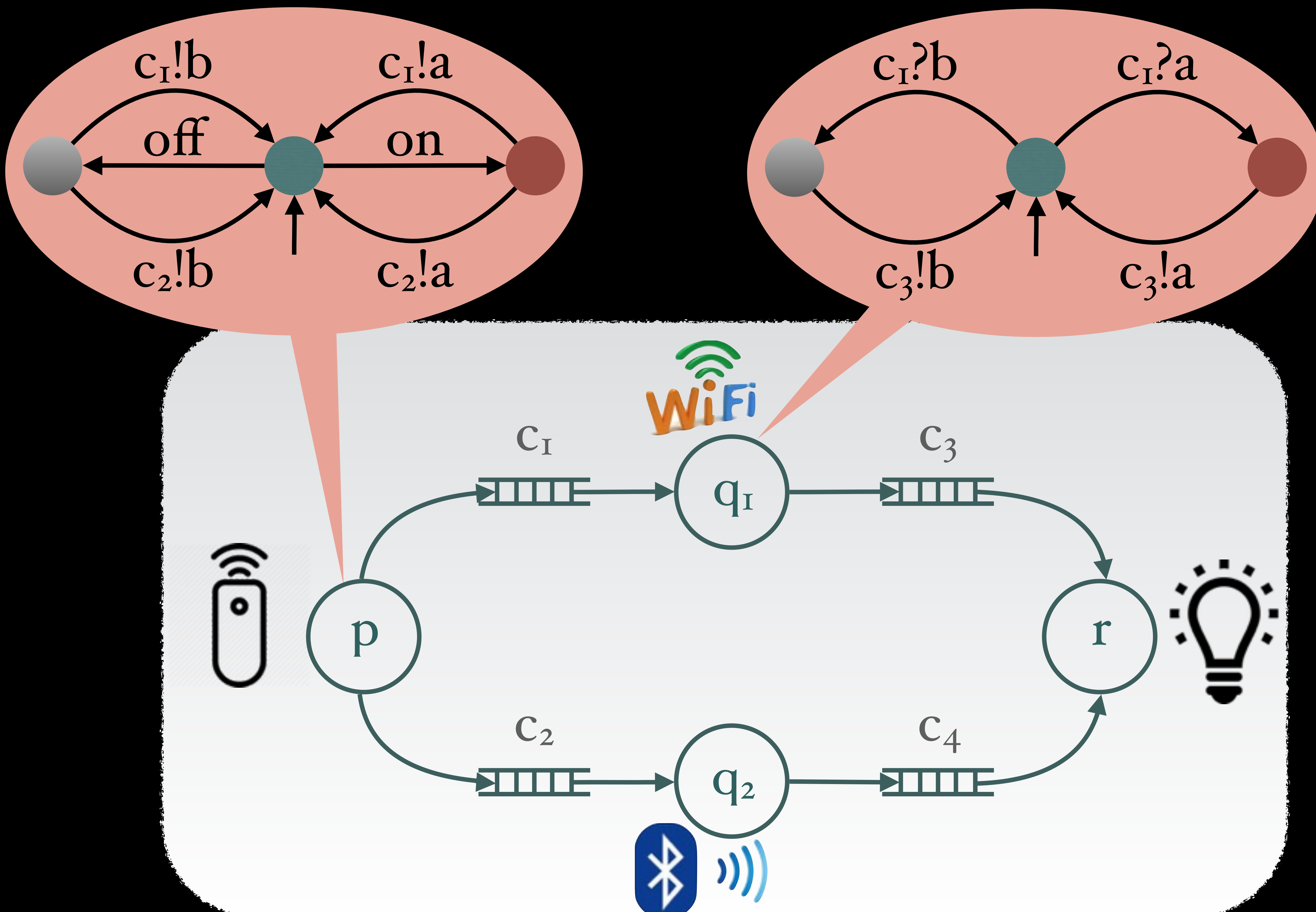
ALFA, June 16th, 2015

Distributed Systems



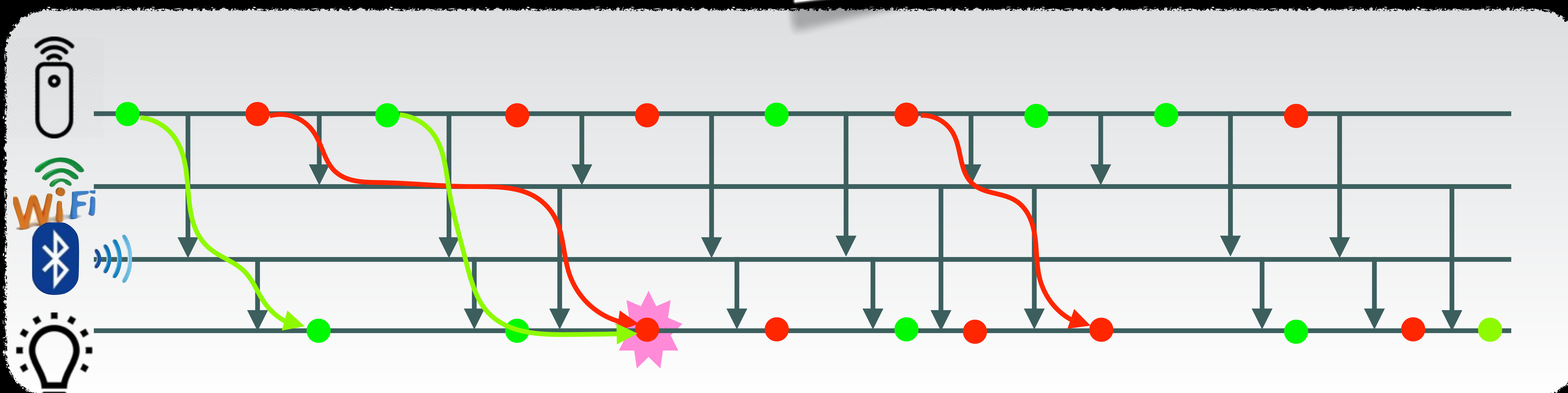
- Finite set of processes
- Communicating via reliable FIFO message passing
- multiple channels between processes

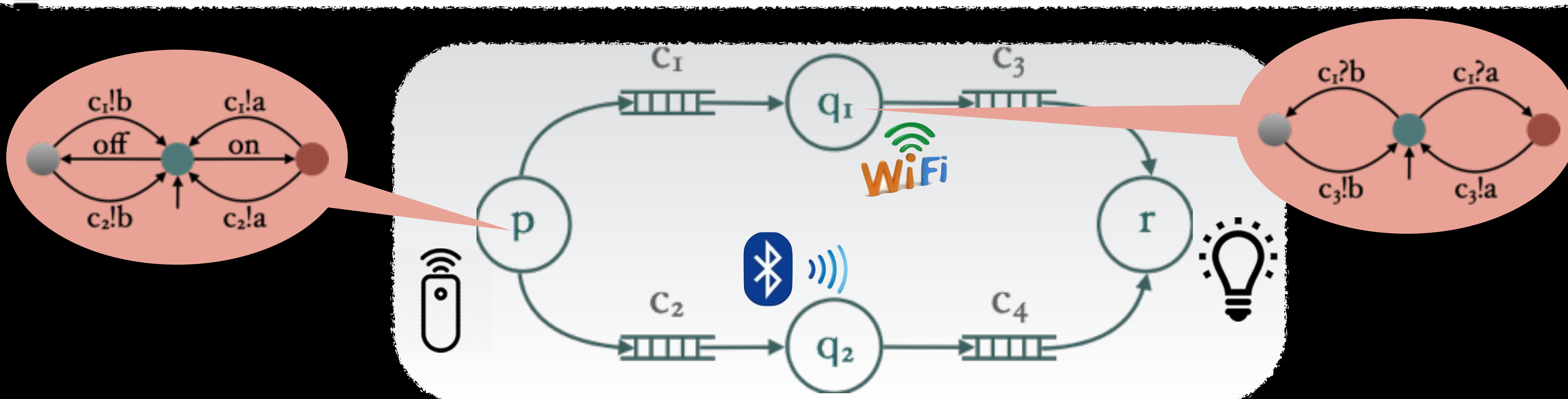
Remote control light



Obey the latest order

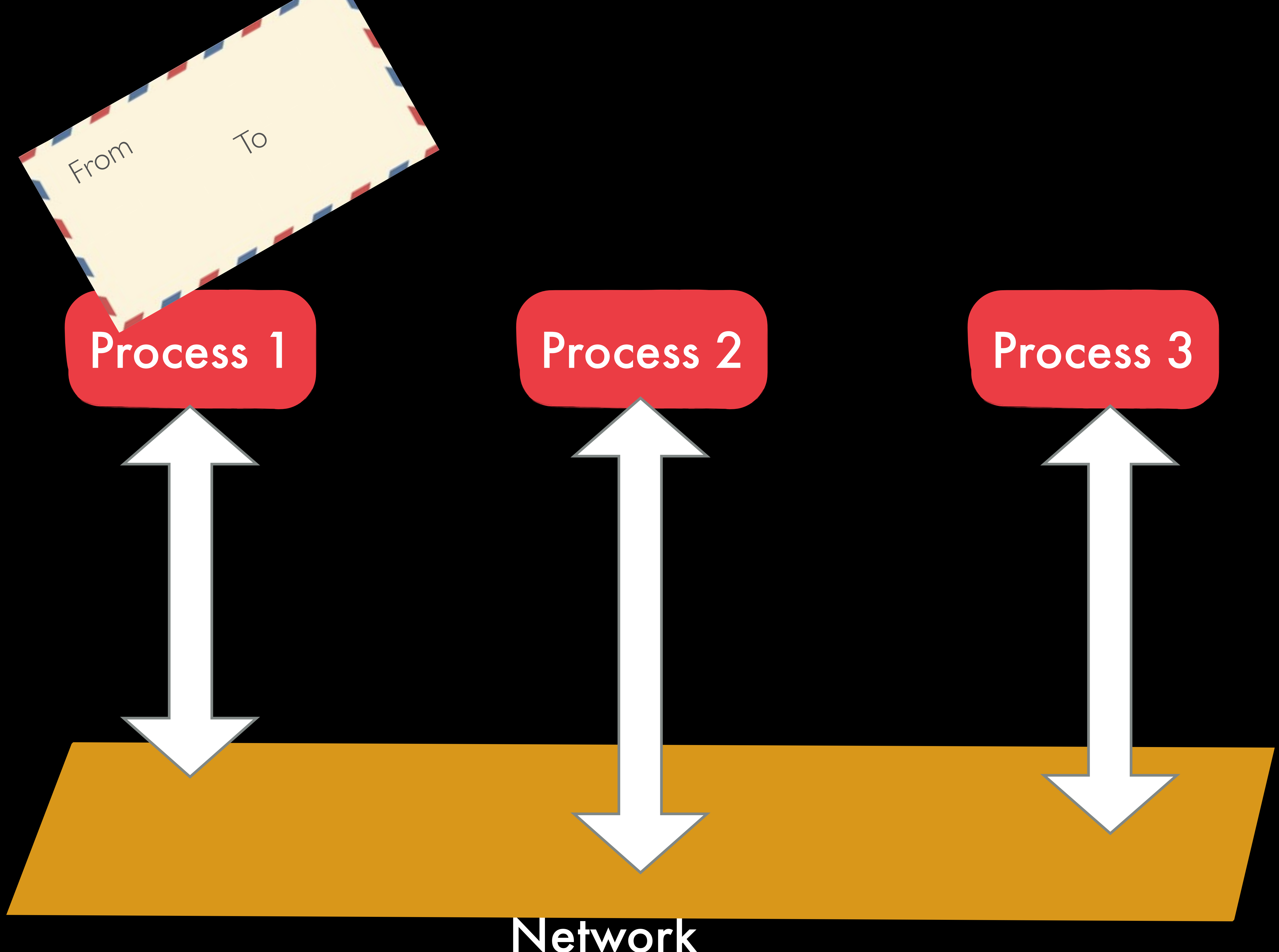
Message Sequence Charts
ITU Standard

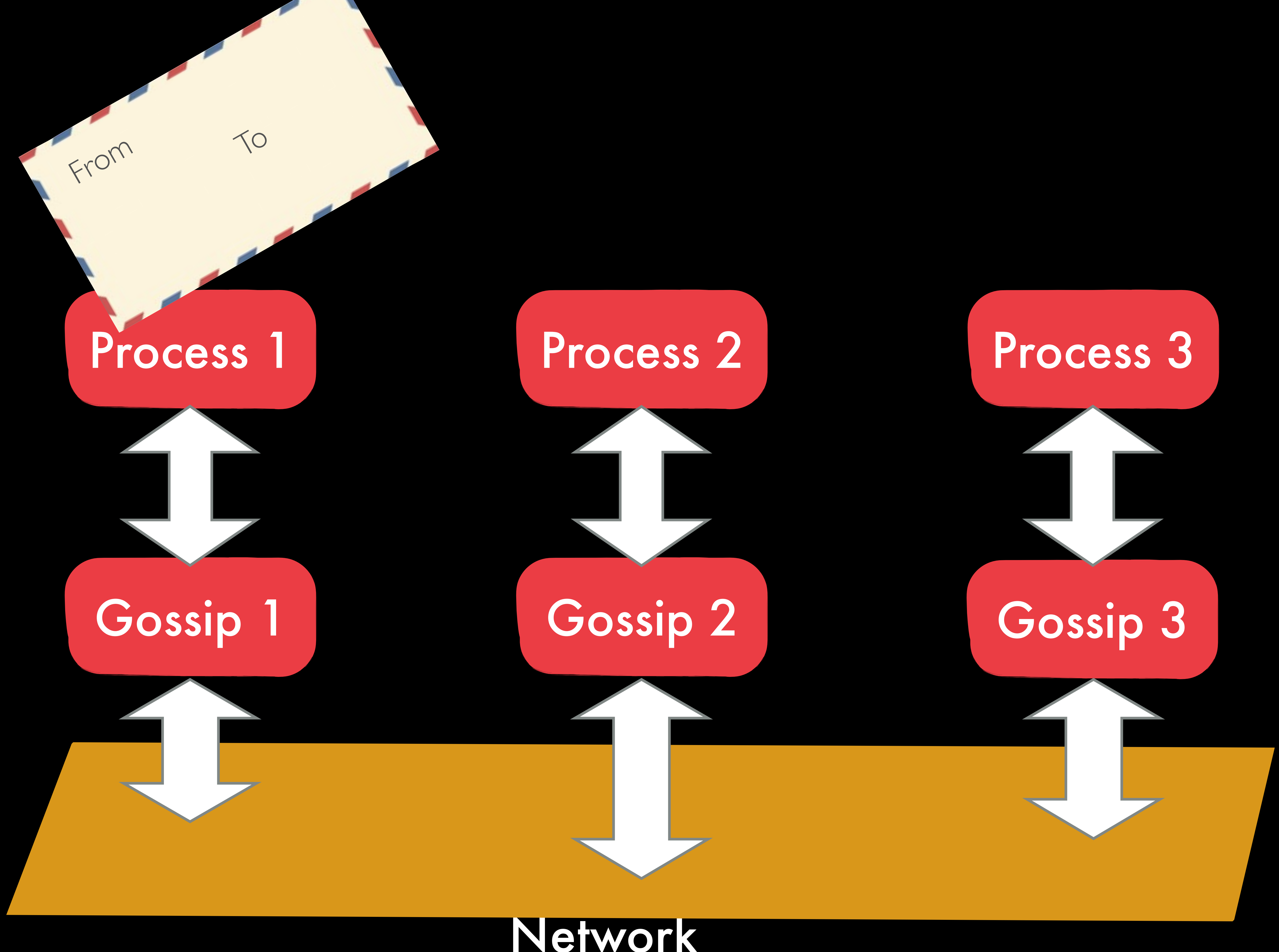


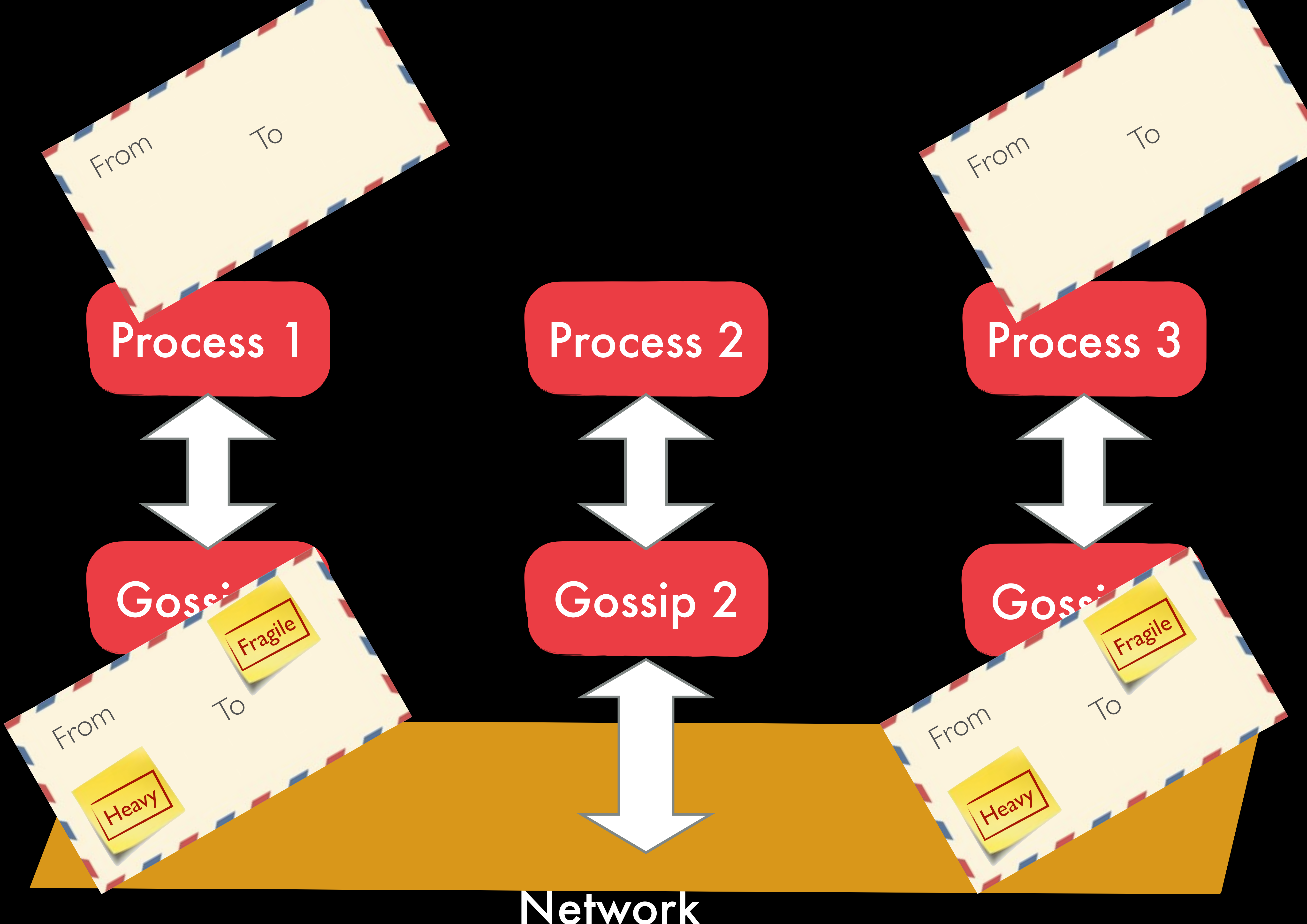


Gossip

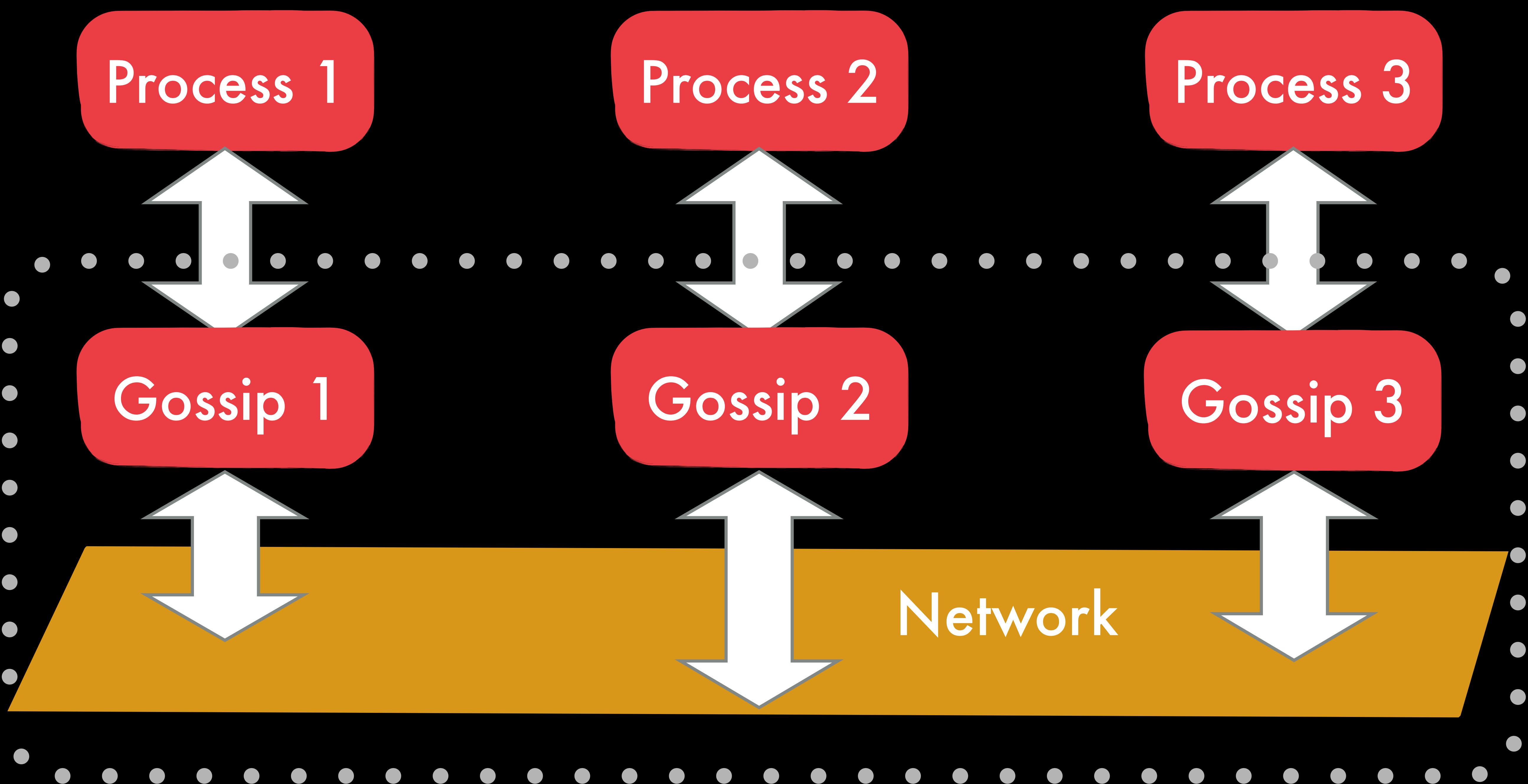
- Cooperate so that every process maintains latest information about every other process
- When receiving a message, a process needs to identify which is more recent:
  - the information it has,
  - the information transmitted by the sender


From
To
Process 1
Process 2
Process 3
Network

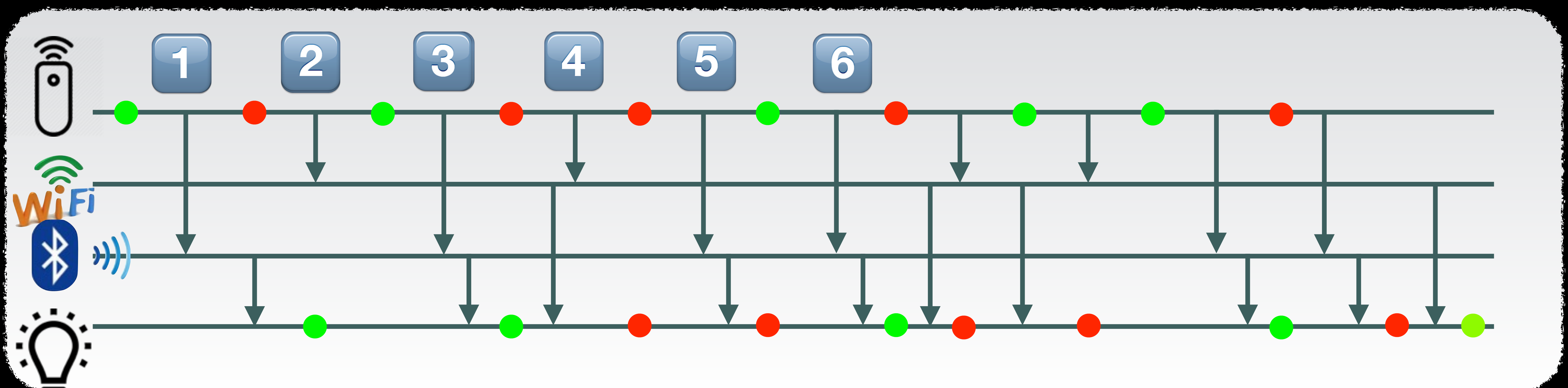

From
To
Process 1
Process 2
Process 3
Gossip 1
Gossip 2
Gossip 3
Network

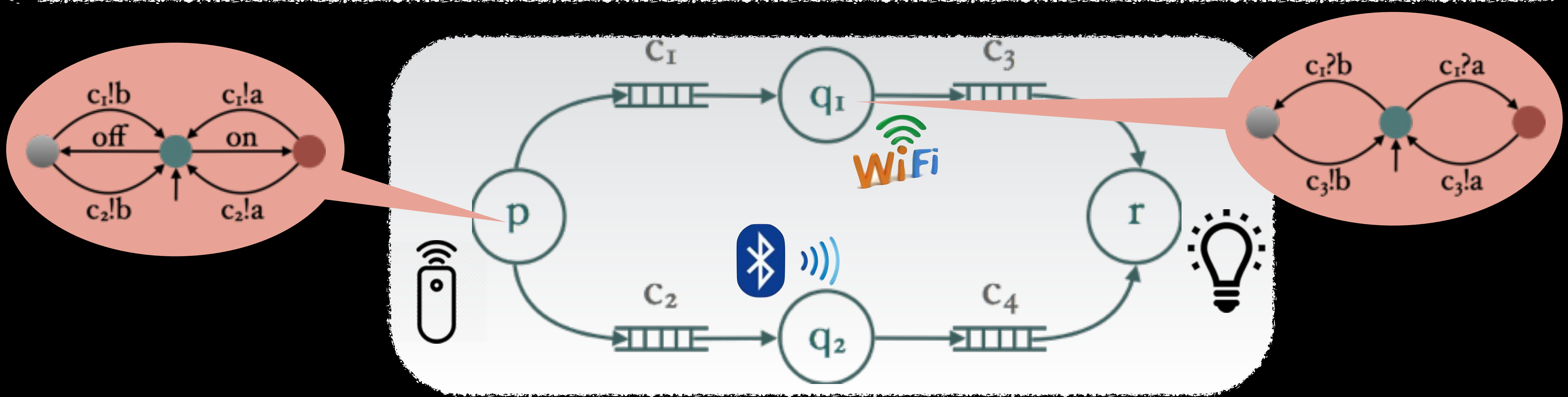


From
To
Process 1
Process 2
Process 3
From
To
Gossip 2
Fragile
From
To
Heavy
Fragile
From
To
Heavy
Network

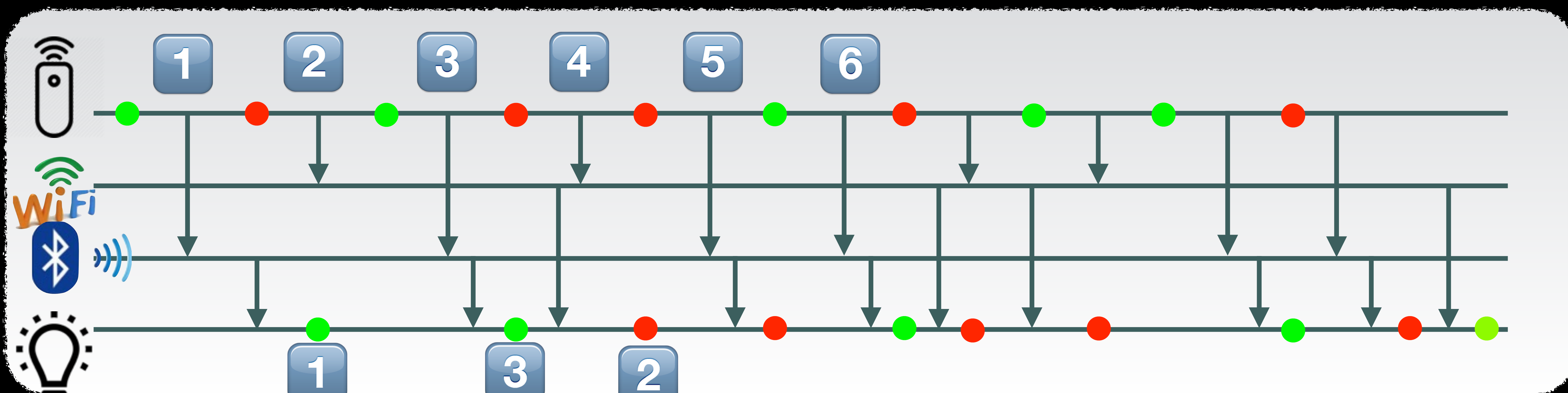

Process 1
Process 2
Process 3
Gossip 1
Gossip 2
Gossip 3
Network

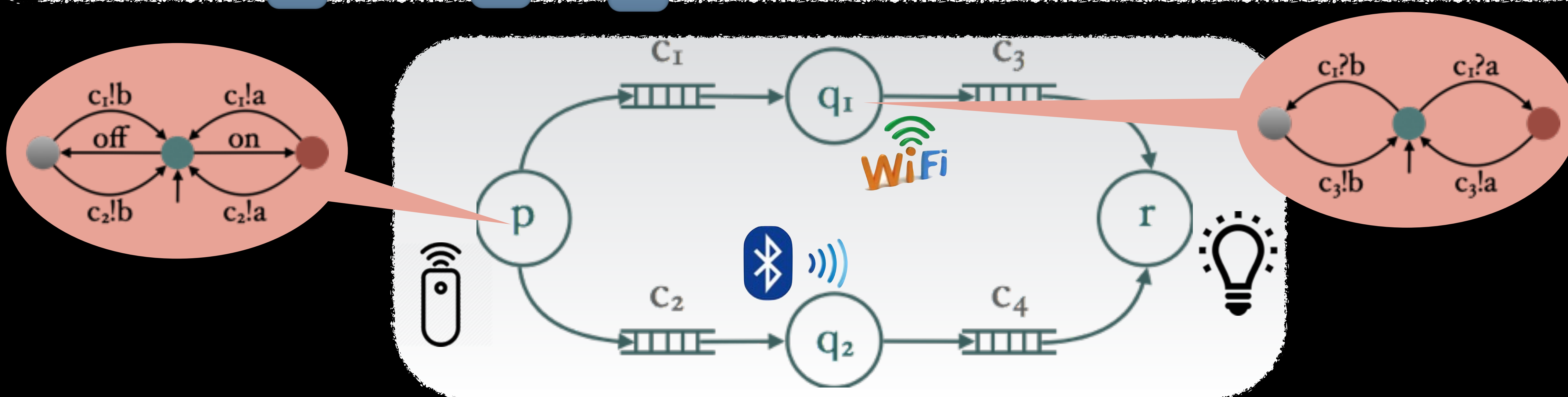
How to maintain the latest information?





How to maintain the latest information?





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

Finite communication complexity

Formal Methods
Model checking

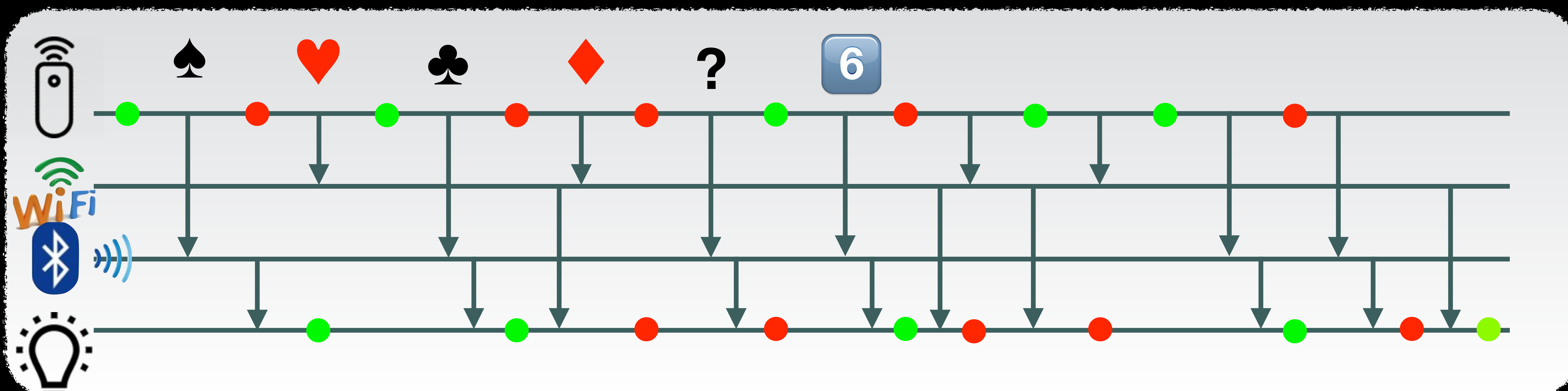
Distributed Synthesis

Global Snapshots
Causal Ordering

Bounded implementations of replicated data-types

Local testing

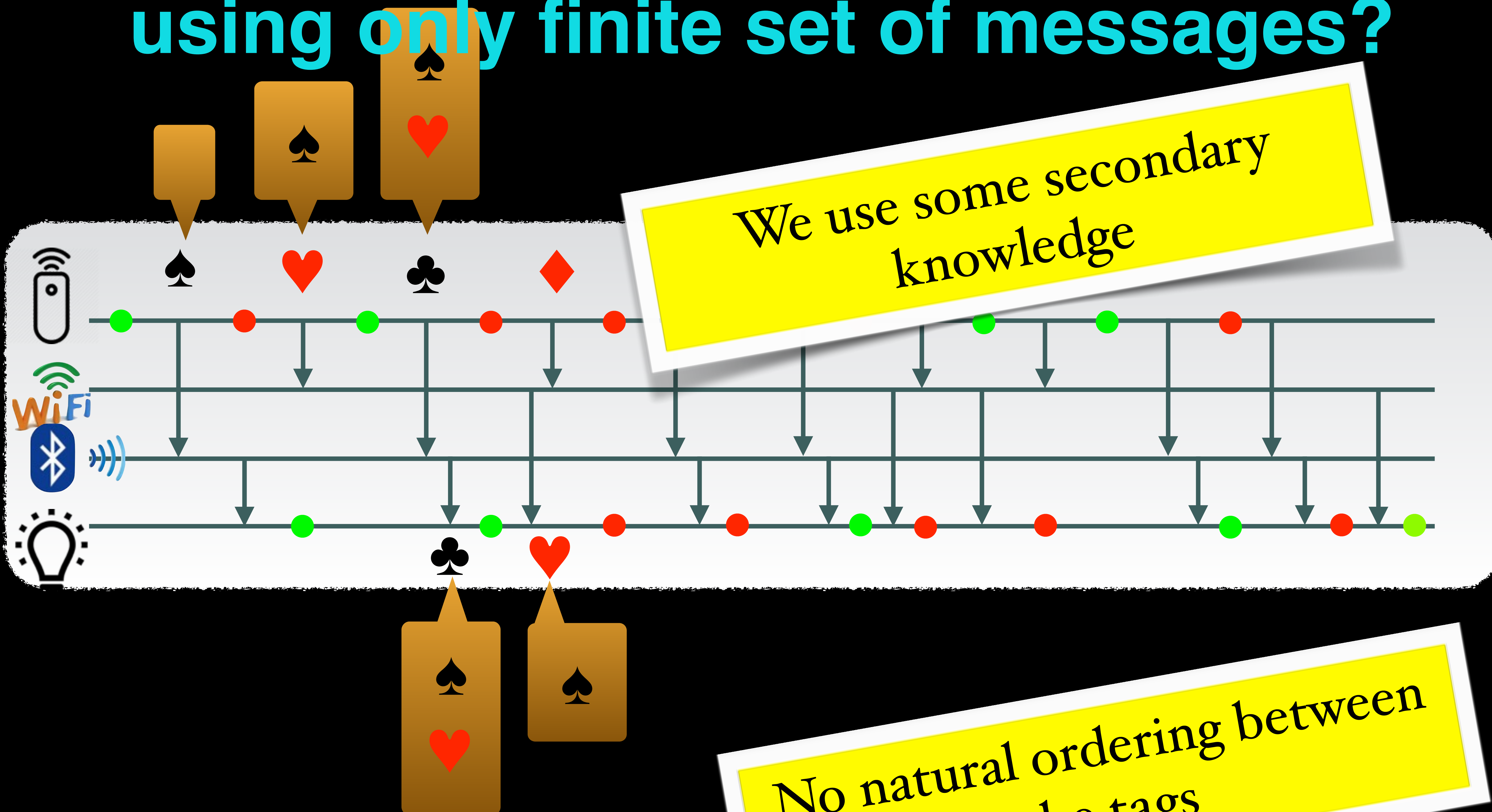
How to maintain the latest information **using only finite set of messages?**



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

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

Is it even possible?

At least in some cases?

Synchronous communication [Zielonka87]

Bounded channels [Mukund et al.03]

Beyond Bounded channels?

Challenges

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

Lets analyse for k-Bounded channels

When is a color not needed any more?

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

requires k colors

necessary, but not sufficient

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

Secondary knowledge

requires k^2 colors

colors are not freed in the order they were used

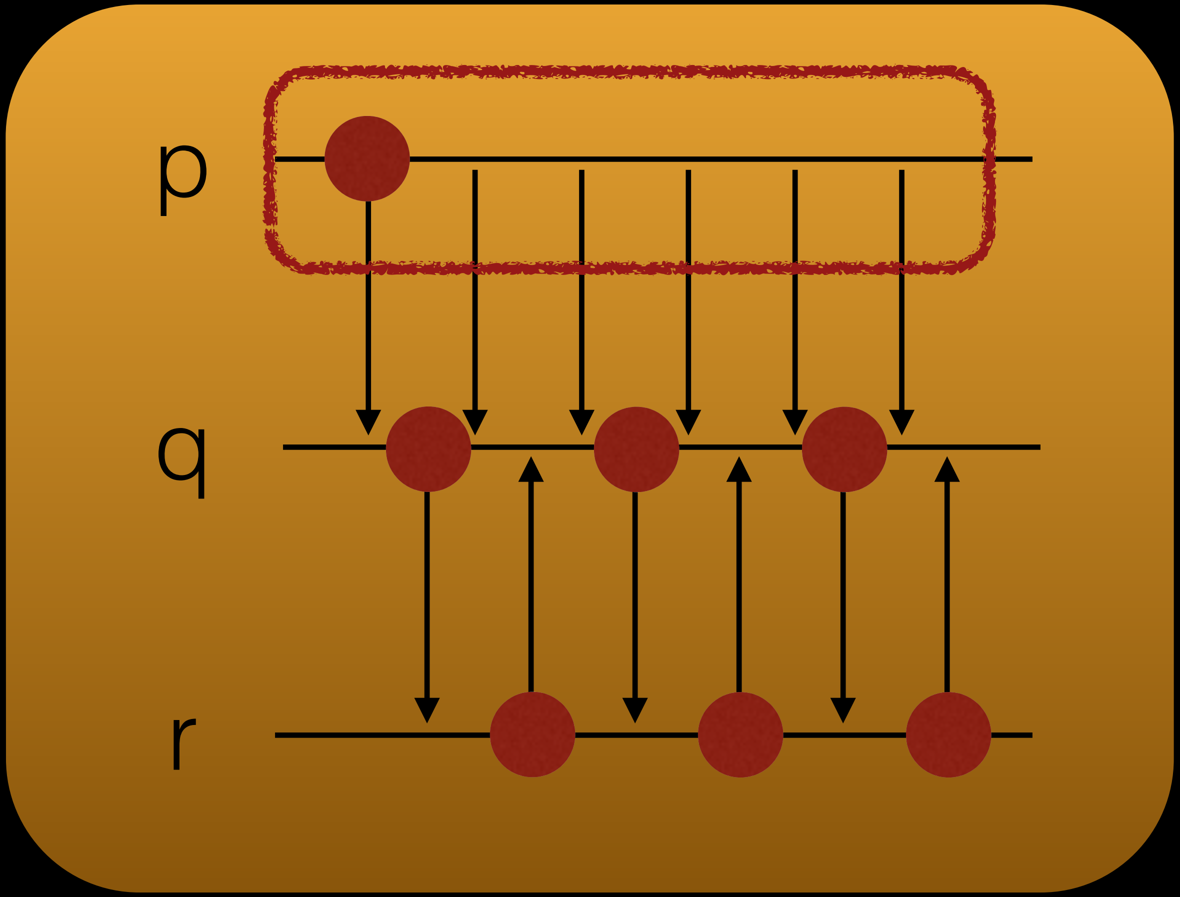
showing a bound, and using a round-robin does not work

k-Bounded channels permit finite time-stamping k-Bounded channels

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

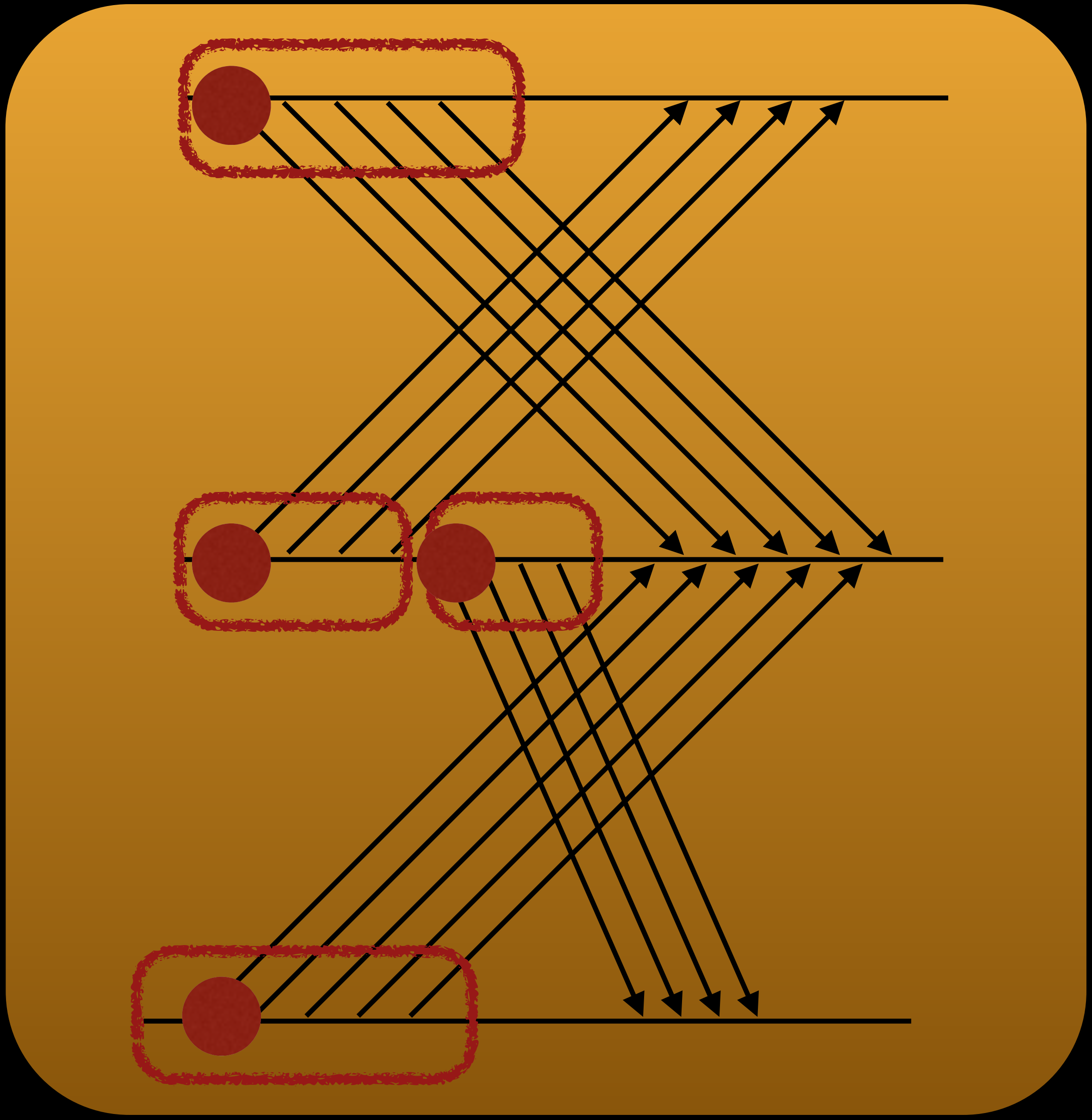
Important writes

Are existential channel bounds necessary?

Important writes

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

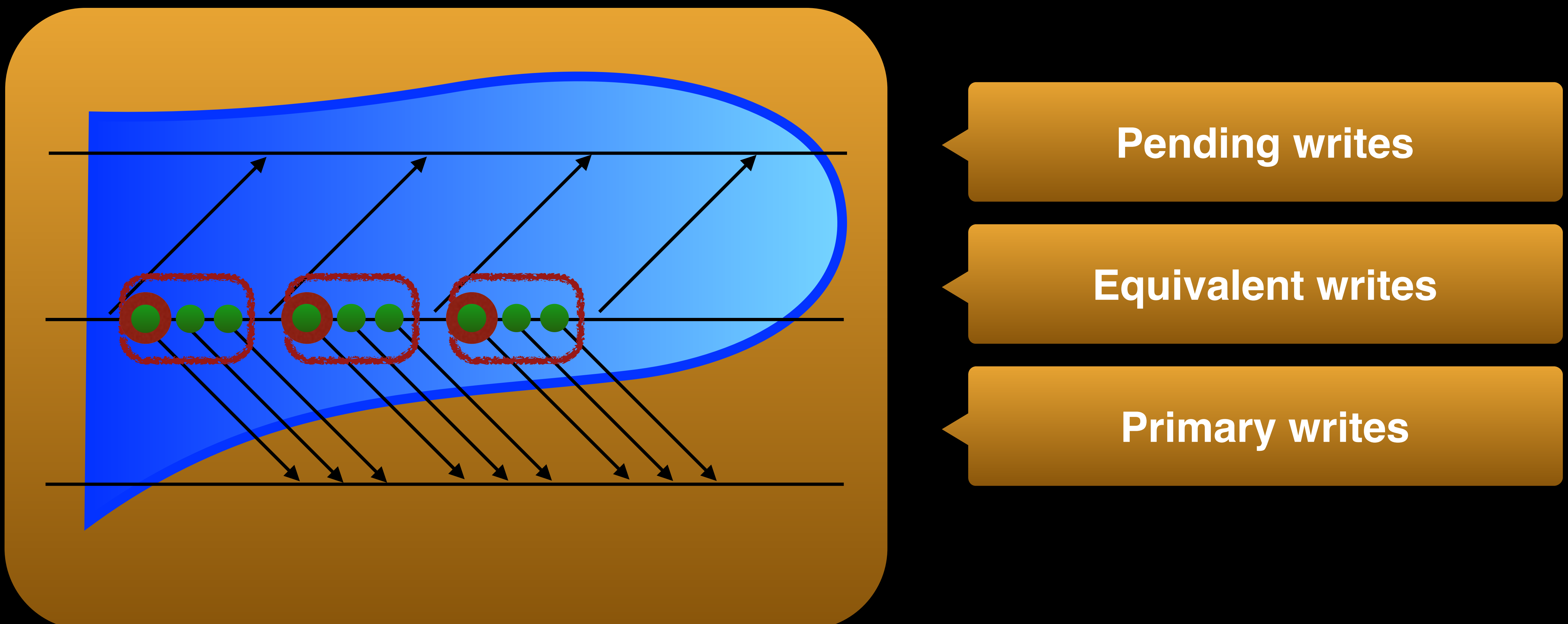
Are existential channel bounds necessary?



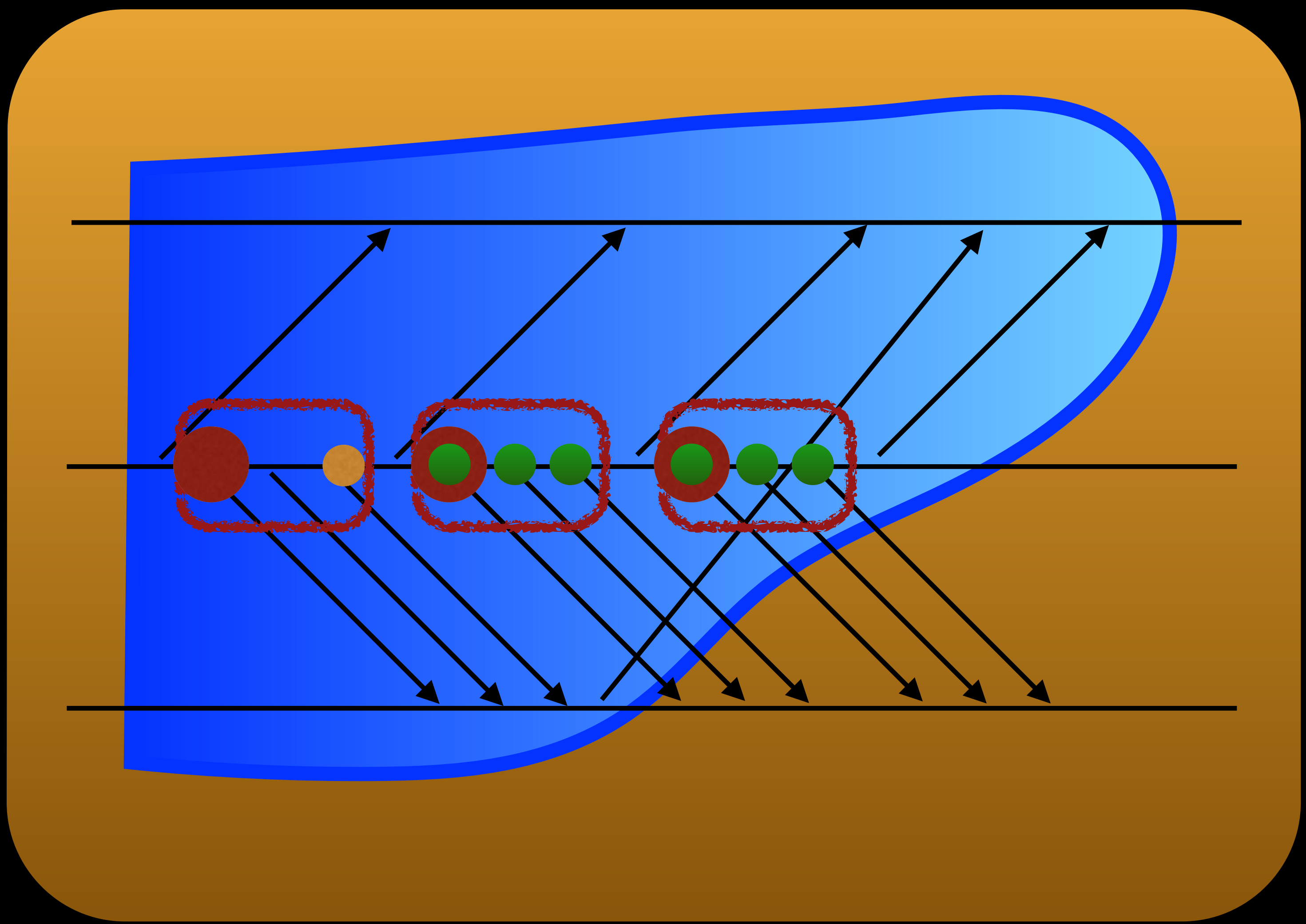
Equivalent writes

Important writes

We need some bound: Primary information



We need some bound: Primary information



Pending writes

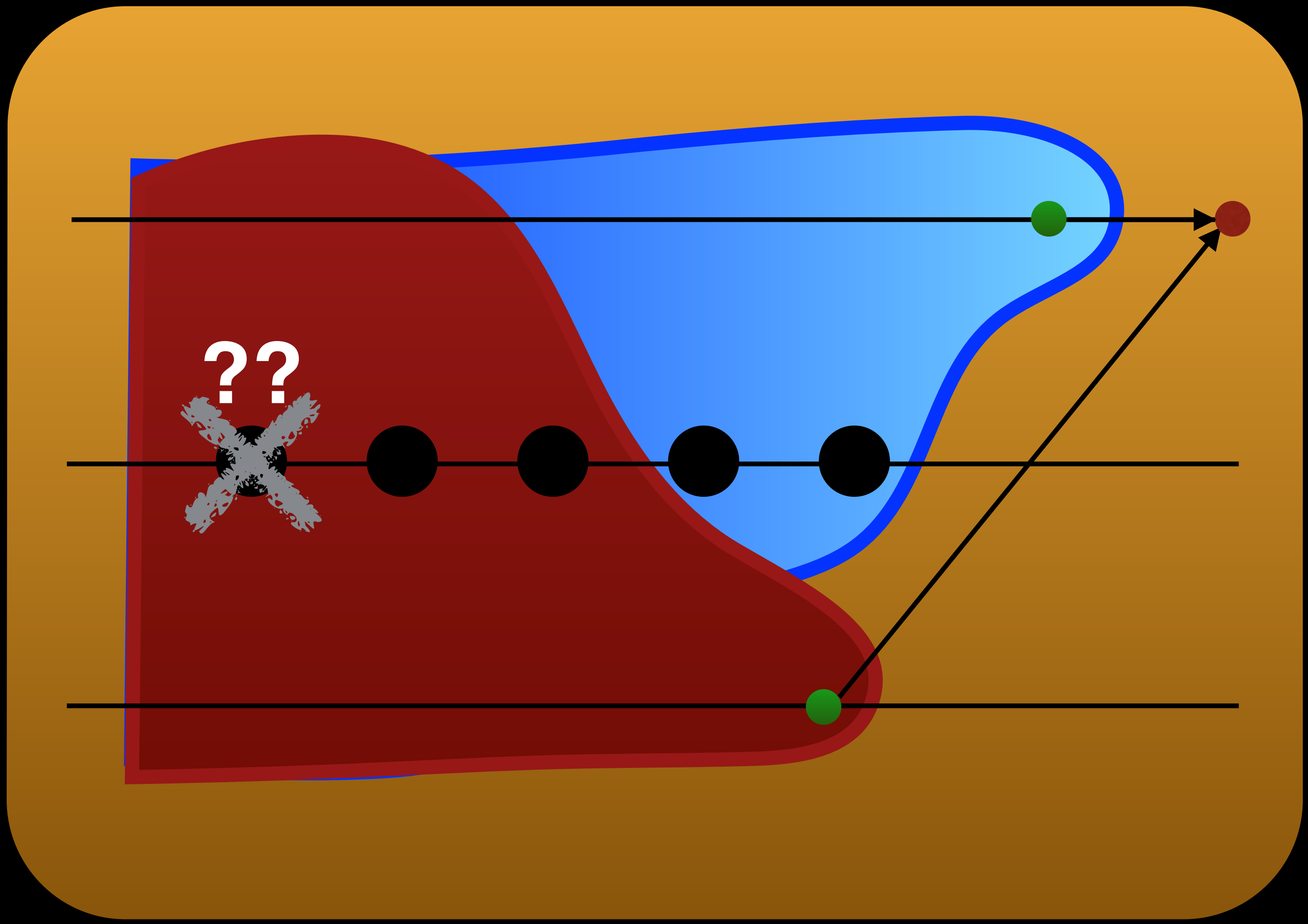
Latest received writes

Equivalent writes

Primary writes

We solve the gossip problem for primary bounded

How do we maintain the primary?

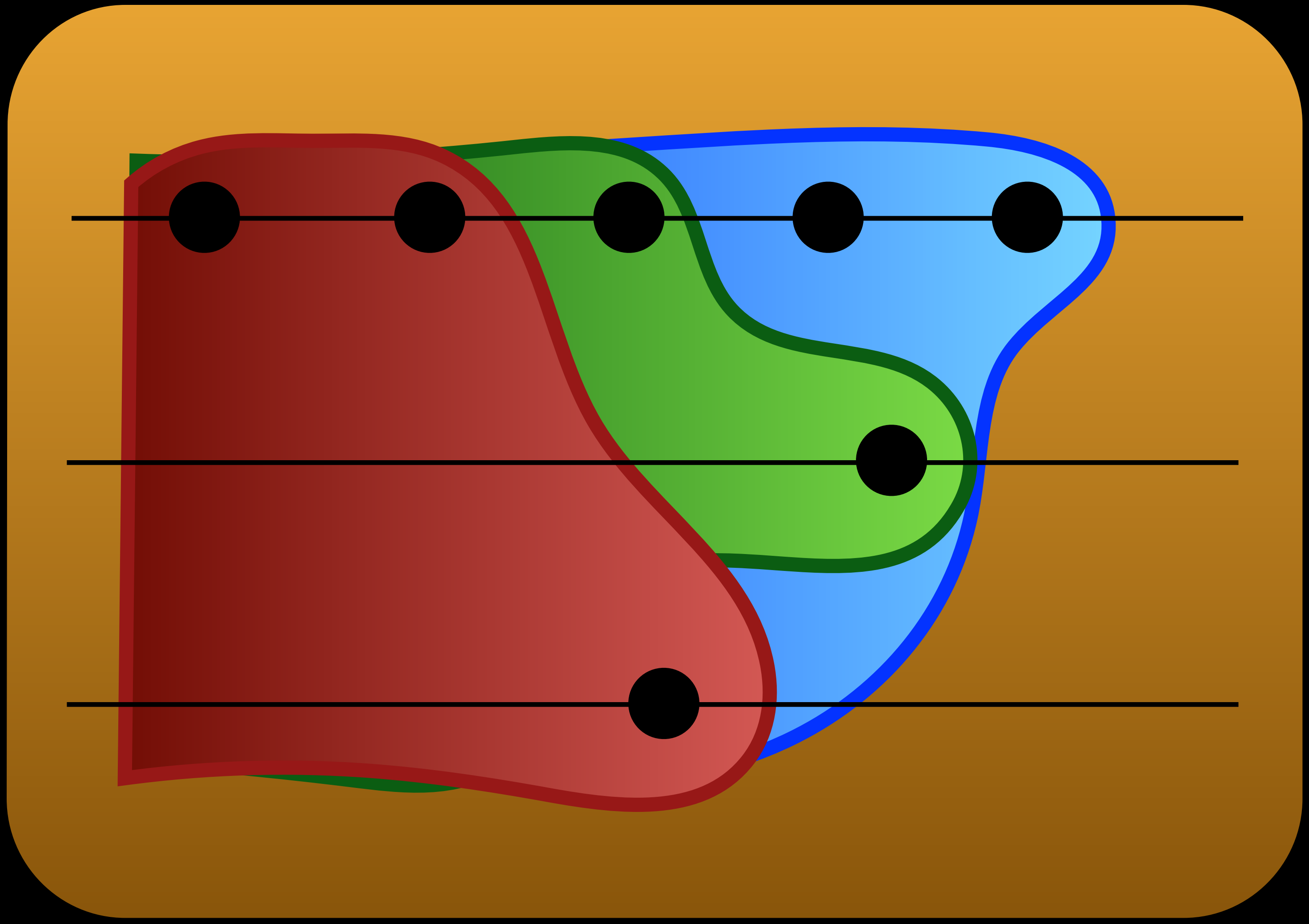


Keeping primary alone is not enough

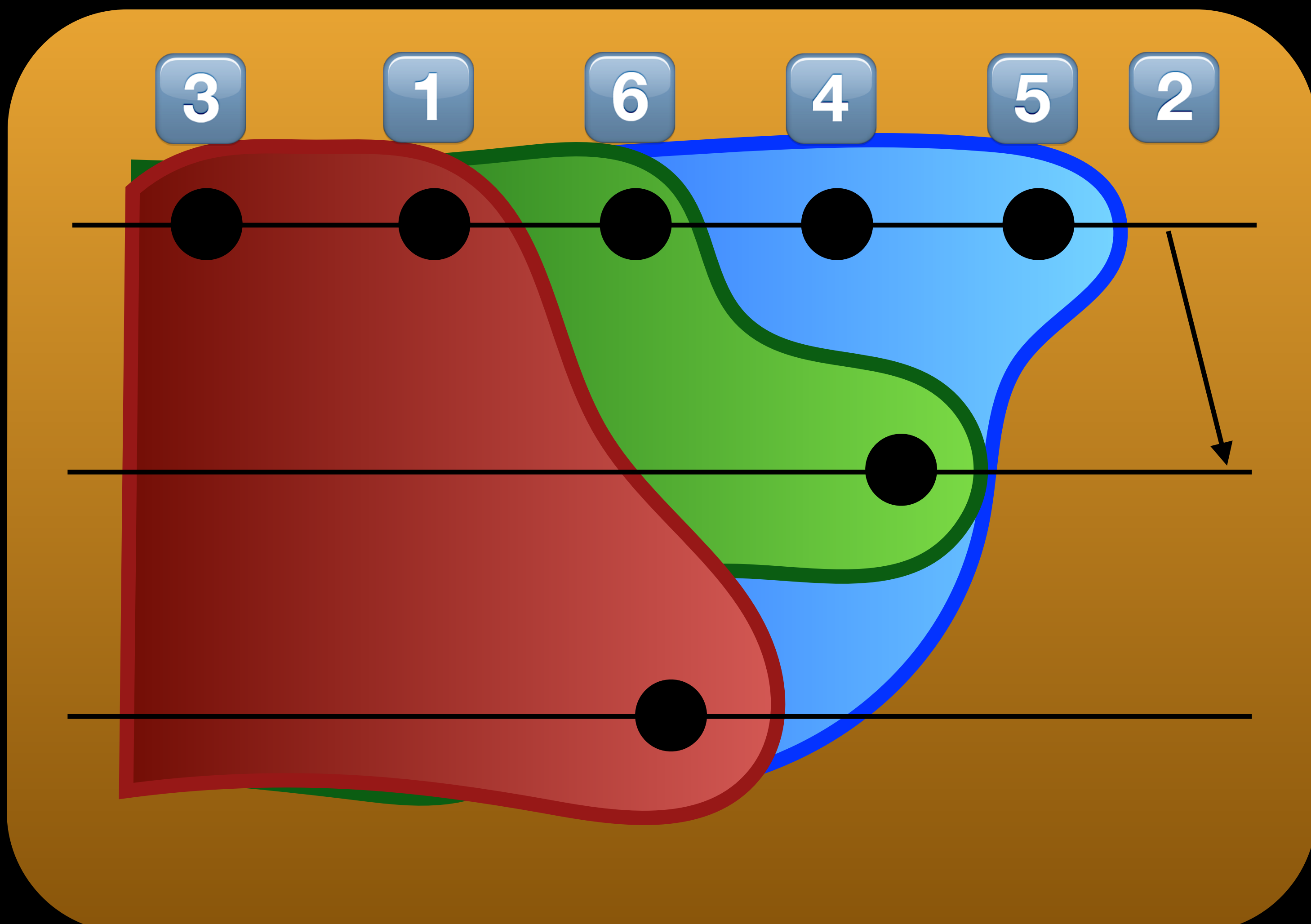
Need secondary knowledge

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?

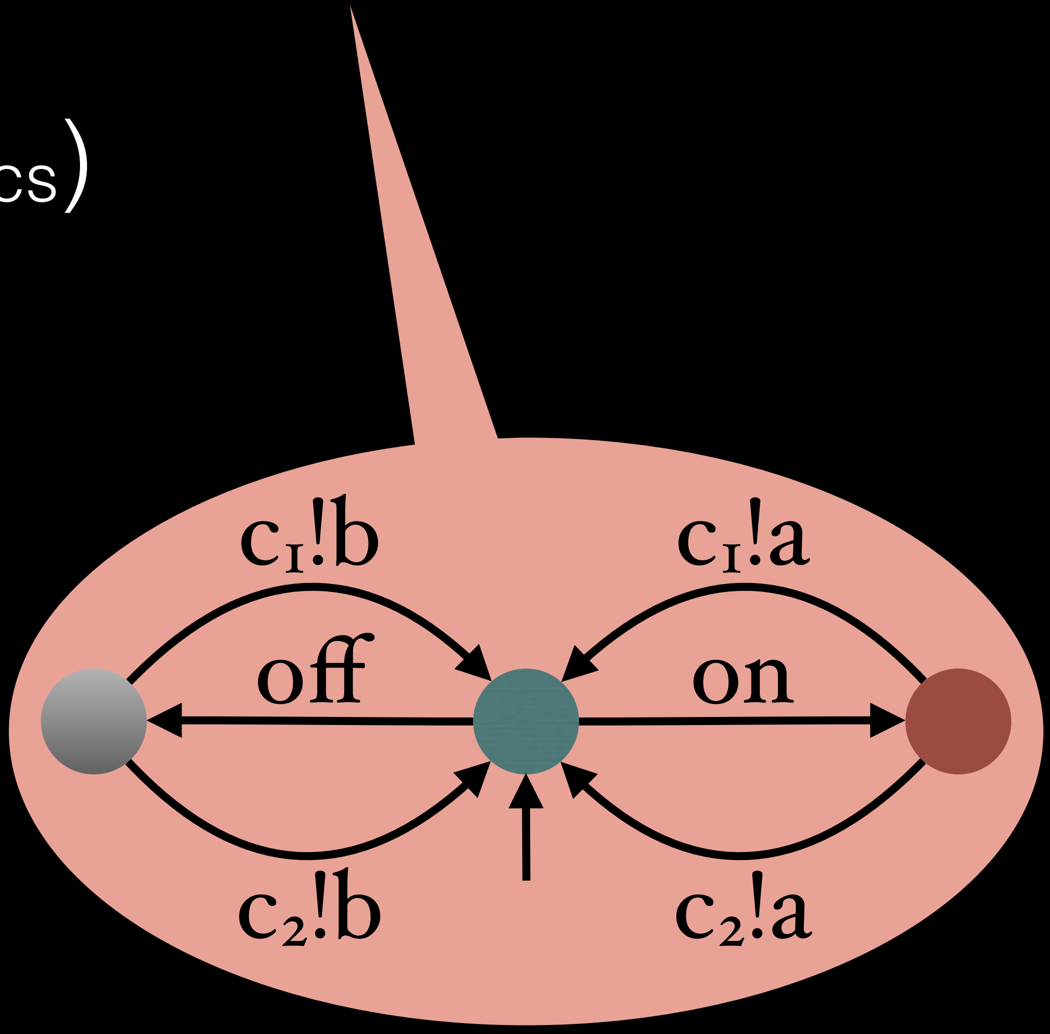
YES, WE CAN!

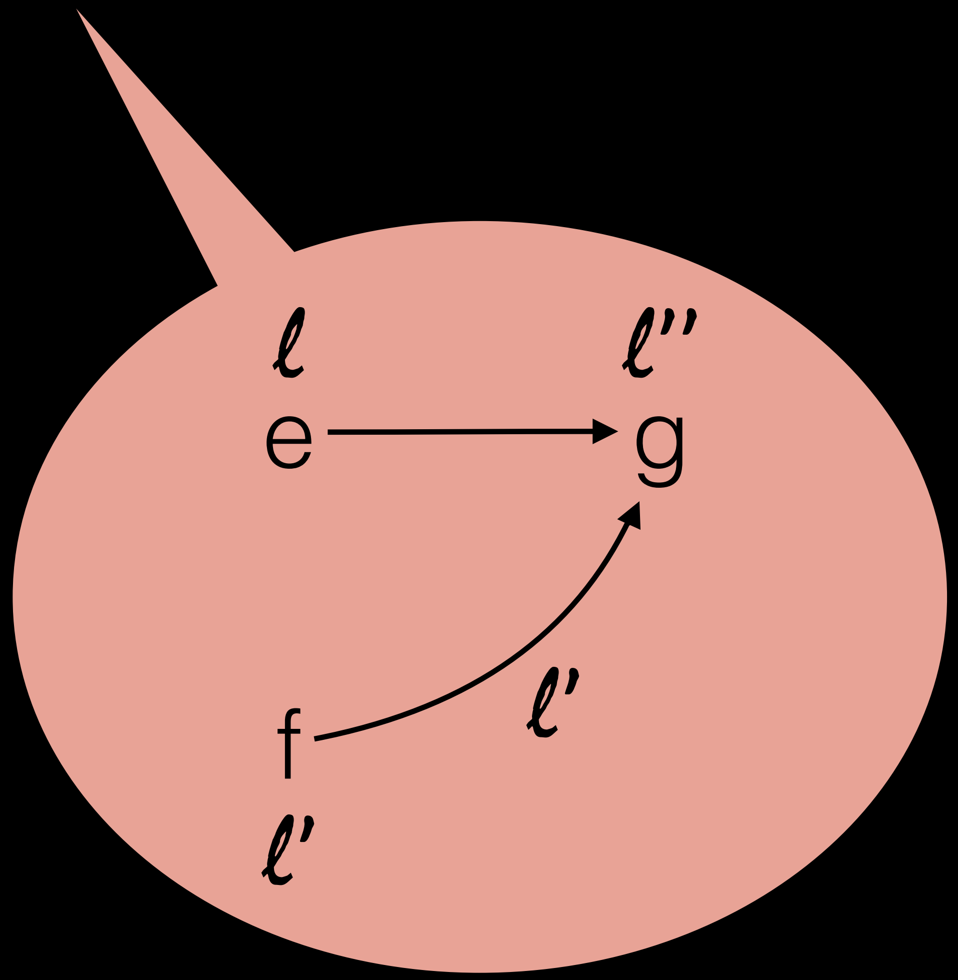
Gossip: more precisely

- Message passing automaton (MPA or CFM)

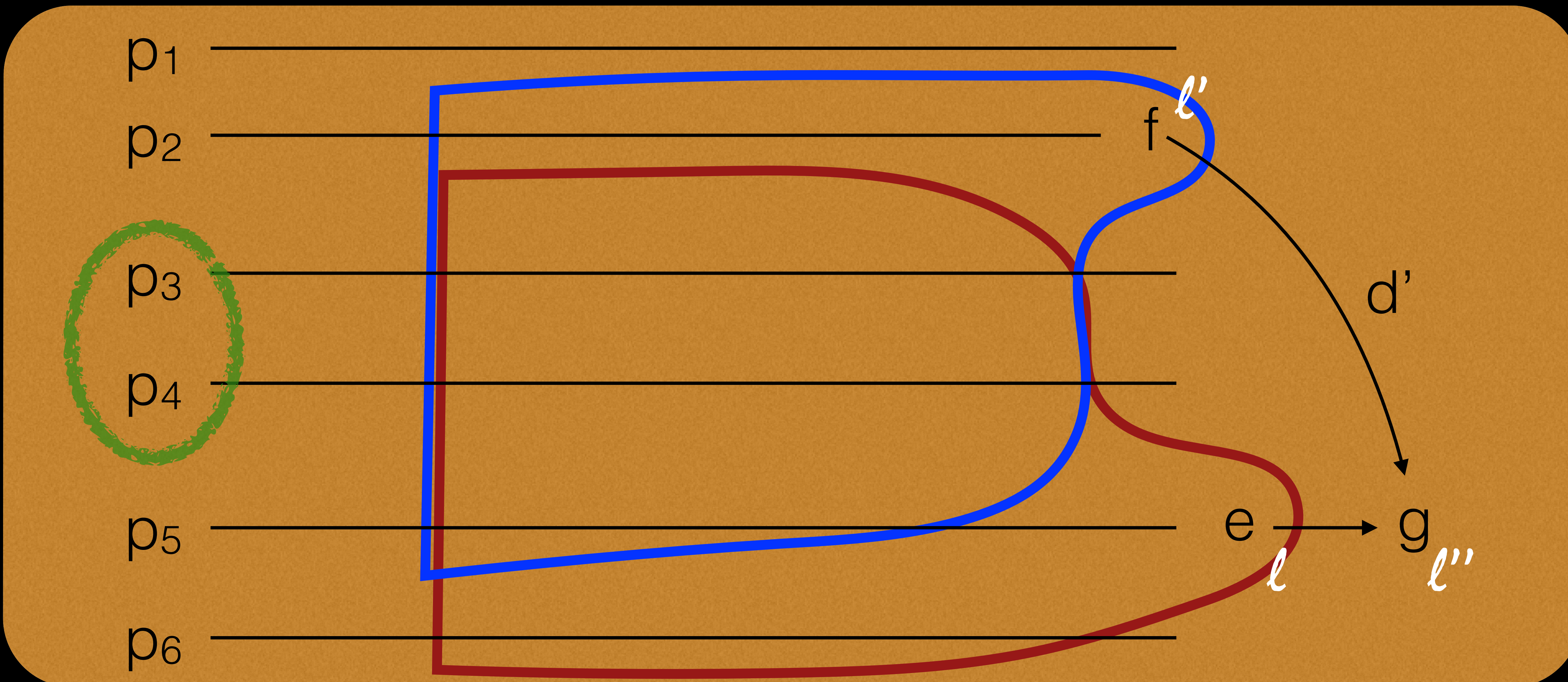
Gossip = (Locs, $(\text{Trans}_p)_{p \in \text{Procs}}$)

Run: ρ : Events → Locs





Known and Latest



Known: Locs → 2^{Procs}

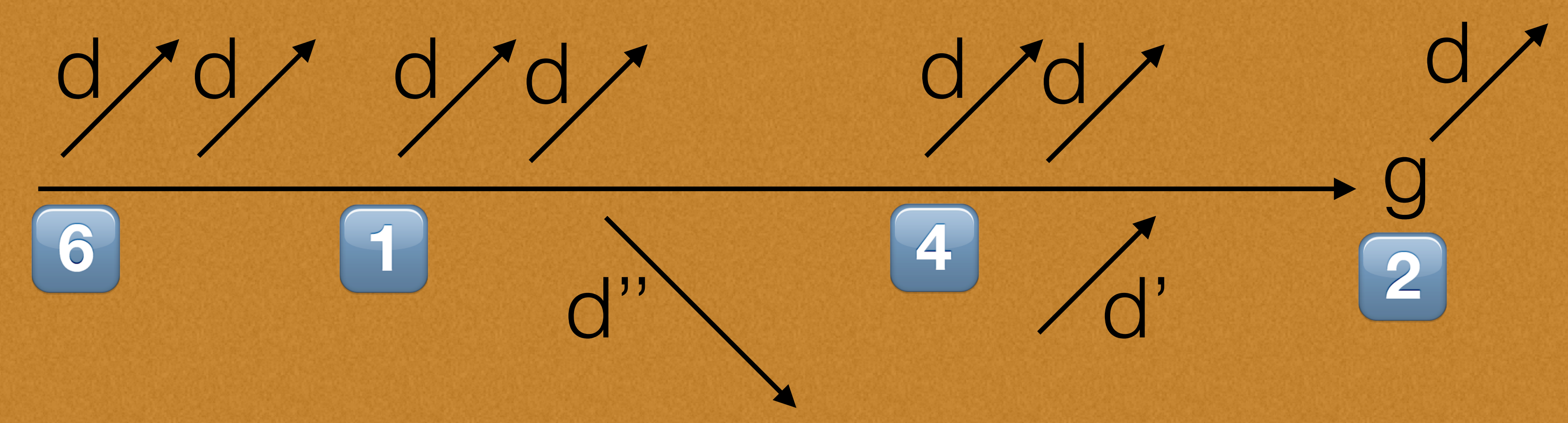
Known(ℓ'') = {p_2,p_3,…,p_6}

Latest: Locs^2 → 2^{Procs}

Latest(ℓ,ℓ') = {p_3,p_5,p_6}

Colors and time-stamps

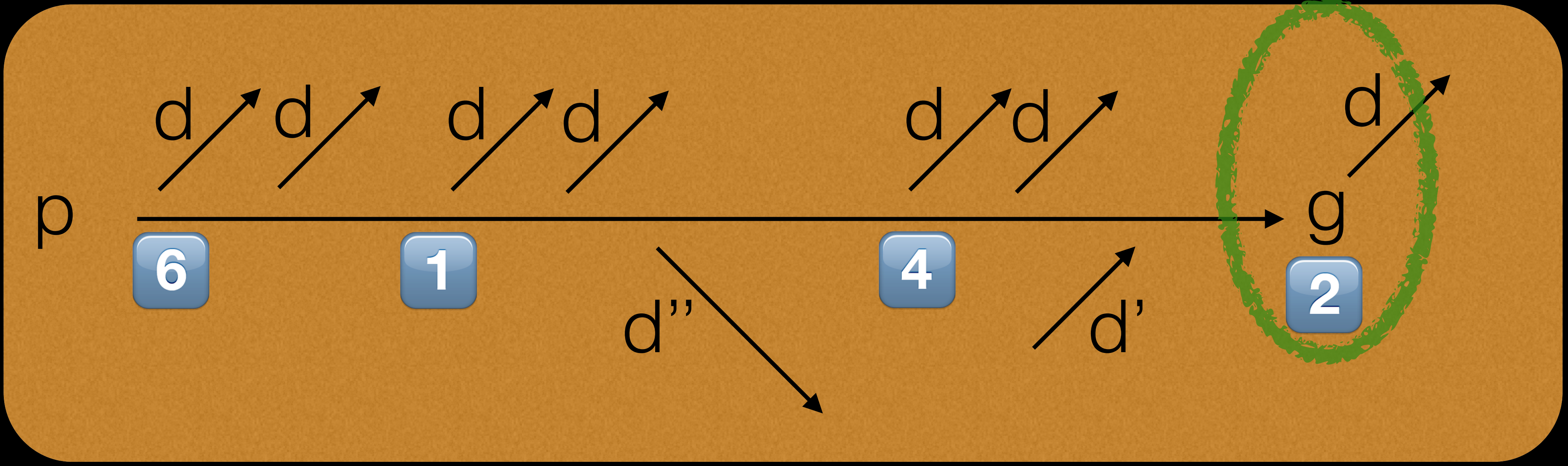
$$\chi(g) = \min(\mathbb{N} \setminus \chi(\mathsf{Sec}(\Downarrow g) \cap \mathsf{Send}(d)))$$



$$h(g) = (d, \chi(g))$$

$$K^1(g) = \{h(e) \mid e \in \mathsf{Prim}(\downarrow g)\}$$

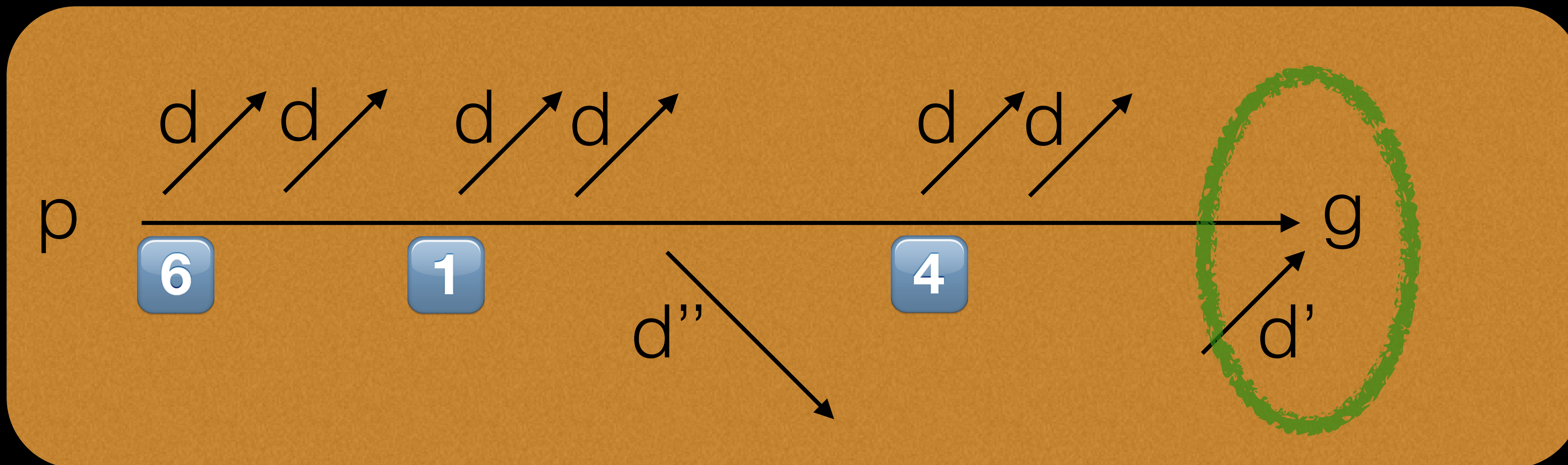
Locations of Gossip



$$K^2(g) = (\mathsf{pid}(g), d, c, K^1(g), (K^1(e))_{e \in \mathsf{Prim}(g)})$$

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

Locations of Gossip

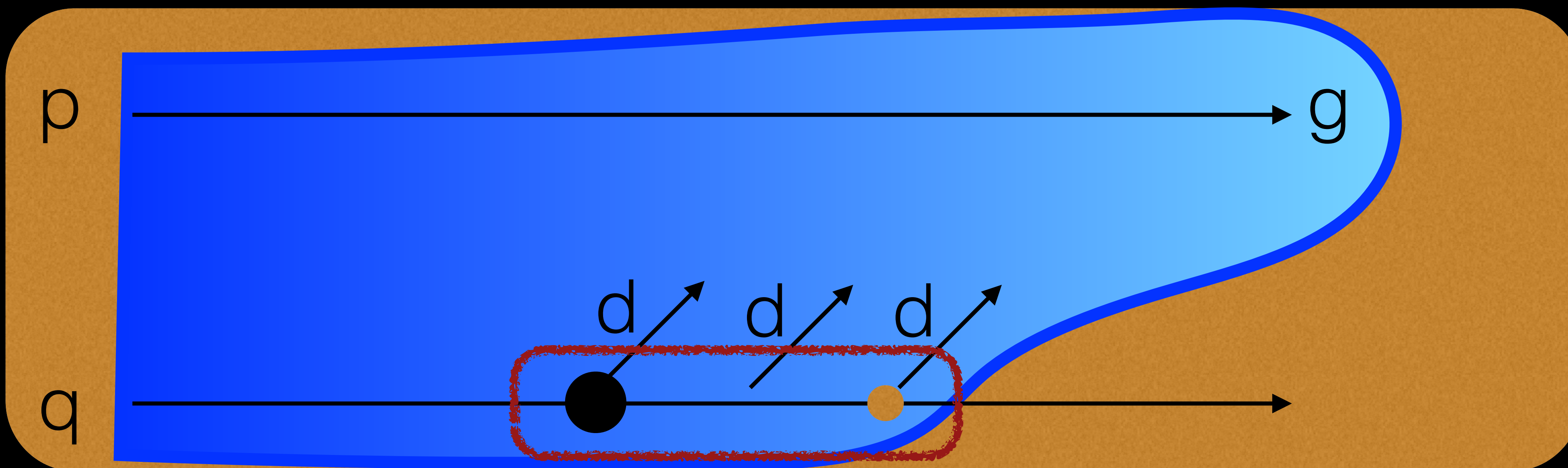


$\perp, \perp$

$$K^2(g) = (\mathsf{pid}(g), d, c, K^1(g), (K^1(e))_{e \in K^1(g)})$$

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

Known



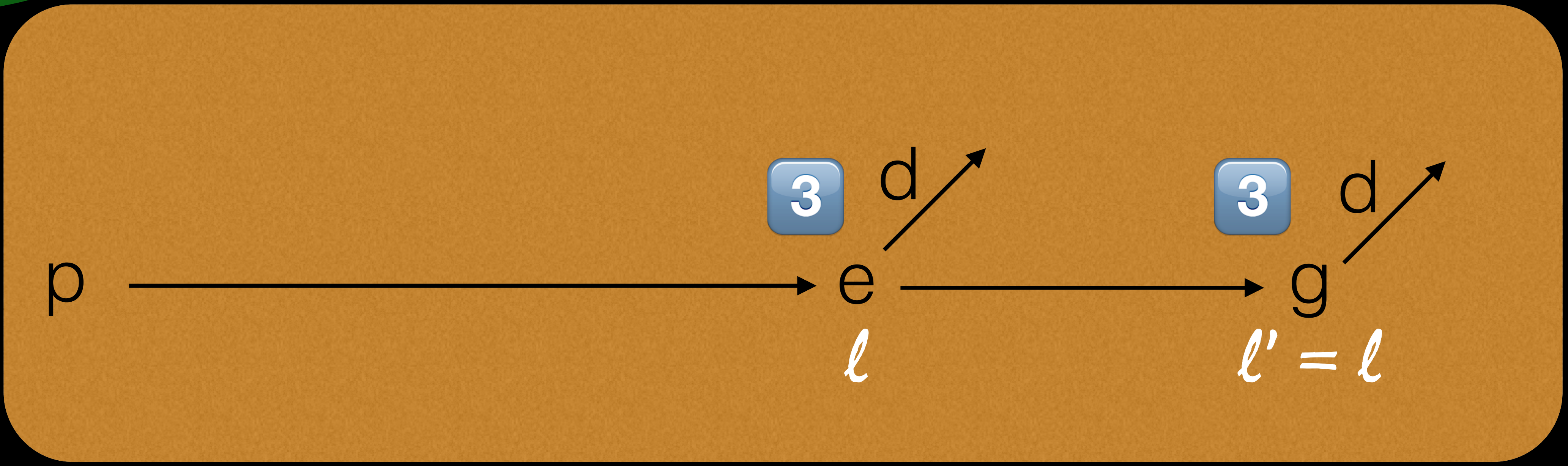
$\mathsf{pid}(\downarrow g) = \{\mathsf{pid}(g)\} \cup \mathsf{pid}(\mathsf{Prim}(\downarrow g))$

$\mathsf{Known}(\ell) = \{p\} \cup \mathsf{pid}(P)$

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

write event case 1

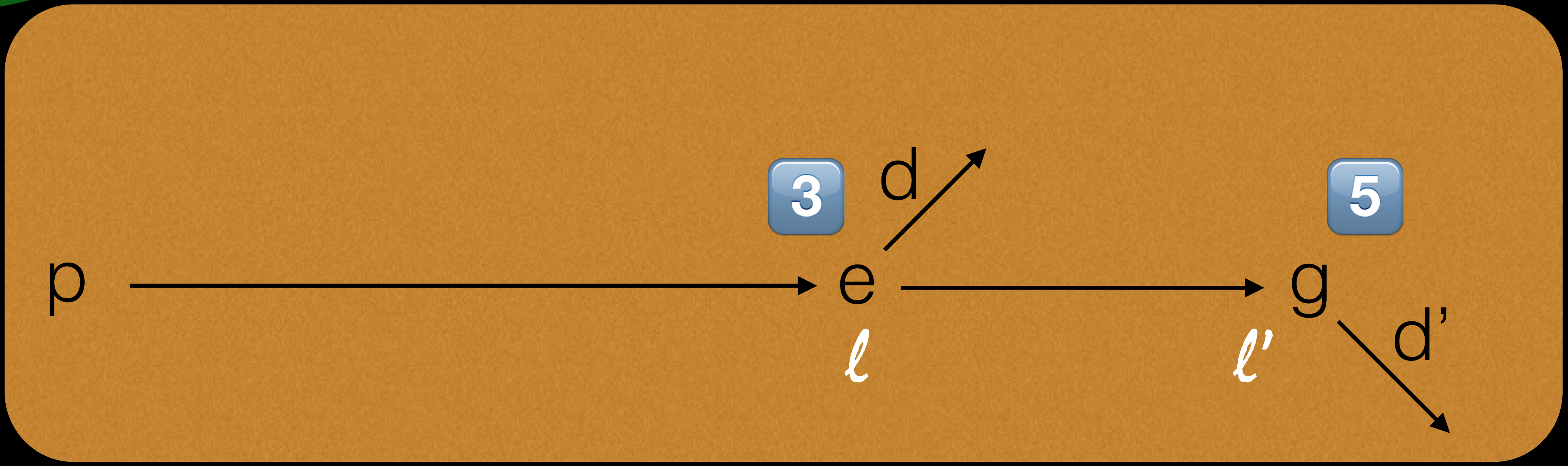
Maintaining K²



Equivalent writes : no changes

Maintaining K^2

write event case 2

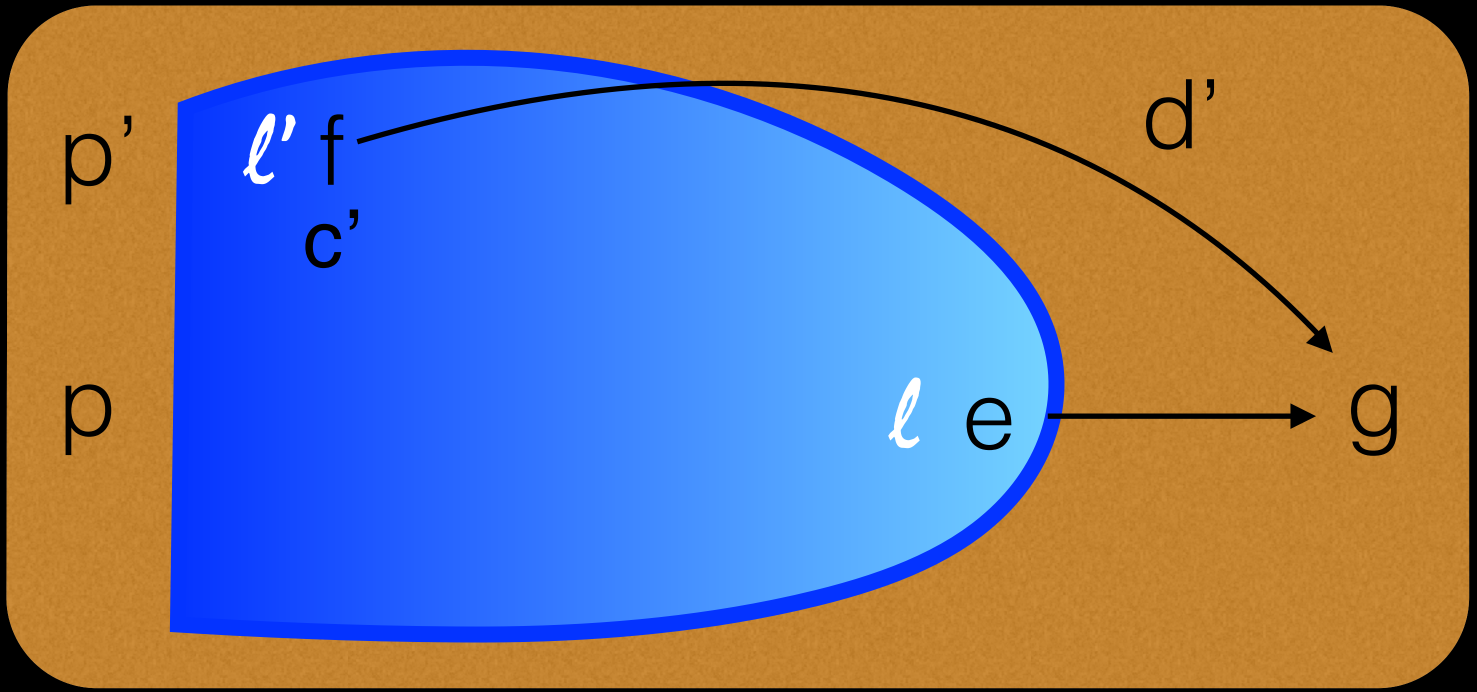


New channel: requires an available color

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

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

Maintaining K^2

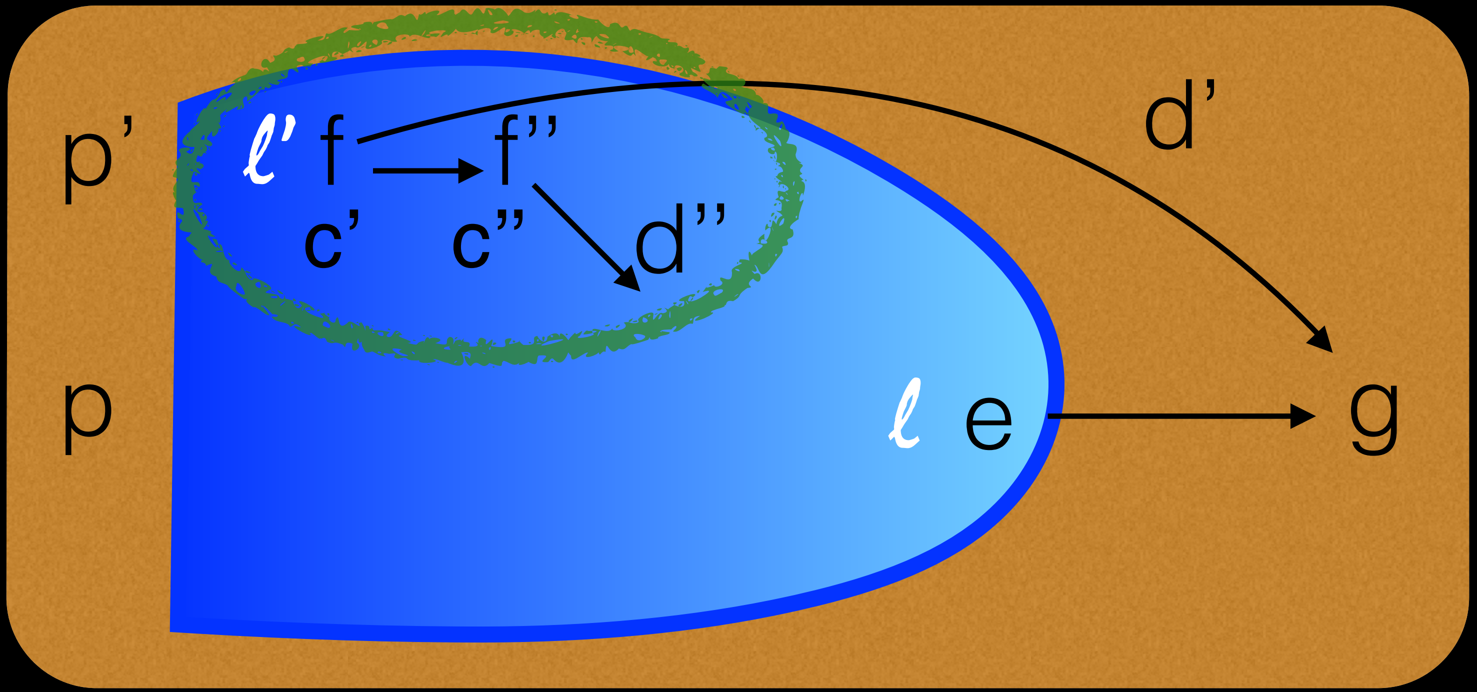


$$\ell = (p, d, c, P, (S_\gamma)_{\gamma \in P})$$
$$\ell' = (p', d', c', P', (S'_\gamma)_{\gamma \in P'})$$

f < e iff

$$(d', c') \in P \wedge \exists (d'', c'') \in P \setminus P' \, (d'' \neq d' \wedge \mathsf{W}(d'') = \mathsf{W}(d'))$$

Maintaining K^2



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

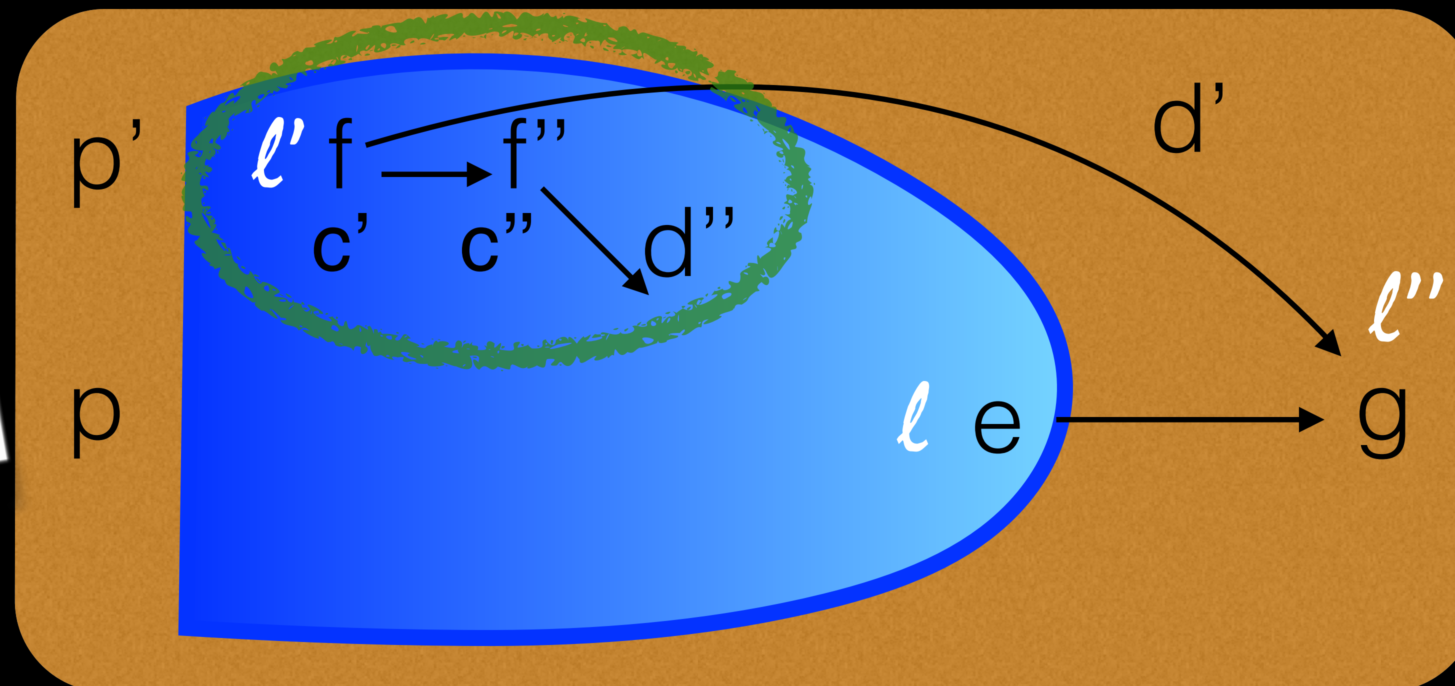
f < e iff

$$(d', c') \in P \wedge \exists (d'', c'') \in P \setminus P' (d'' \neq d' \wedge \mathsf{W}(d'') = \mathsf{W}(d'))$$

Maintaining K^2

read event case 1

Latest(ℓ,ℓ') = Known(ℓ)



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

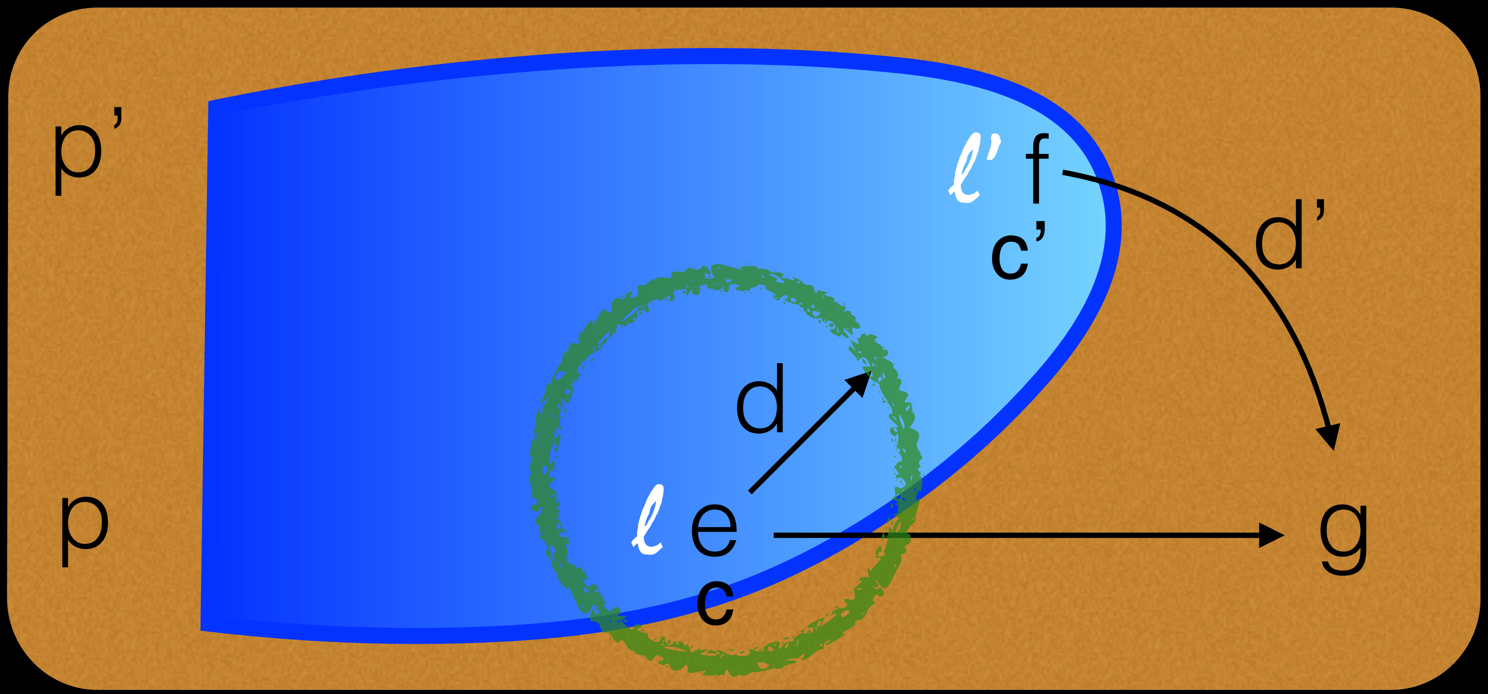
$$\ell'' = (p, \perp, \perp, P'', (S''_\gamma)_{\gamma \in P''})$$
$$P'' = (P \setminus (P' \cap d')) \cup \{(d', c')\}$$

f < e iff

$$(d', c') \in P \wedge \exists (d'', c'') \in P \setminus P' (d'' \neq d' \wedge \mathsf{W}(d'') = \mathsf{W}(d'))$$

Maintaining K^2

read event
case 2

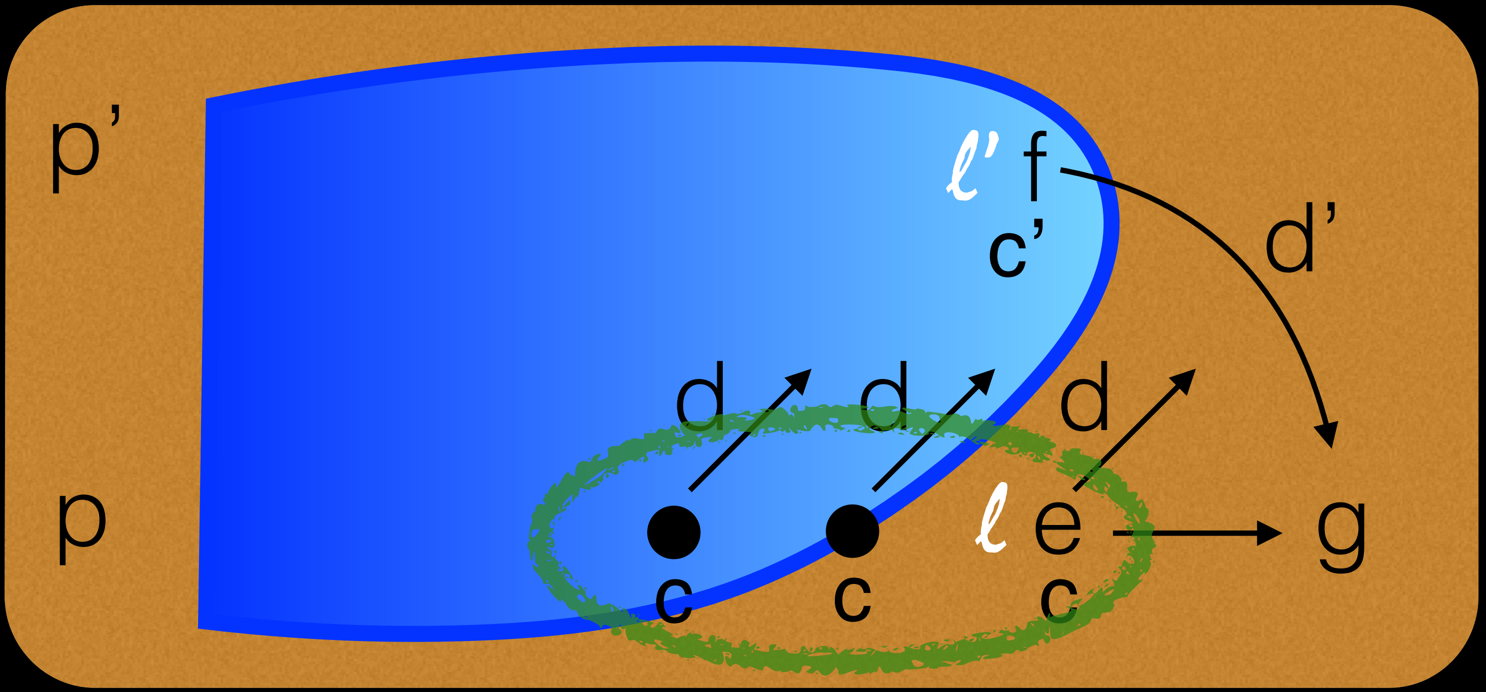


$$\ell = (p, d, c, P, (S_\gamma)_{\gamma \in P})$$
$$\ell' = (p, d', c', P', (S'_\gamma)_{\gamma \in P'})$$

e < f implies (d,c) ∈ P'

read event case 2

Maintaining K^2



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

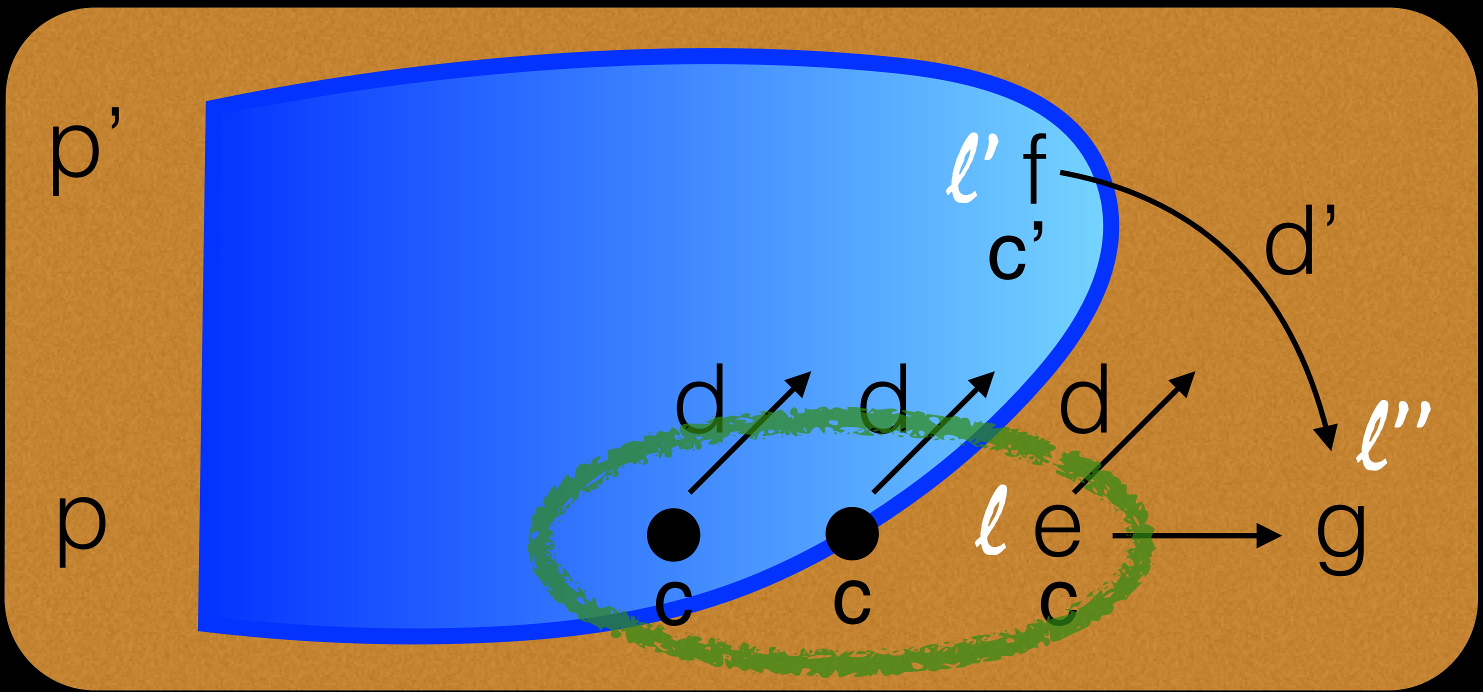
e < f implies (d,c) ∈ P'

not iff

Maintaining K²

read event case 2

Latest(ℓ,ℓ') = {p}



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

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

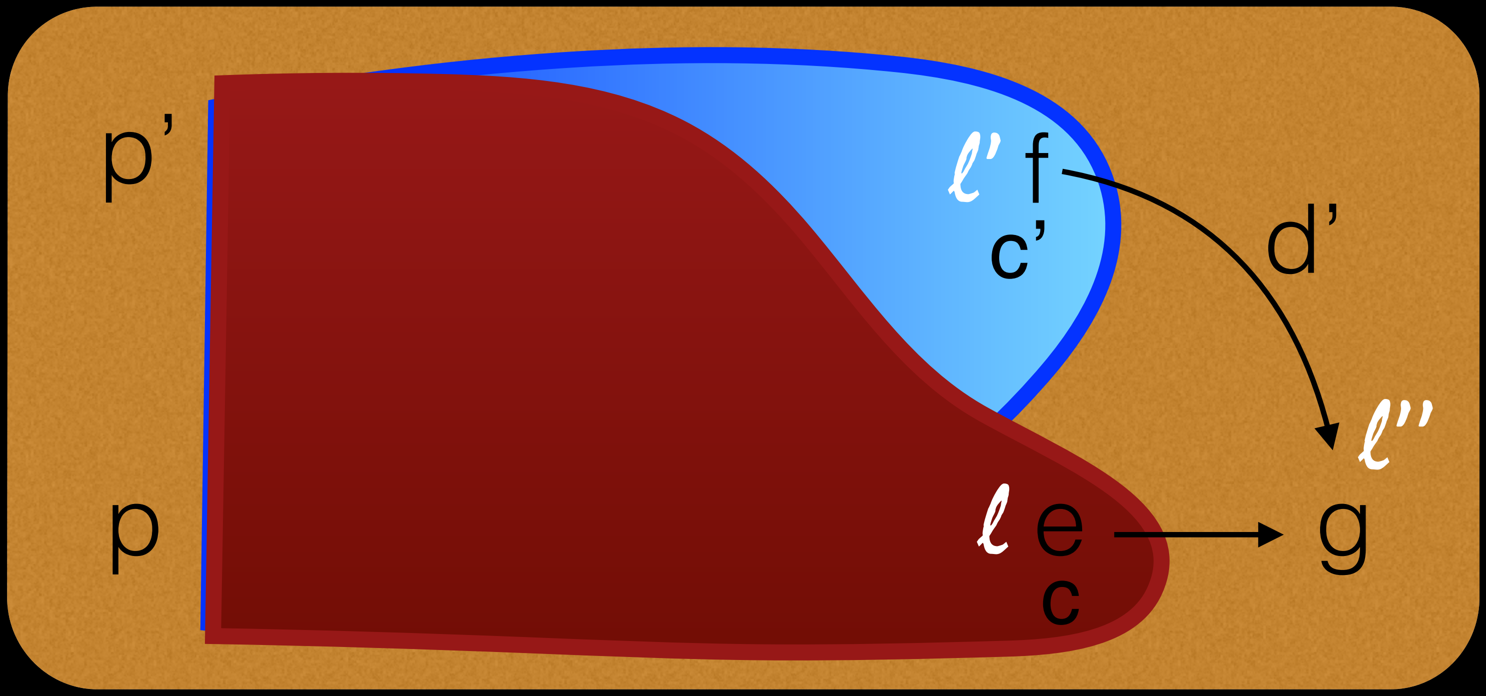
e < f implies (d,c) ∊ P’

not case 1 and (d,c) ∊ P’ implies …

not iff

Maintaining K^2

read event case 3



not (case 1 or case 2) implies e || f

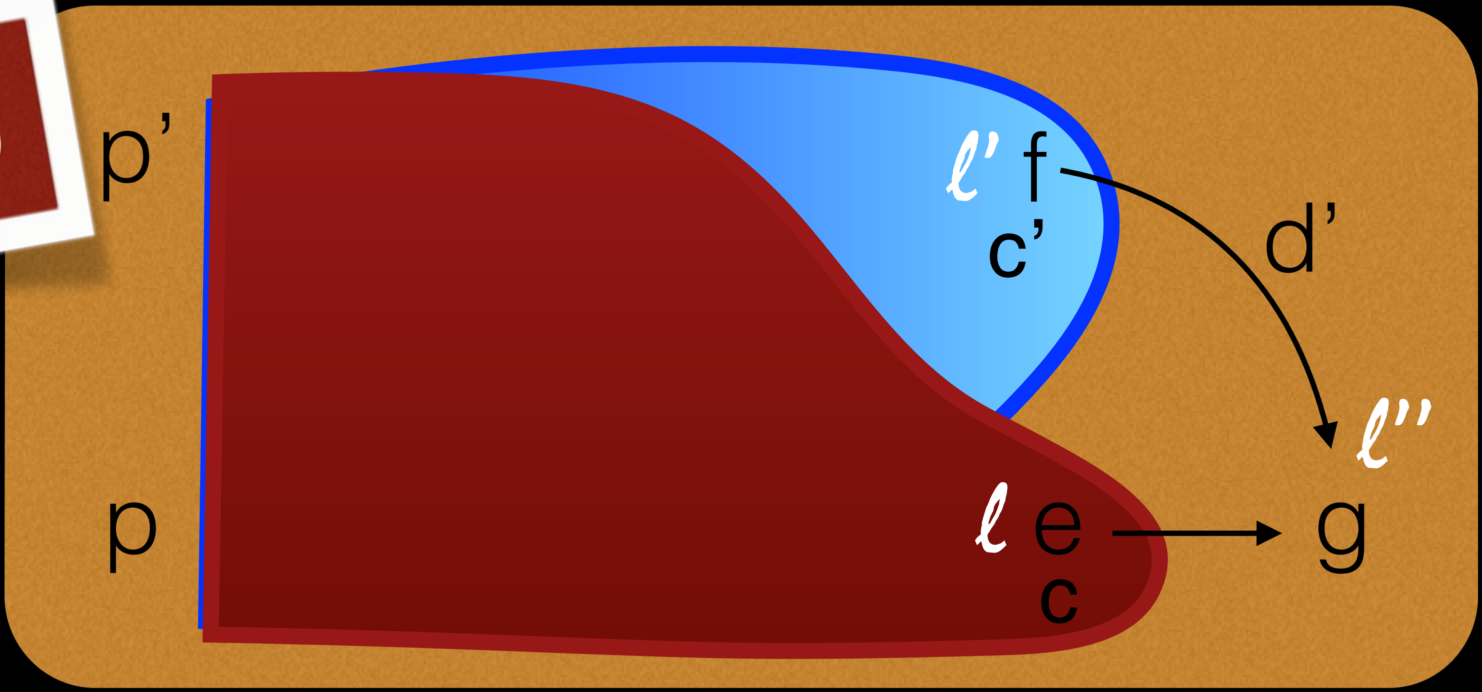
$$\mathsf{Prim}(\Downarrow g) = (\mathsf{Prim}(\downarrow e) \cap \mathsf{Prim}(\downarrow f)) \cup (\mathsf{Prim}(\downarrow e) \setminus \downarrow f) \cup (\mathsf{Prim}(\downarrow f) \setminus \downarrow e)$$

$$\mathsf{Prim}(\downarrow e) \setminus \bigcup_{e' \in \mathsf{Prim}(\downarrow e) \cap \mathsf{Prim}(\downarrow f)} \mathsf{Prim}(\downarrow e')$$

Maintaining K^2

read event
case 3

h injective on Sec(↓e ∪ ↓f)



not (case 1 or case 2) implies e || f

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

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

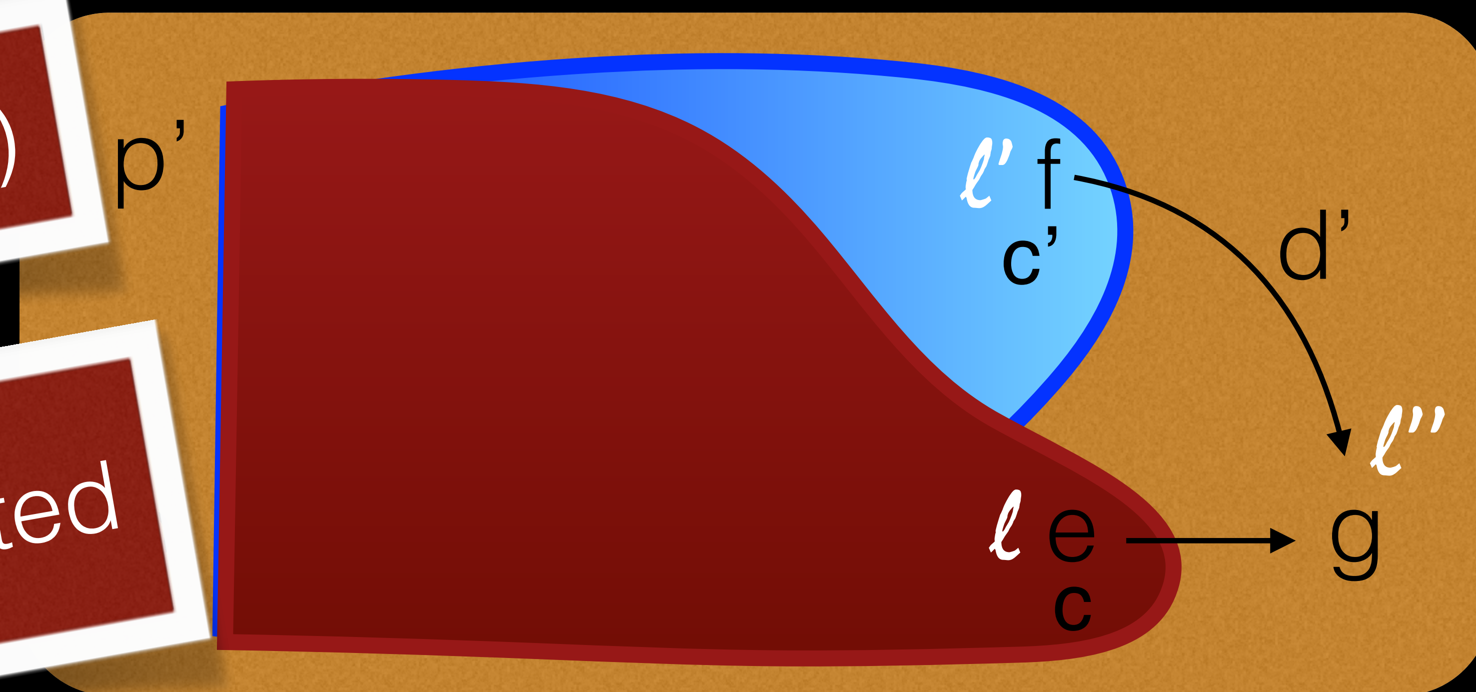
$$\mathsf{Prim}(\downarrow e) \setminus \bigcup_{e' \in \mathsf{Prim}(\downarrow e) \cap \mathsf{Prim}(\downarrow f)} \mathsf{Prim}(\downarrow e')$$

Maintaining K^2

read event case 3

h injective on Sec(↓e ∪ ↓f)

Latest(ℓ,ℓ') can be computed



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

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

not (case 1 or case 2) implies e || f

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

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

Is it even possible?

At least in some cases?

Synchronous communication [Zielonka87]

Bounded channels [Mukund et al.03]

Primary bounded