

3: Maintaining network state

state: information *stored* in network nodes by network protocols

- ❑ updated when network “conditions” change
- ❑ stored in multiple nodes
- ❑ often associated with end-system generated call or session
- ❑ examples:
 - RSVP routers maintain lists of upstream sender IDs, downstream receiver reservations
 - ATM switches maintain lists of VCs: bandwidth allocations, VCI/VPI input-output mappings
 - TCP: Sequence numbers, timer values, RTT estimates

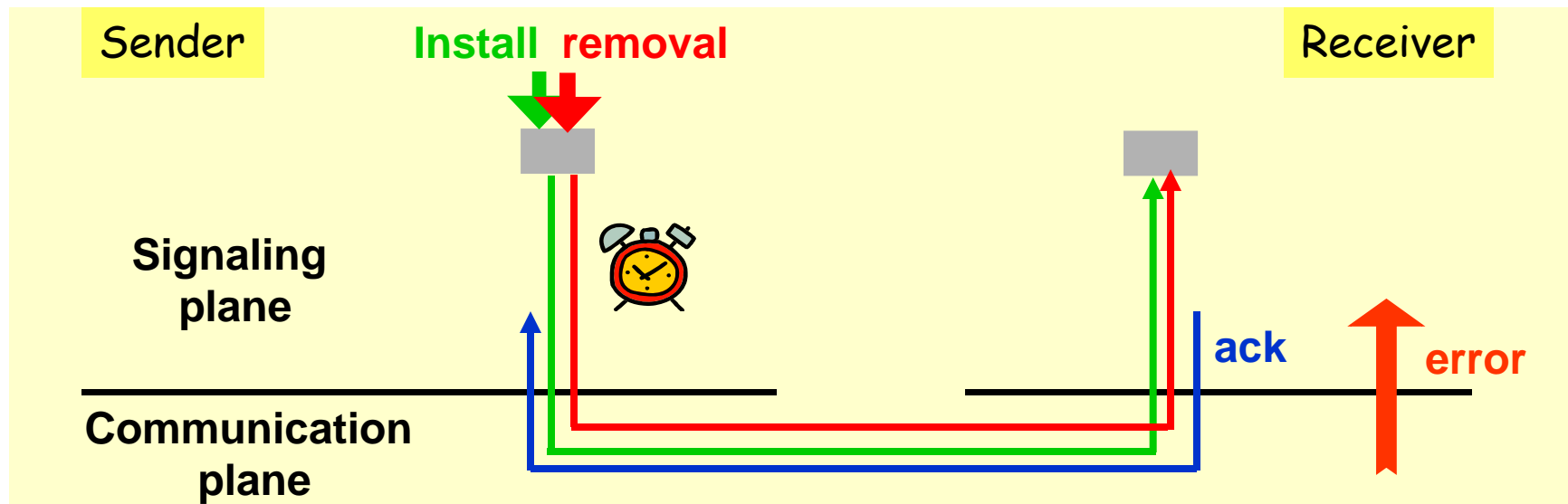
State: senders, receivers

- **sender:** network node that (re)generates signaling (control) msgs to install, keep-alive, remove state from other other nodes
- **receiver:** node that creates, maintains, removes state based on signaling msgs received from sender

Hard-state

- state *installed* by receiver on receipt of *setup msg* from sender
- state *removed* by receiver on receipt of *teardown msg* from sender
- *default assumption*: state valid unless told otherwise
 - in practice: failsafe-mechanisms (to remove orphaned state) in case of sender failure e.g., receiver-to-sender "heartbeat": is this state still valid ?
- examples:
 - Q.2931 (ATM Signaling)
 - ST-II (Internet hard-state signaling)
 - TCP

Hard-state signaling

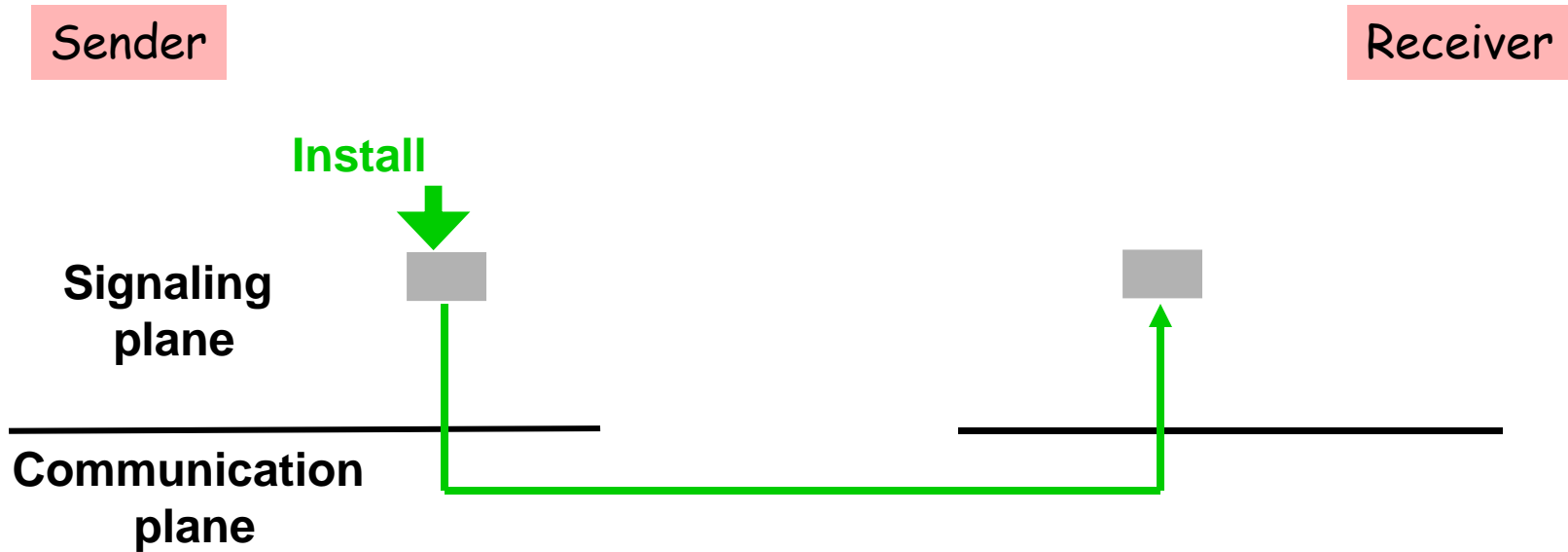


- ❑ reliable signaling
- ❑ state removal by request
- ❑ requires additional error handling
 - e.g., sender failure

Soft-state

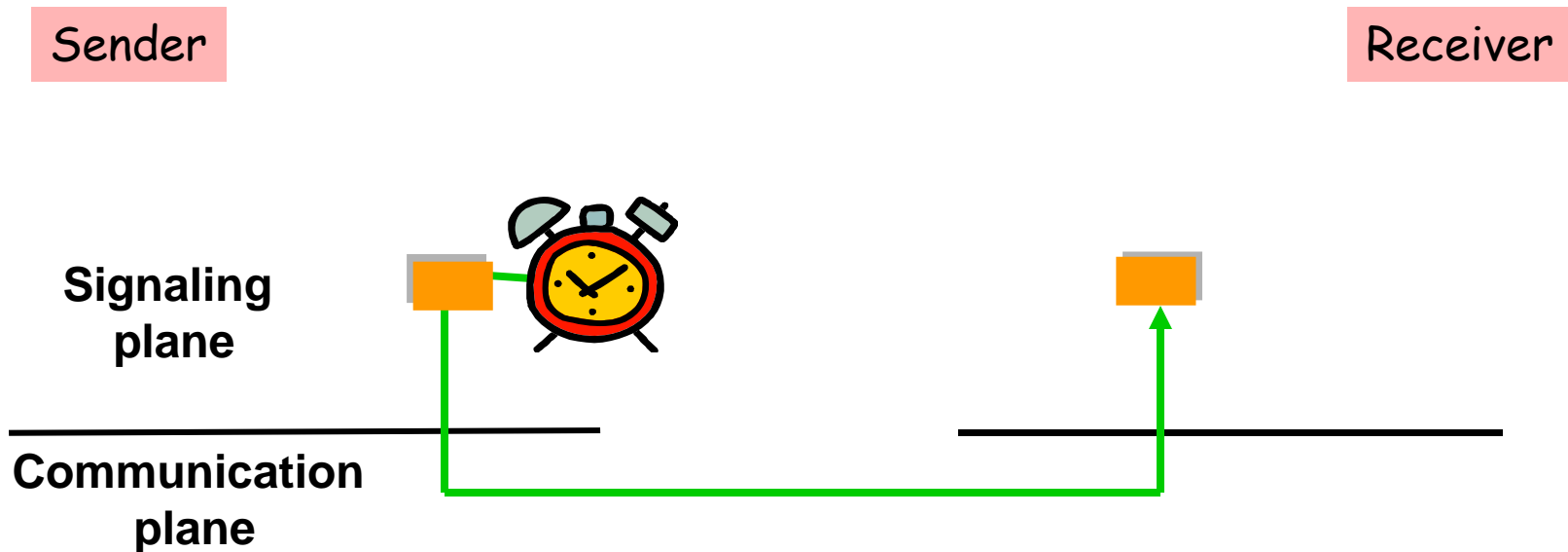
- state *installed* by receiver on receipt of *setup* (trigger) *msg* from sender (typically, an endpoint)
 - sender also sends periodic *refresh msg*: indicating receiver should continue to maintain state
- state *removed* by receiver via timeout, in absence of refresh msg from sender
- default assumption: state becomes invalid unless refreshed
 - in practice: explicit state removal (*teardown*) msgs also used
- examples:
 - RSVP, RTP, IGMP

Soft-state signaling



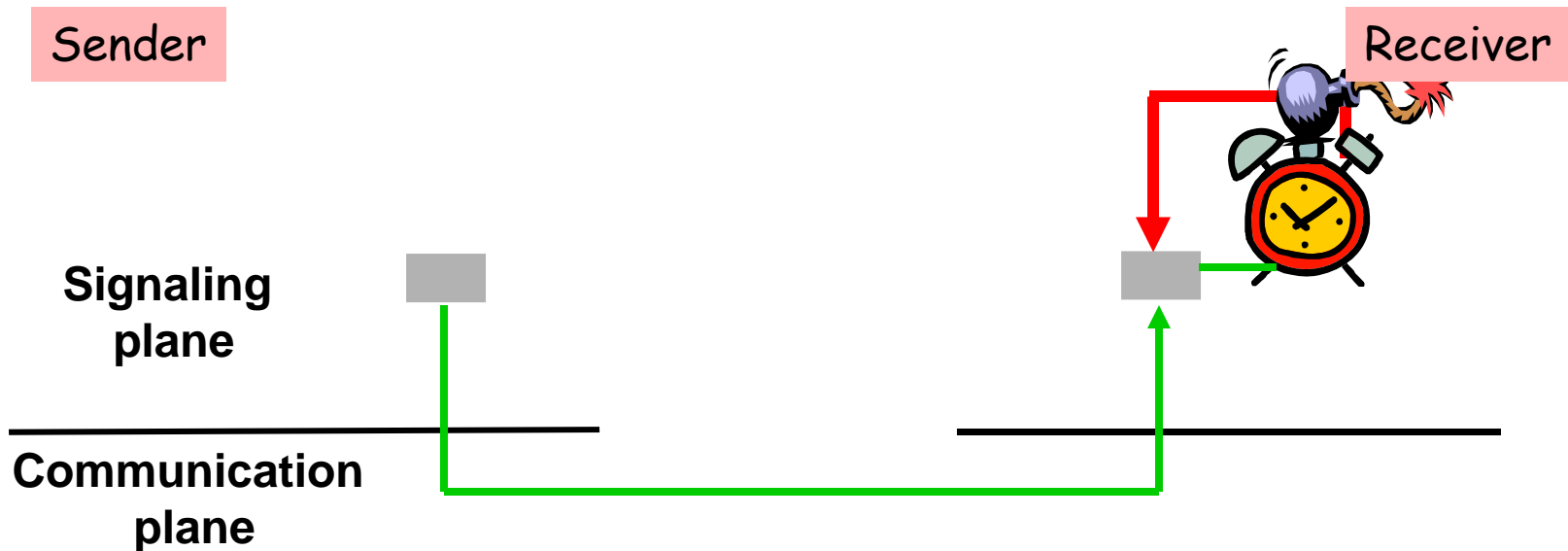
- best effort signaling

Soft-state signaling



- best effort signaling
- refresh timer, periodic refresh

Soft-state signaling



- best effort signaling
- refresh timer, periodic refresh
- state time-out timer, state removal only by time-out

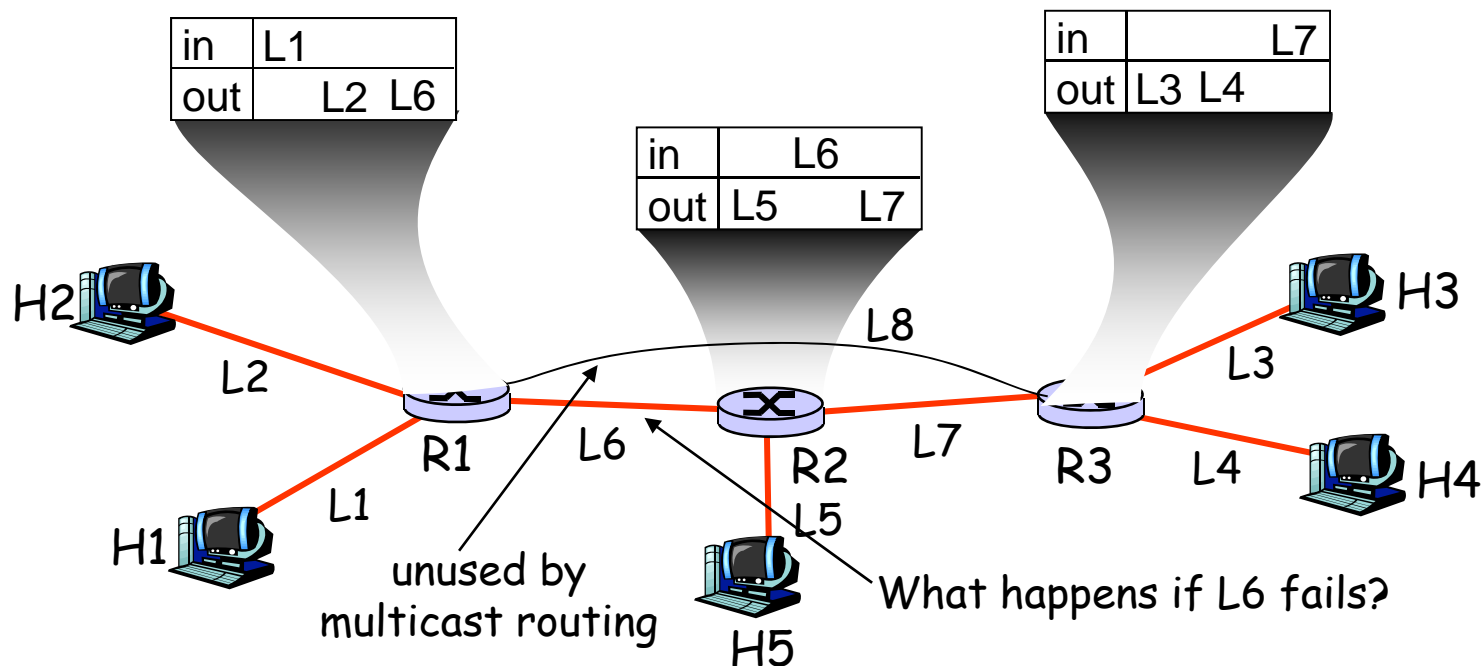
Soft-state: claims

- "Systems built on soft-state are robust"
[Raman 99]
- "Soft-state protocols provide .. greater robustness to changes in the underlying network conditions..." [Sharma 97]
- "obviates the need for complex error handling software" [Balakrishnan 99]

What does this mean?

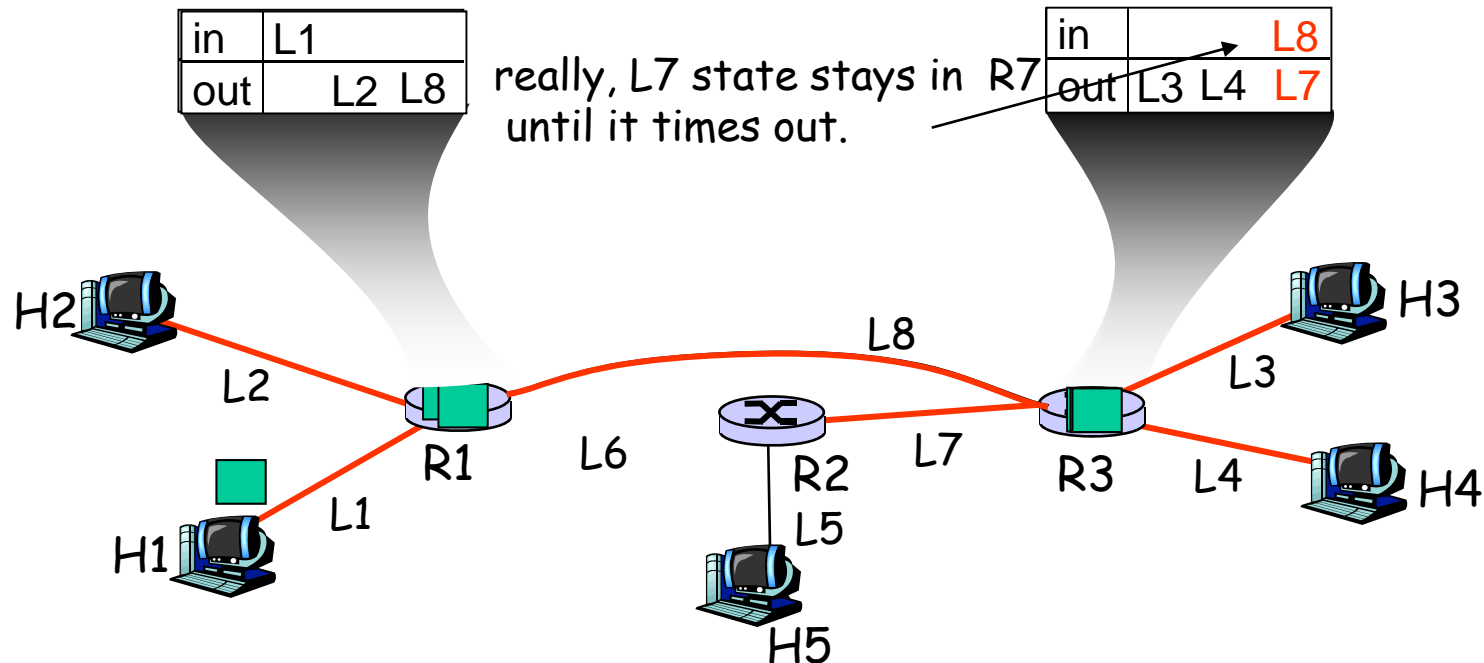
Soft-state: "easy" handling of changes

- ❑ **Periodic refresh:** if network "conditions" change, refresh will re-establish state under new conditions
- ❑ example: RSVP/routing interaction: if routes change (nodes fail) RSVP PATH refresh will *re-establish* state along new path



Soft-state: "easy" handling of changes

- ❑ L6 goes down, multicast routing reconfigures but...
- ❑ H1 data no longer reaches H3, H4, H5 (no sender or receiver state for L8)
- ❑ H1 refreshes PATH, establishes *new* state for L8 in R1, R3
- ❑ H4 refreshes RESV, propagates upstream to H1, establishes new receiver state for H4 in R1, R3



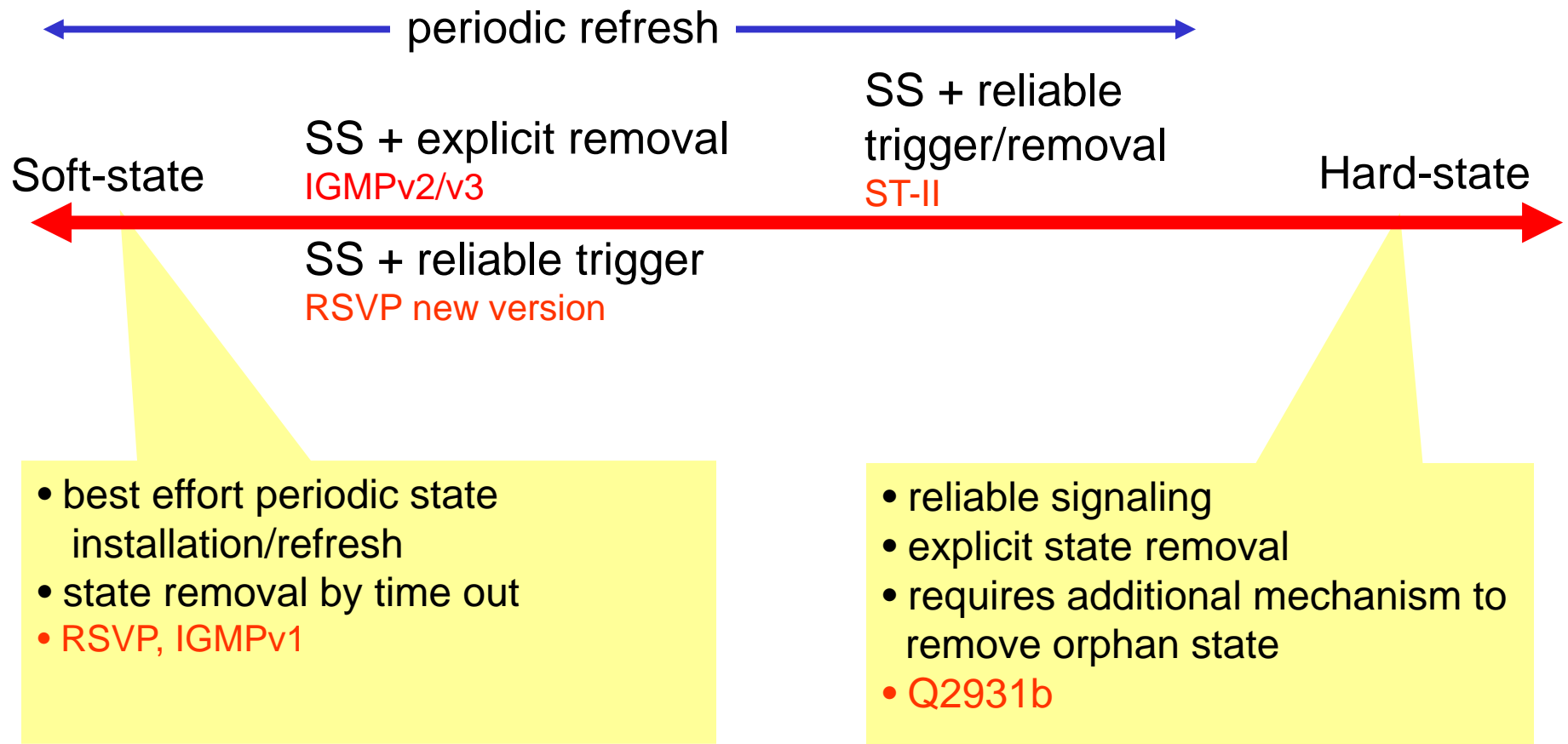
Soft-state: "easy" handling of changes

- ❑ "recovery" performed transparently to end-system by normal refresh procedures
- ❑ no need for network to signal failure/change to end system, or end system to respond to specific error
- ❑ less signaling (volume, types of messages) than hard-state from network to end-system but...
- ❑ more signaling (volume) than hard-state from end-system to network for refreshes

Soft-state: refreshes

- refresh msgs serve many purposes:
 - **trigger**: first time state-installation
 - **refresh**: refresh state known to exist ("I am still here")
 - <lack of refresh>: remove state ("I am gone")
- challenge: all refresh msgs unreliable
 - would like triggers to result in state-installation asap
 - enhancement: add receiver-to-sender refresh_ACK for triggers
 - e.g., see "Staged Refresh Timers for RSVP"

Signaling spectrum



How do we model/evaluate?

Metrics

- ❑ **inconsistency ratio** - fraction time participating nodes disagree
- ❑ **signaling overhead** - average # of messages during session lifetime

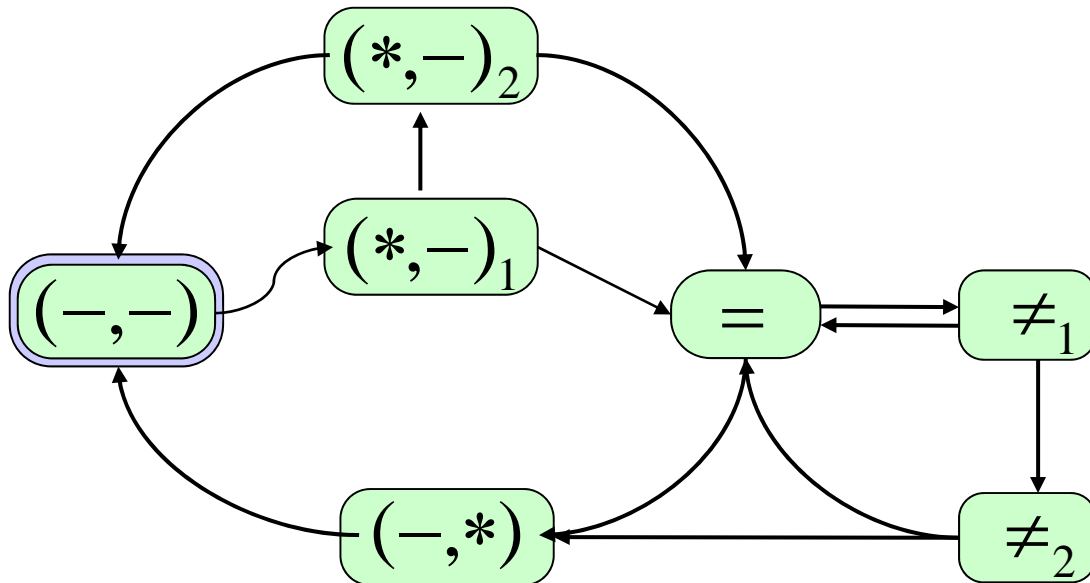
- ❑ measures of robustness?
 - convergence time
- ❑ complexity?

Single hop model

- sender, receiver
- single state variable
- state lifetime, $\exp(\mu)$
- updates - Poisson(λ)
- timers - exponentially distributed
 - refresh - $1/T$
 - state expiration - $1/X$
- link: delay $\exp(1/D)$,loss prob. p

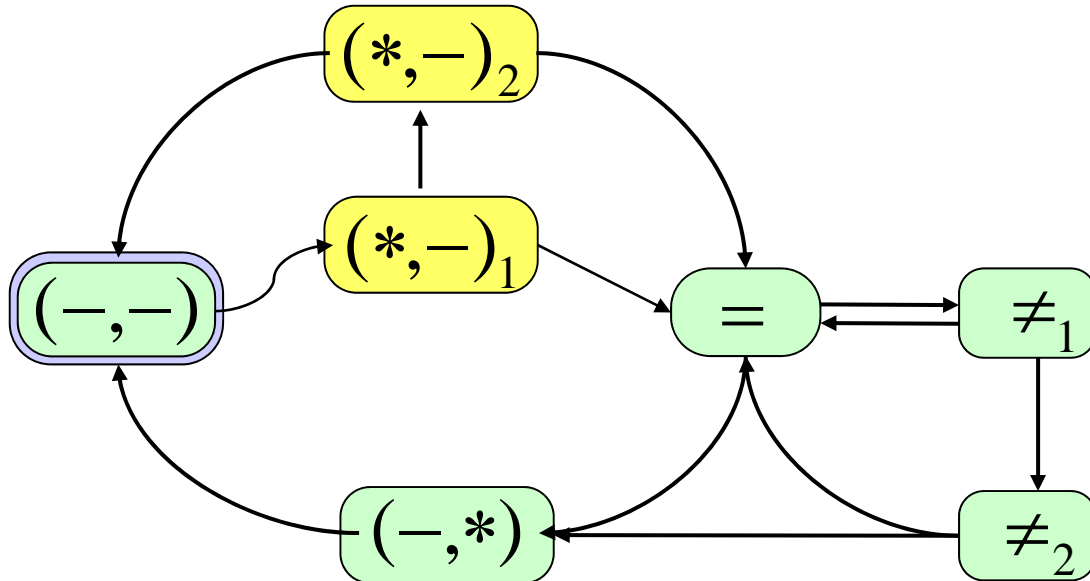


Model for SS (Ji03)



Model for SS

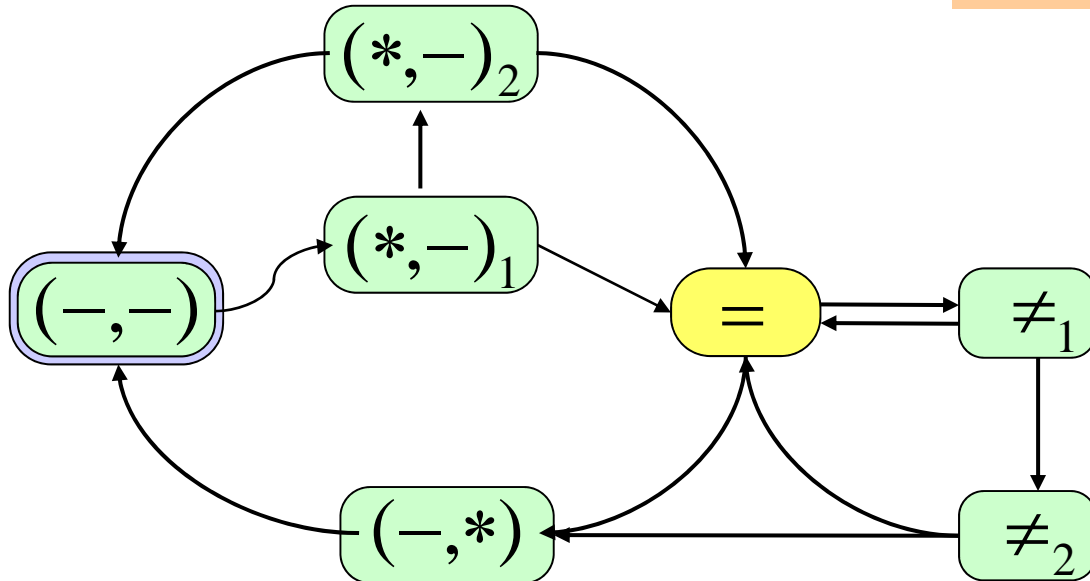
$(*, -)$: signaling state generated at sdr, not installed at rcvr



Model for SS

$(*, -)$ signaling state generated at sdr, not installed at rcvr

$=$: signaling state consistent at sdr/rcvr

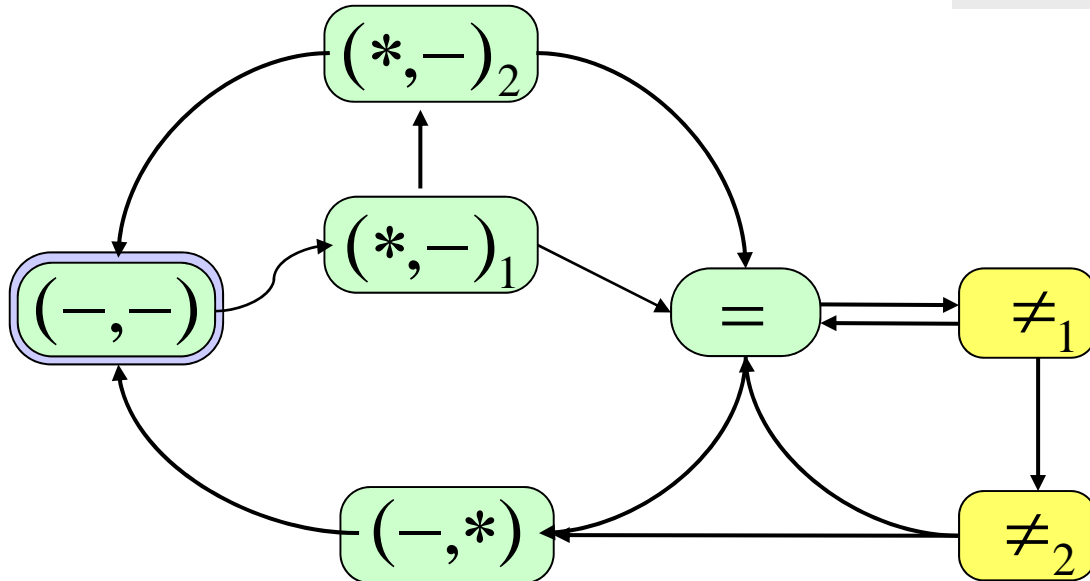


Model for SS

$(*, -)$ signaling state generated at sdr, not installed at rcvr

$=$: signaling state consistent at sdr/rcvr

\neq : signaling state inconsistent at sdr/rcvr

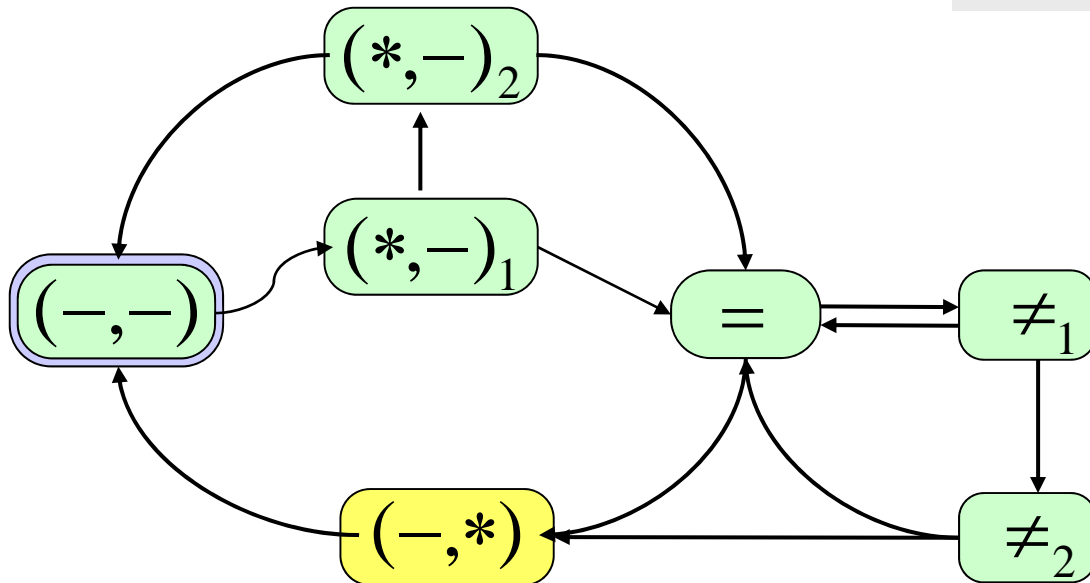


Model for SS

$(*, -)$ signaling state generated at sdr, not installed at rcvr

$=$: signaling state consistent at sdr/rcvr

\neq : signaling state inconsistent at sdr/rcvr



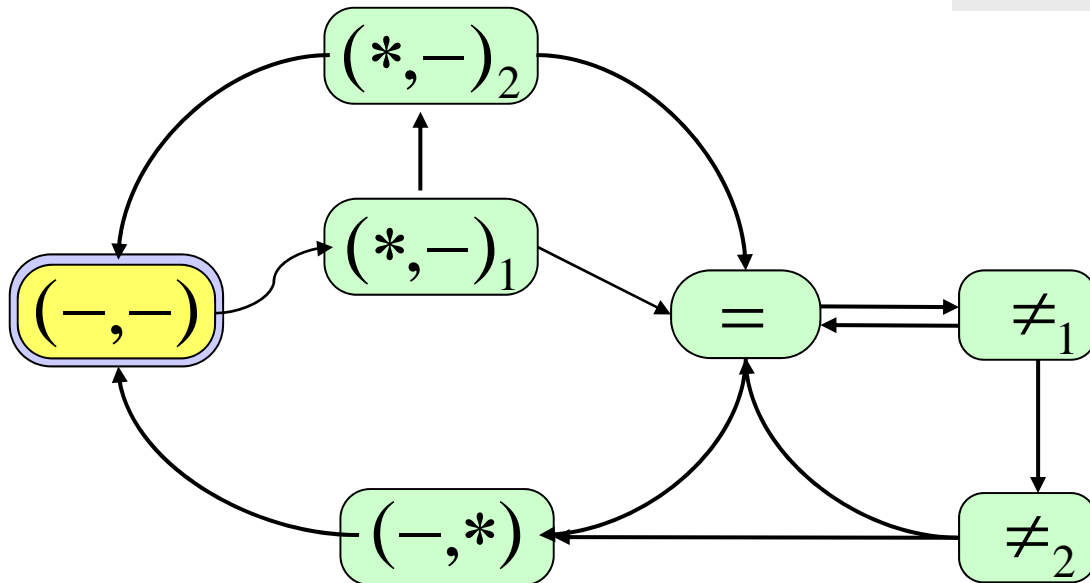
$(-, *)$: signaling state removed at sender, present at receiver

Model for SS

$(*, -)$: signaling state generated at sdr, not installed at rcvr

$=$: signaling state consistent at sdr/rcvr

\neq : signaling state inconsistent at sdr/rcvr



$(-, -)$: signaling state removed at sdr/rcvr

$(-, *)$: signaling state removed at sender, present at receiver

Model for SS

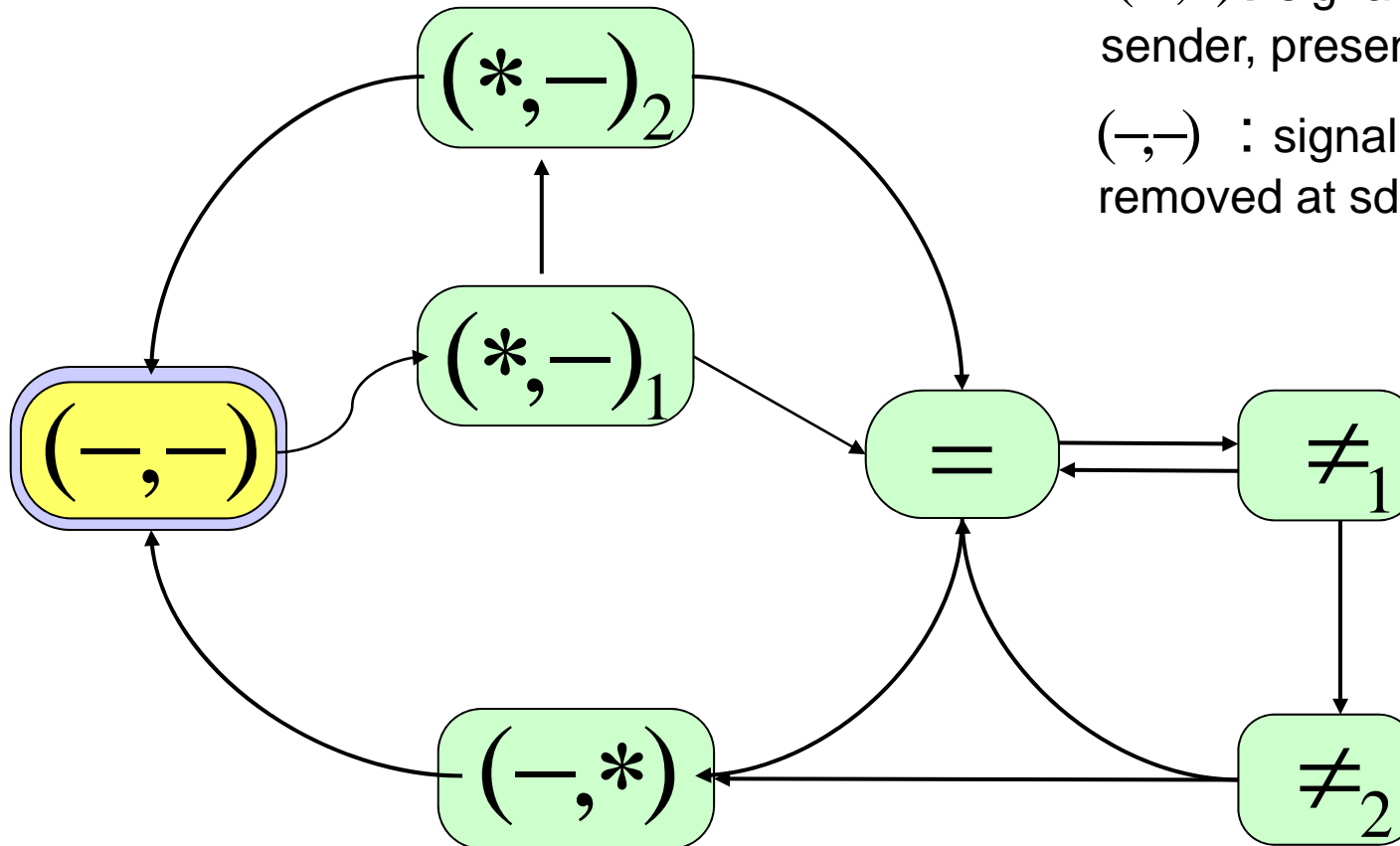
$(*, -)$: signaling state generated at sdr, not installed at rcvr

$=$: signaling state consistent at sdr/rcvr

\neq : signaling state inconsistent at sdr/rcvr

$(-, *)$: signaling state removed at sender, present at receiver

$(-, -)$: signaling state removed at sdr/rcvr



Model for SS

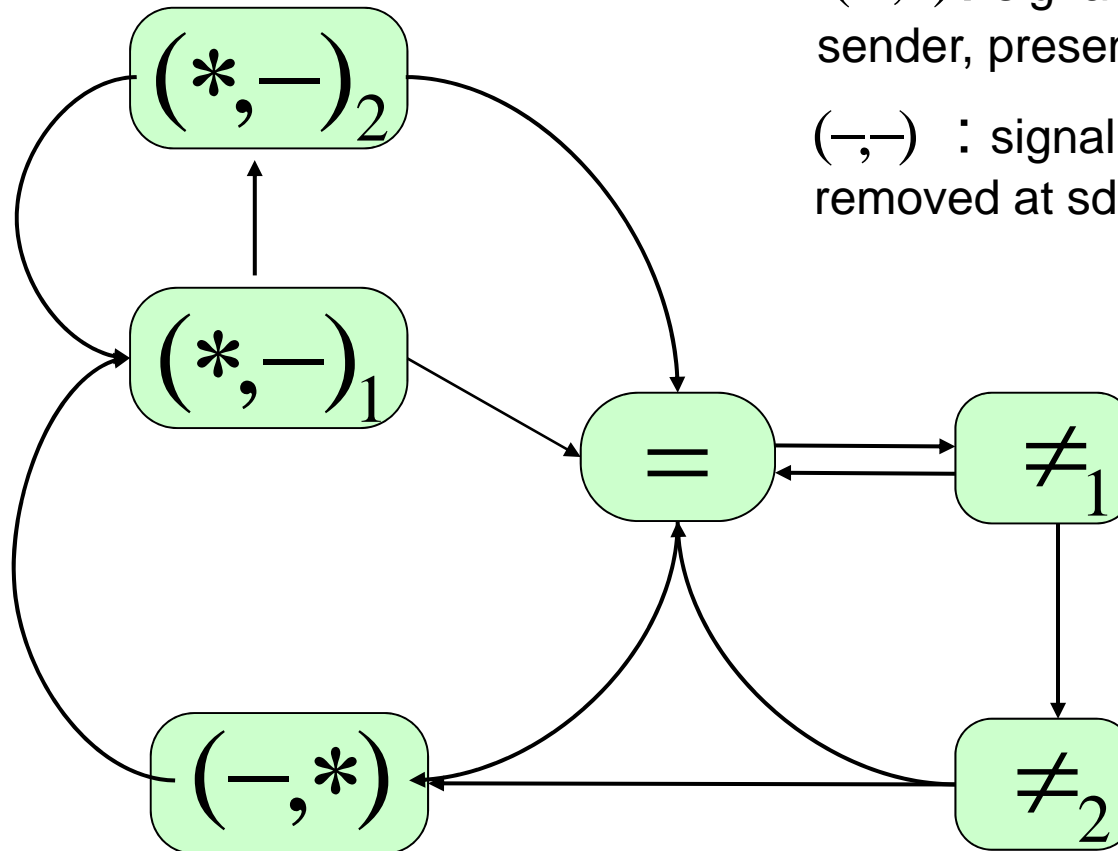
$(*, -)$: signaling state generated at sdr, not installed at rcvr

$=$: signaling state consistent at sdr/rcvr

\neq : signaling state inconsistent at sdr/rcvr

$(-, *)$: signaling state removed at sender, present at receiver

$(-, -)$: signaling state removed at sdr/rcvr



Performance metrics (SS)

□ inconsistency ratio -

$$\delta = 1 - \pi_{\neq}$$

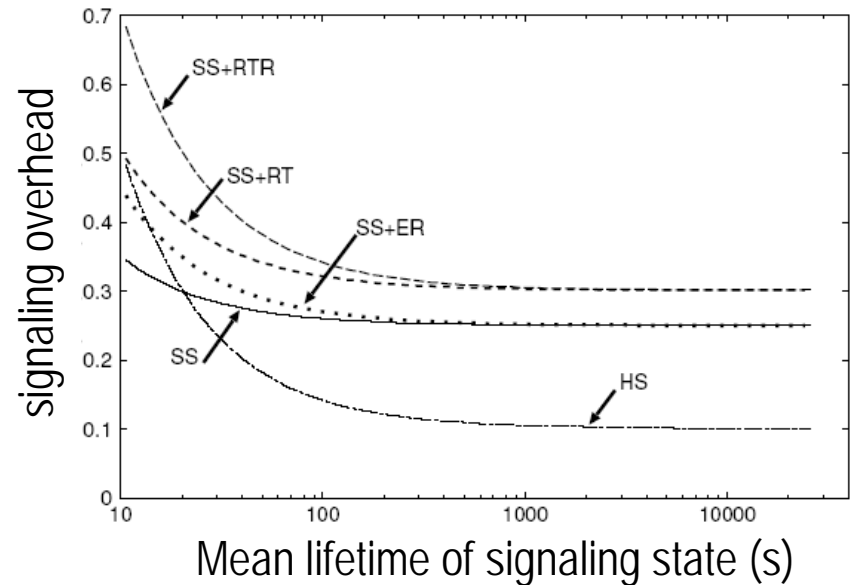
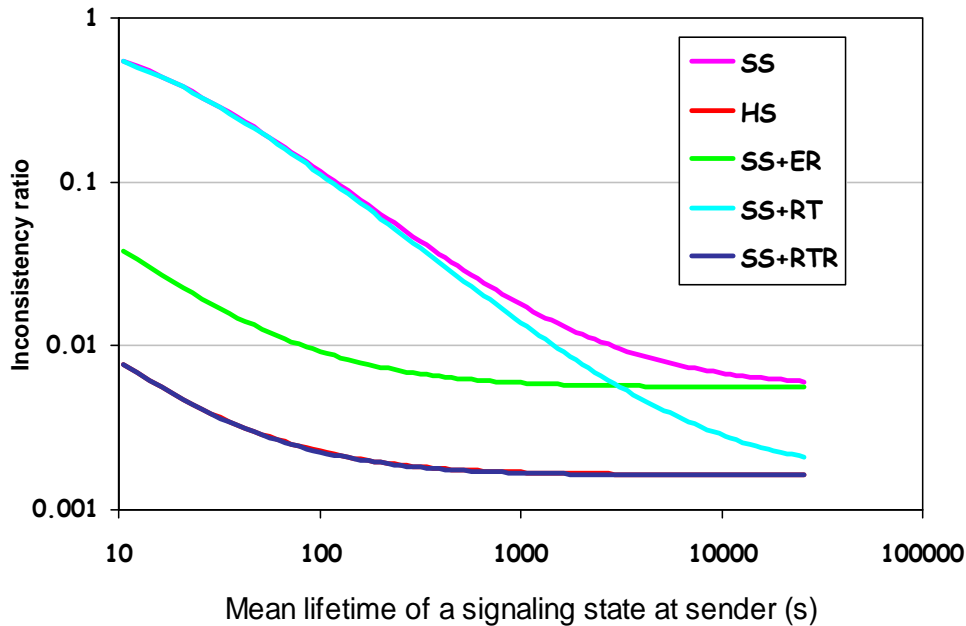
□ signaling overhead

Parameter settings

- mean lifetime - 30 min.
- refresh timer, $T=5\text{sec}$
- state timer, $X = 15 \text{ sec}$
- update rate - $1/20\text{sec}$
- signal loss rate - 2%

Motivated by Kazaa

Impact of state lifetime



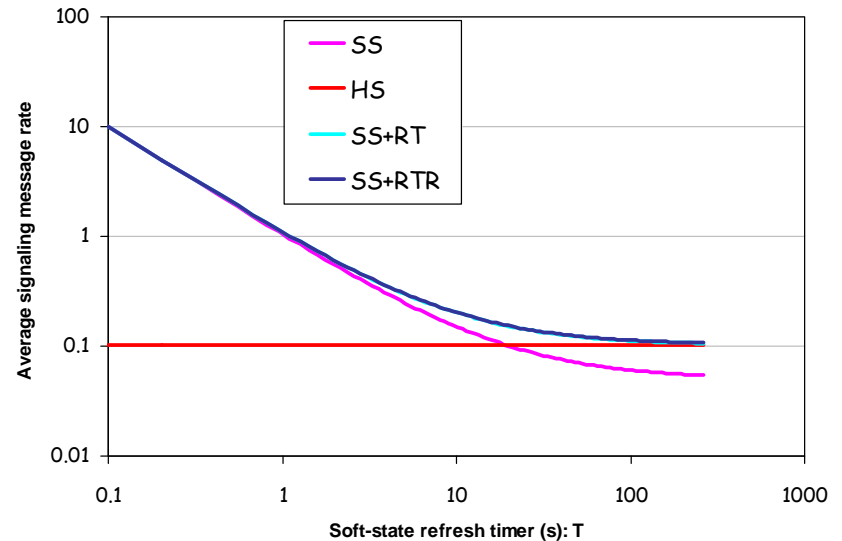
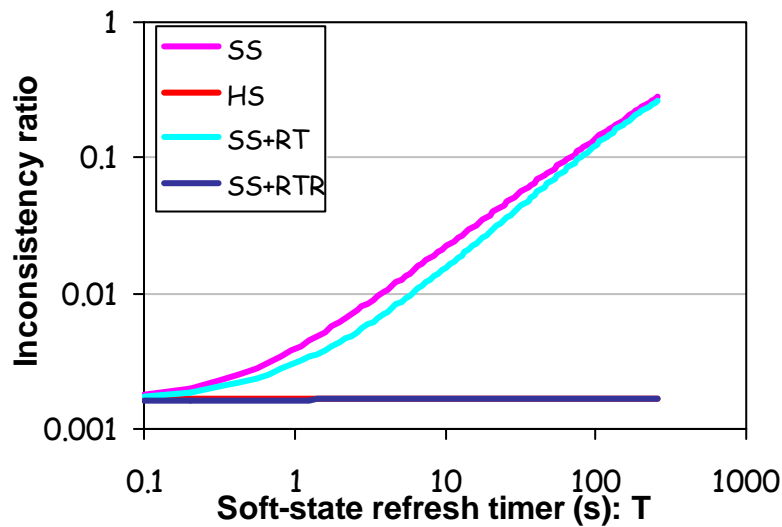
- inconsistency, overhead decrease as state lifetime increases
- explicit removal improves consistency with little additional overhead

Soft-state: setting timer values

Q: How to set refresh/timeout timers

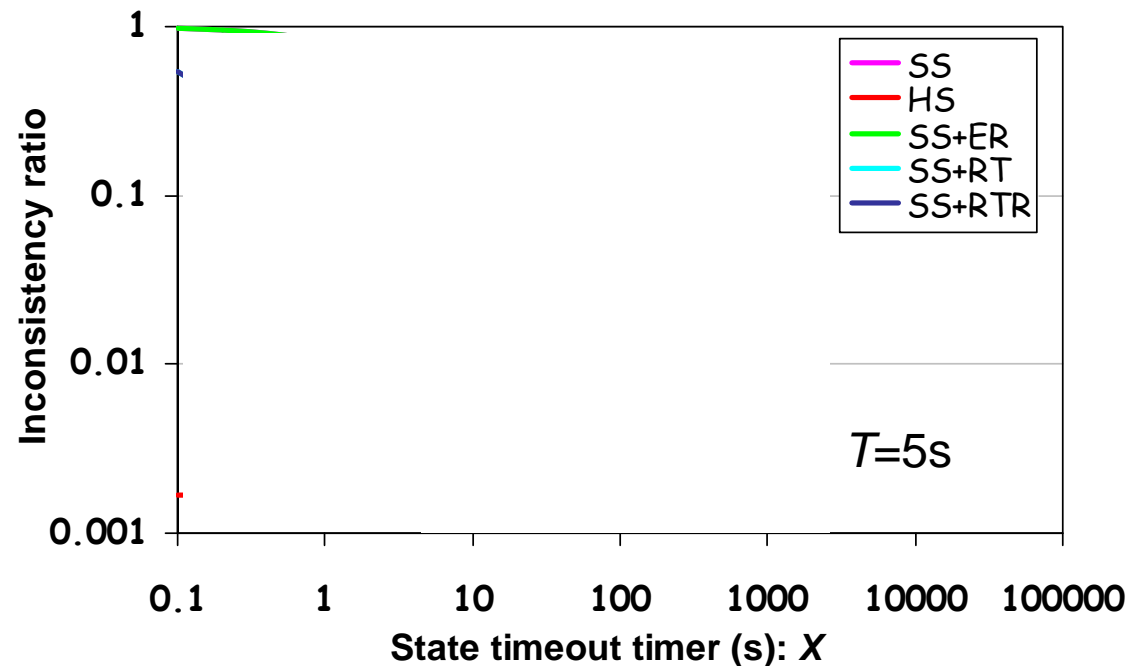
- ❑ state-timeout interval = $n * \text{refresh-interval} - \text{timeout}$
 - what value of n to choose?
- ❑ will determine amount of signaling traffic, responsiveness to change
 - small timers: fast response to changes, more signaling
 - long timers: slow response to changes, less signaling
- ❑ ultimately: consequence of slow/fast response, msg loss probability will dictate appropriate timer values

Impact of soft-state refresh timer



- ❑ as refresh timer increases
 - inconsistency increases
 - overhead decreases
- ❑ tradeoff exists for SS signaling systems

Impact of state timeout timer



- ❑ $X < T$: inconsistency high (premature state removal)
- ❑ $X > 2T$: increasing $X \Rightarrow$ increasing inconsistency for SS, SS+ER, SS+RT (due to orphan state)
- ❑ $X = 2T$: sweet spot

Hard-state versus soft-state: discussion

Q: which is preferable and why?