

Progress by the SON of SELFMAN

- ZIB

- Replication scheme for solving the inconsistent routing problem in Chord
- Multi-dimensional range queries, e.g. for geographic information systems.

- KTH

- Self-healing property in the scope of network partitioning (detection and correction)
- New component based architecture and implementation of DKS with atomic join/leave

Progress by the SON of SELFMAN

- UCL
 - New design and implementation of P2PS providing a fault tolerant relaxed ring with consistent lookup
 - Log viewer and simulator
- KTH-UCL
 - Gathering ideas on pseudo-**reliable** broadcast as actuator component in the feedback loop architecture

Project questions

- Who's the responsible of Work package 1?
“Structured overlay network and basic mechanisms”
 - DoW says KTH
 - Wiki says “the guy who is talking”
- How many PM has ZIB in WP1?
 - DoW says 0
- Deliverable D1.1: **“Report on low-level self-management primitives for structured overlay networks”**

The New P2PS architecture

T3: Reliable message sending + Pseudo-reliable broadcast

T2: Unreliable message sending + Routing

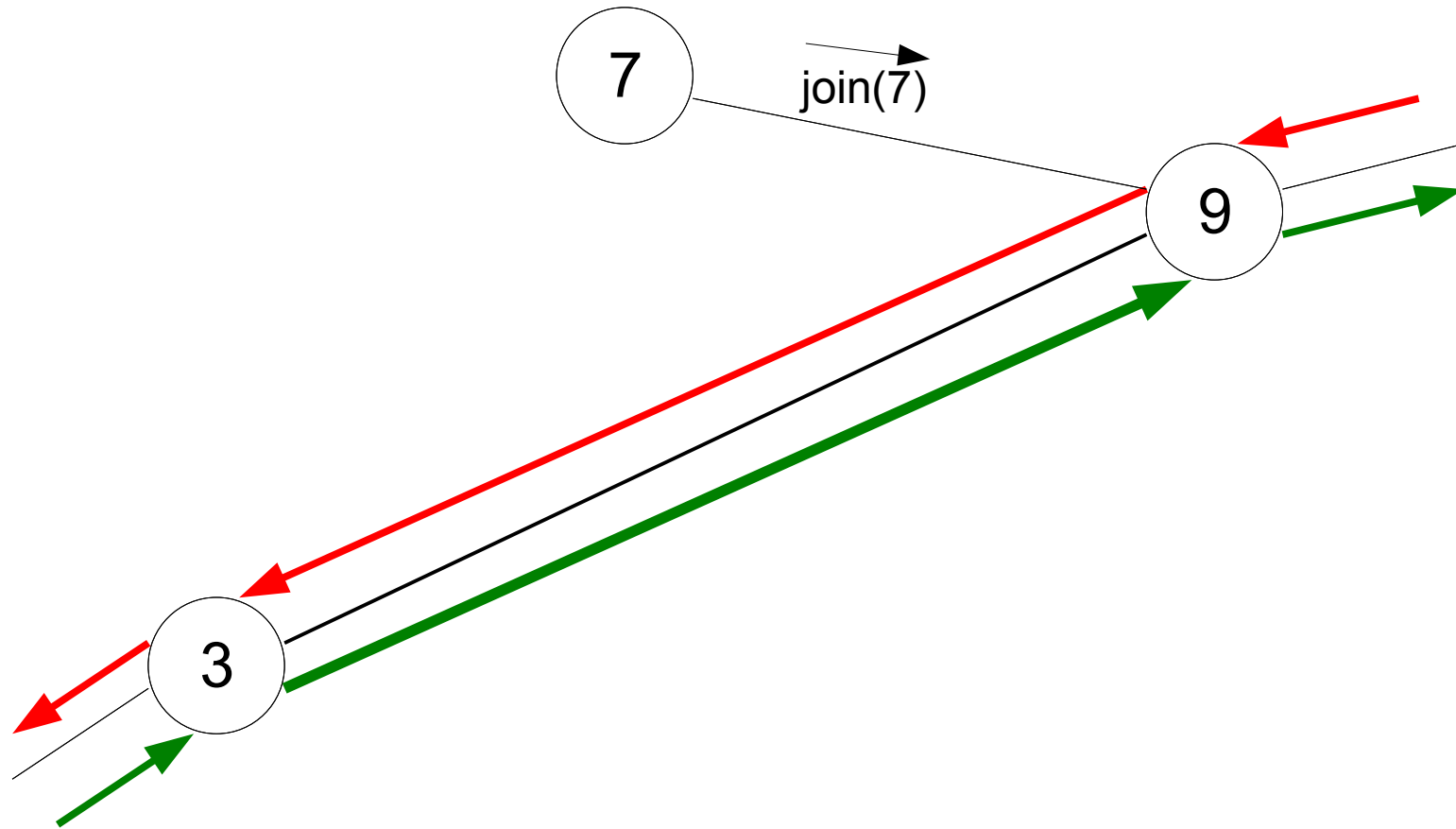
T1.5: Ring pruning

T1: Fault tolerant Relaxed Ring

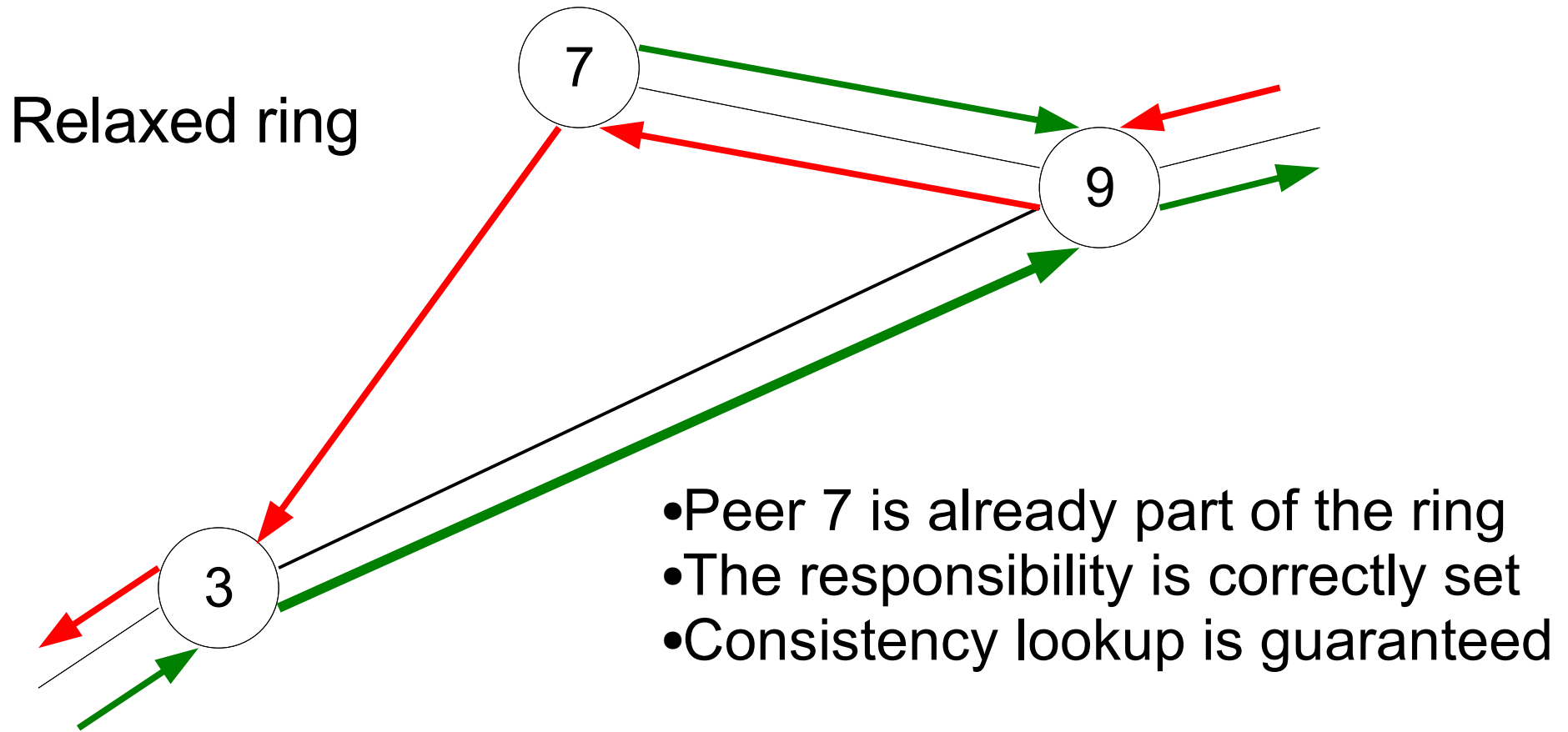
T0.5: Multi-session Component

T0: Session between 2 nodes

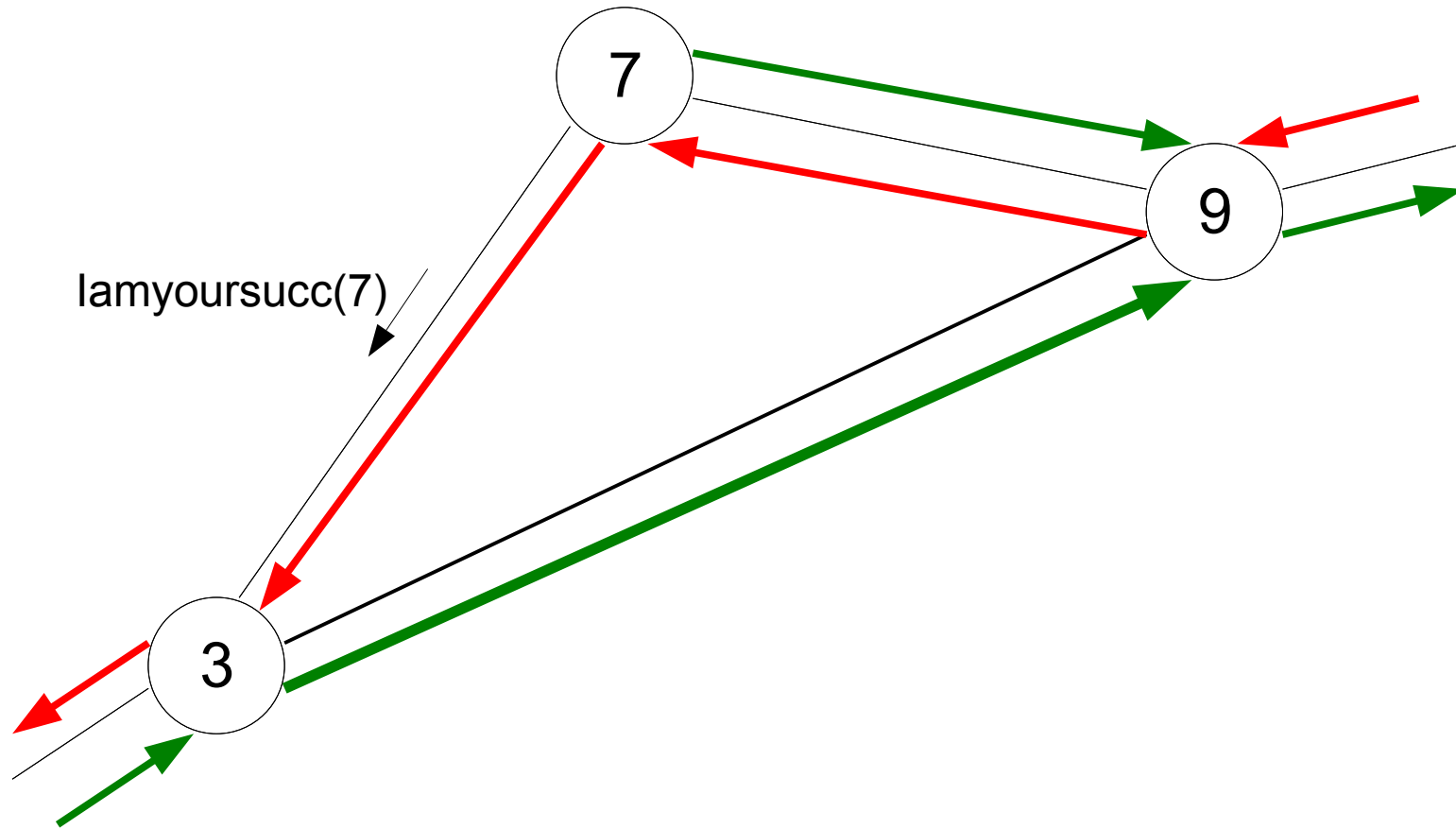
The joining peer



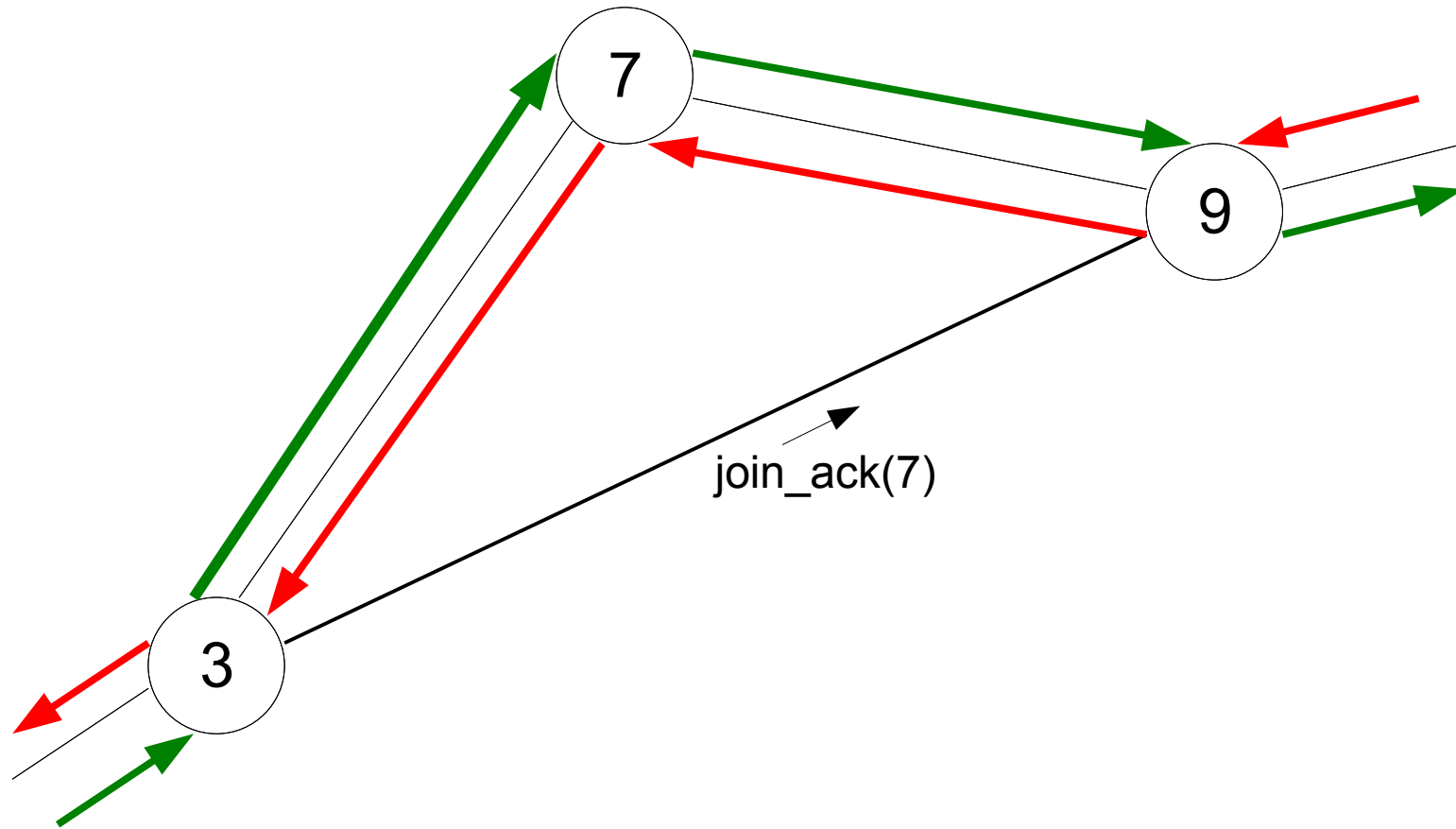
The joining peer



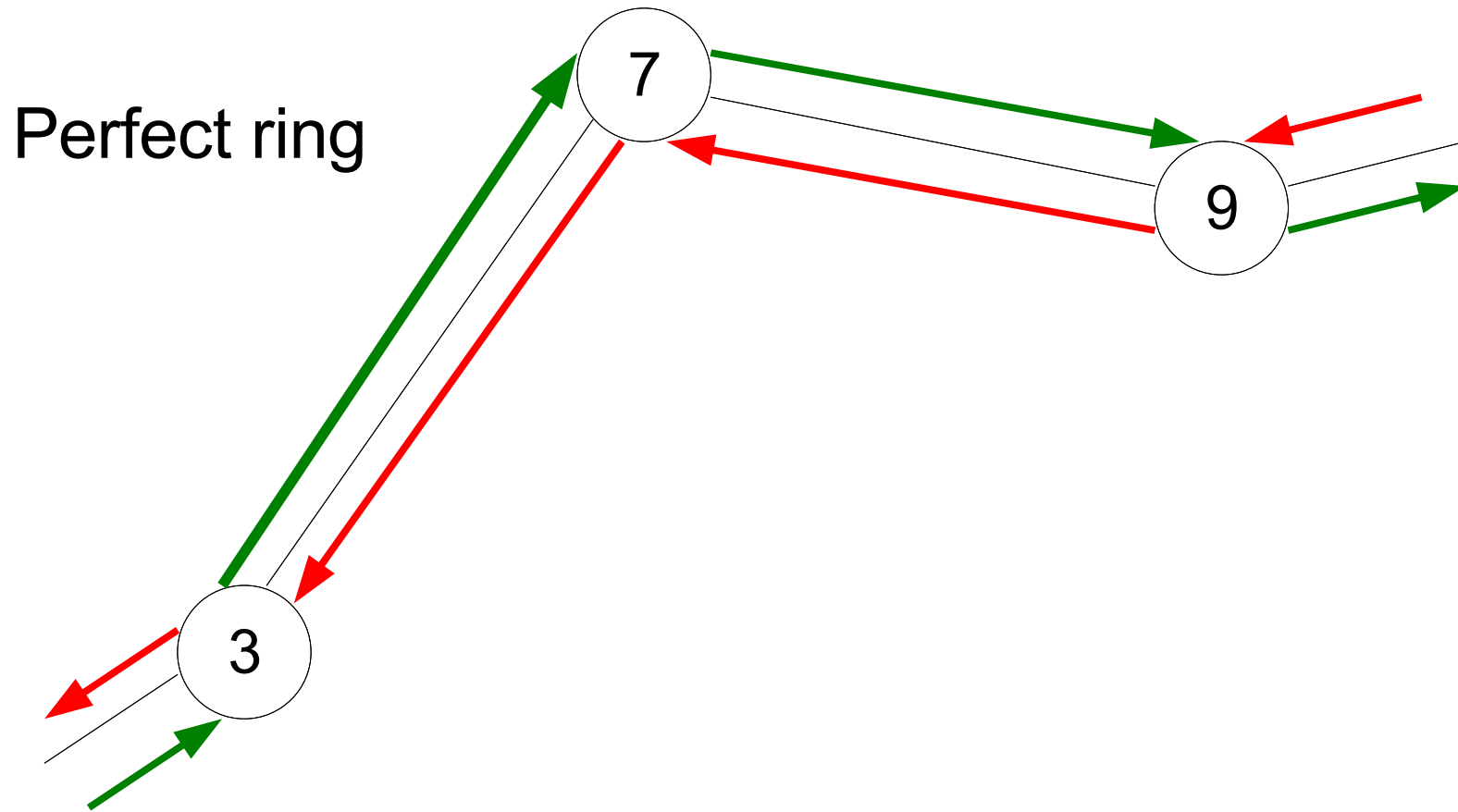
The joining peer



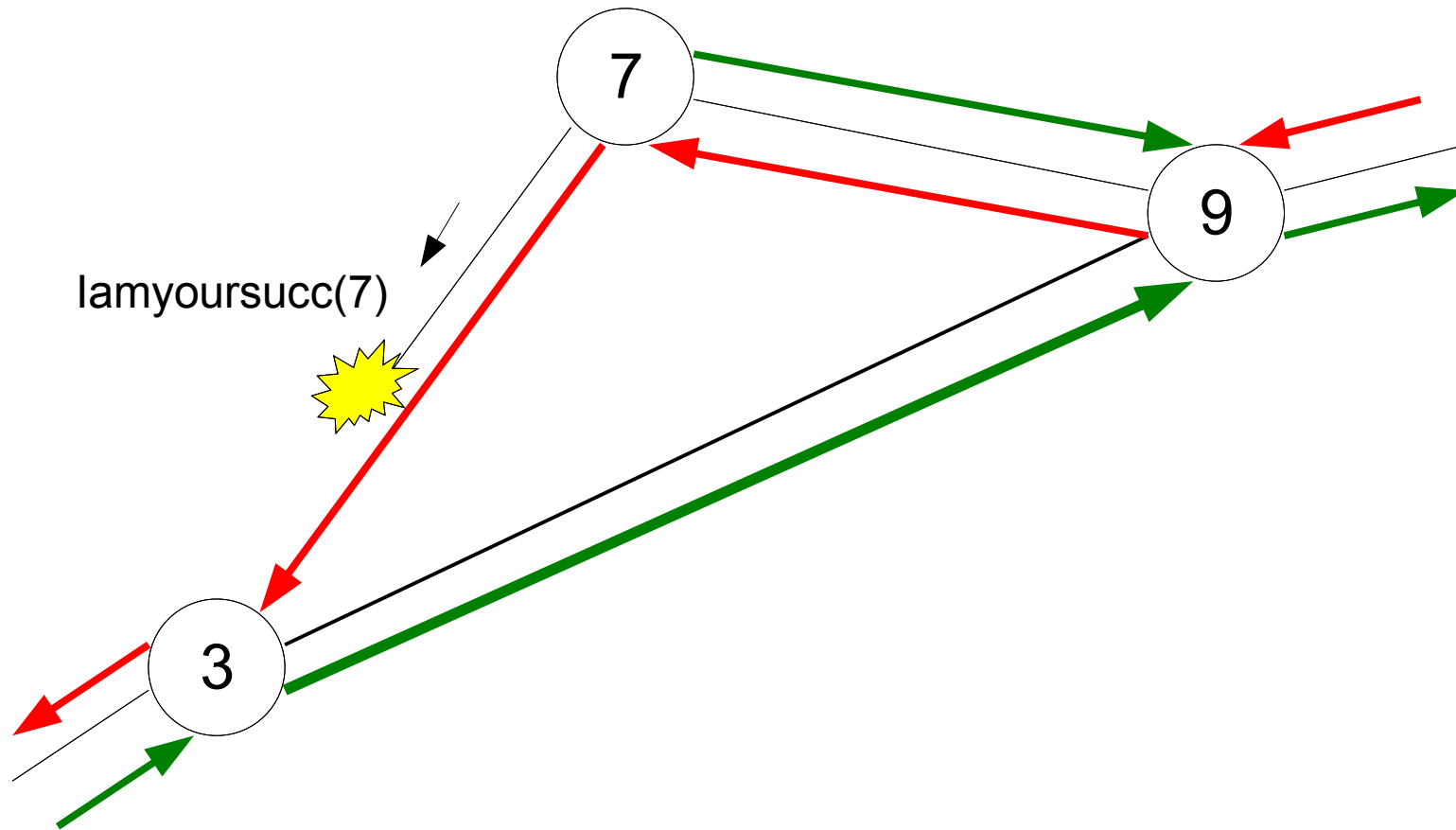
The joining peer



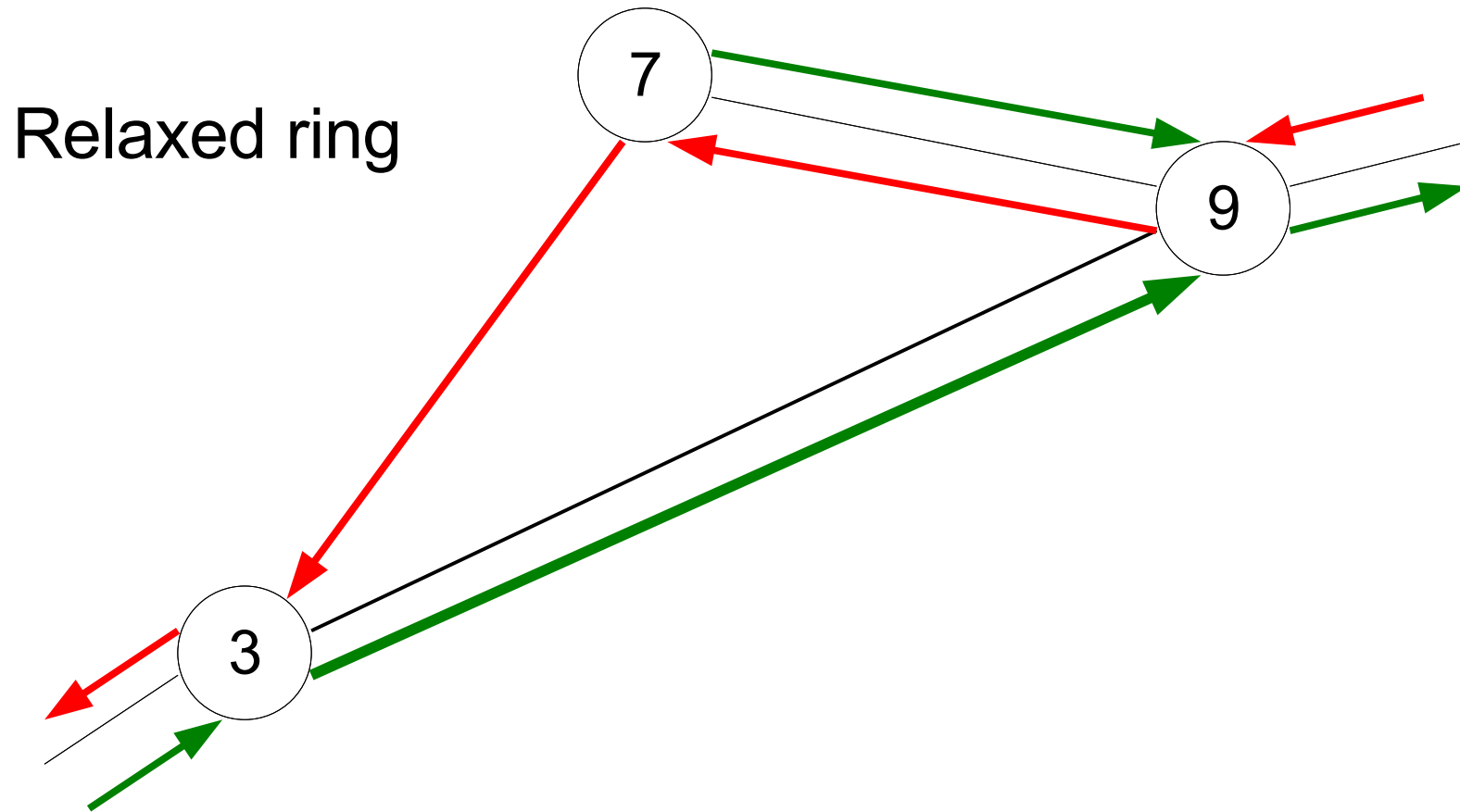
The joining peer



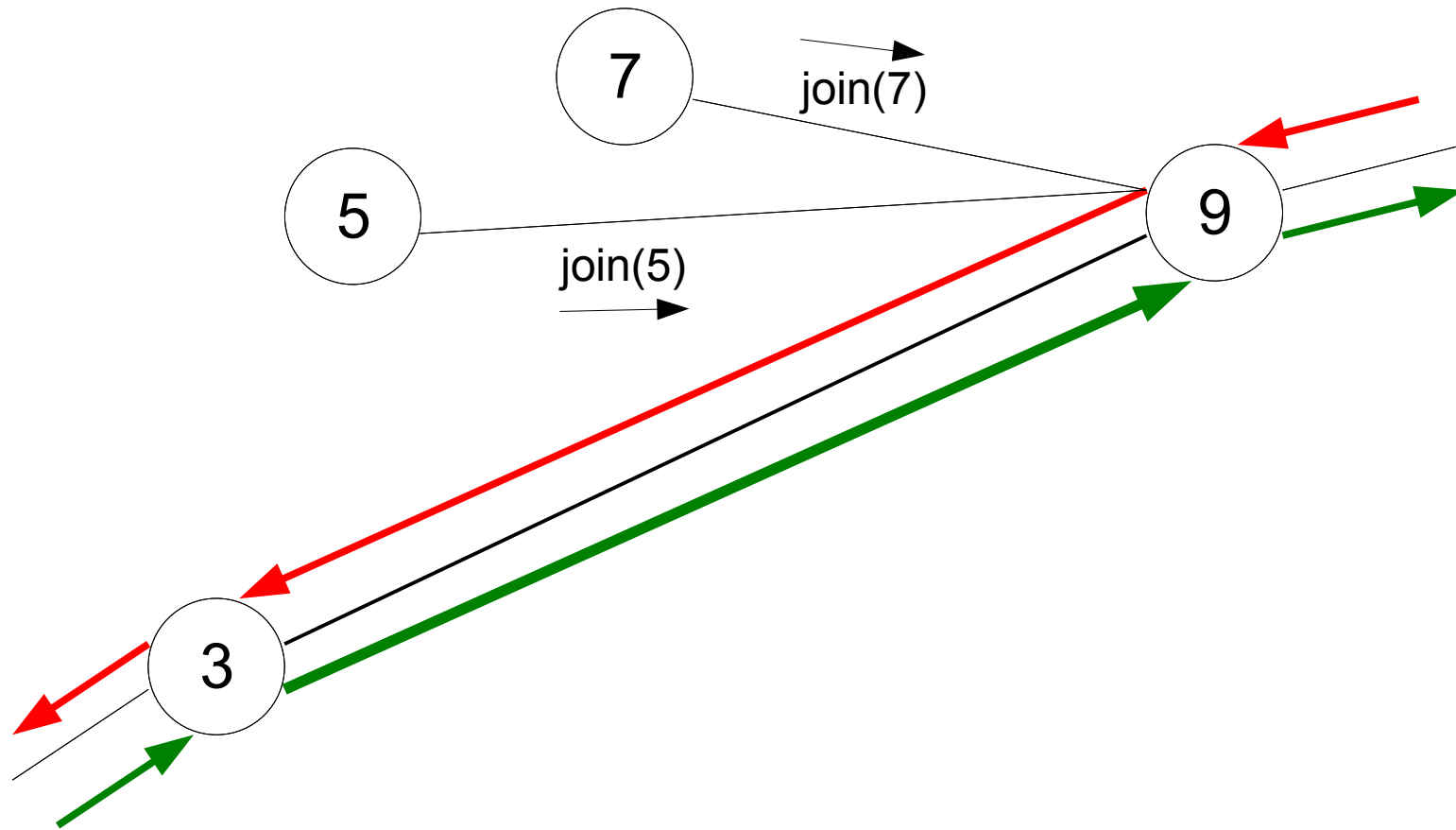
The joining peer



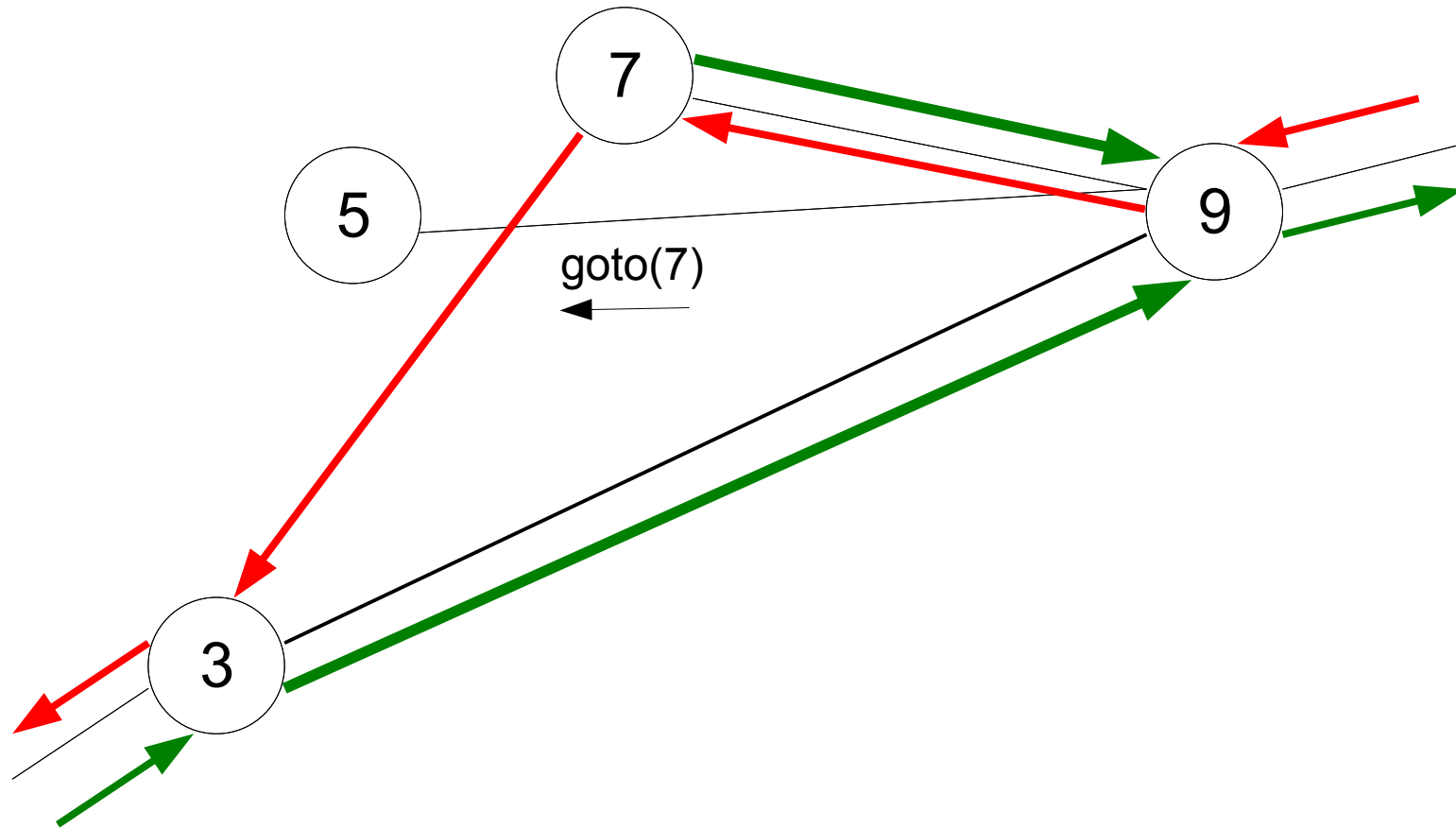
The joining peer



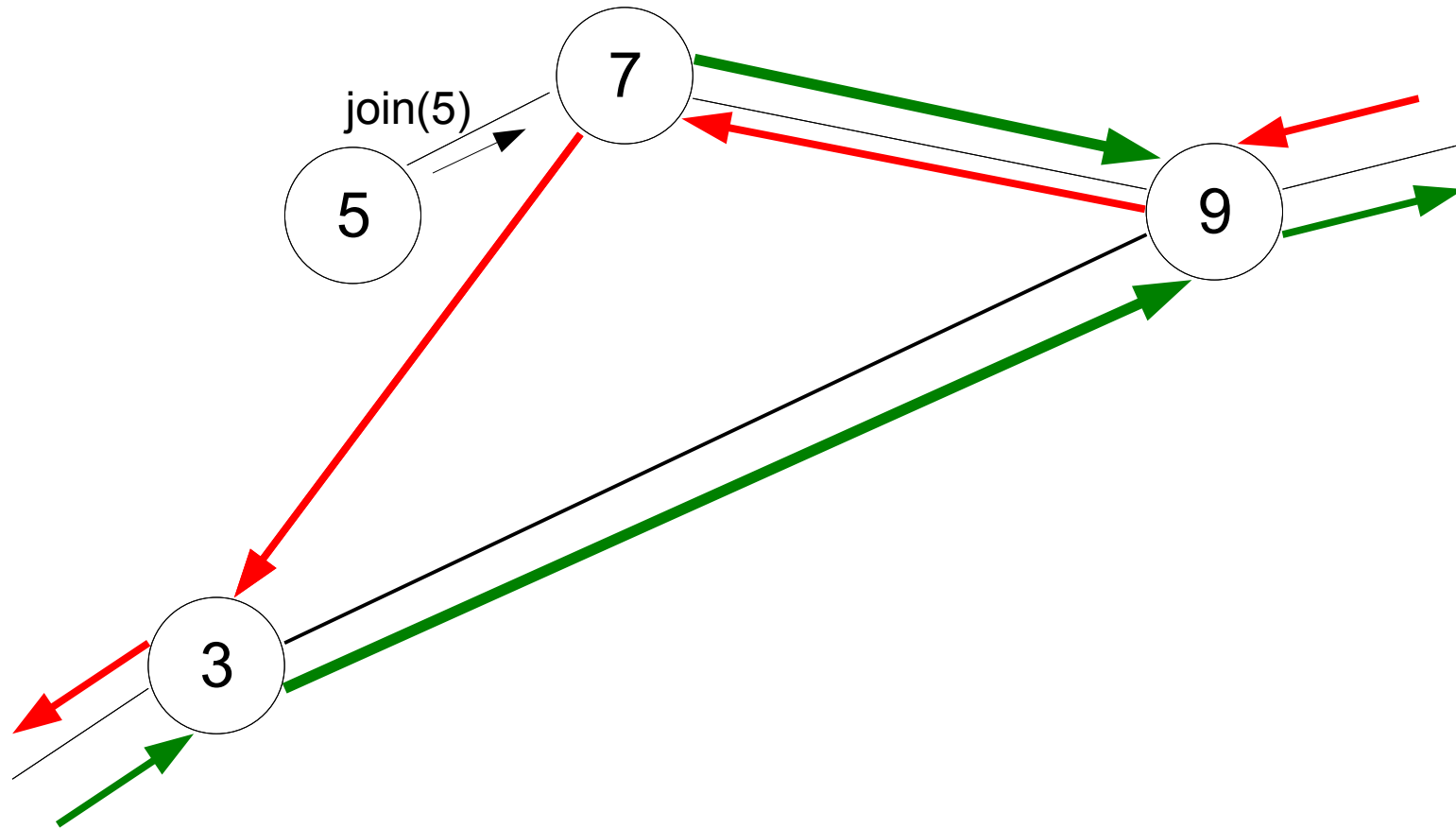
The joining peers



The joining peers

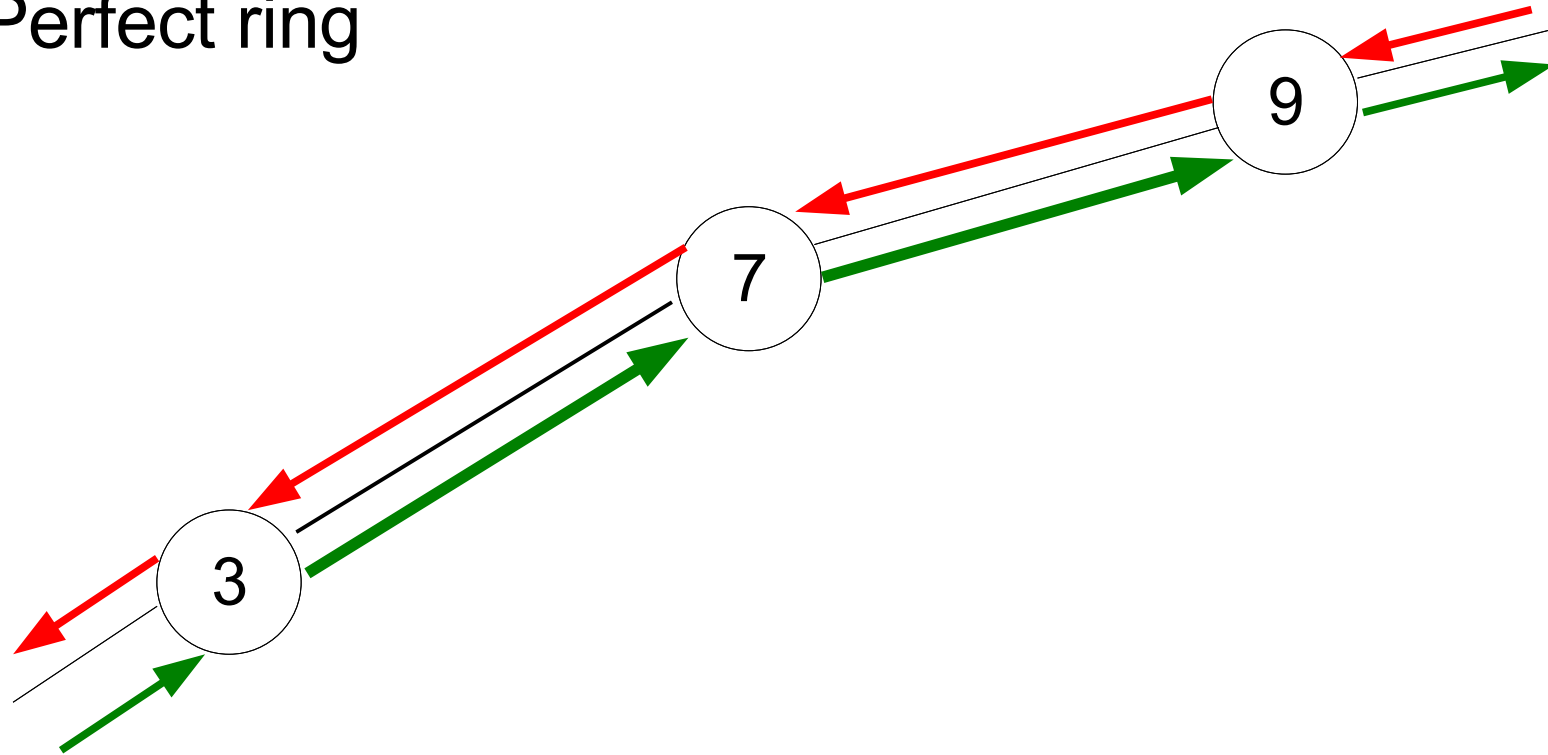


The joining peer



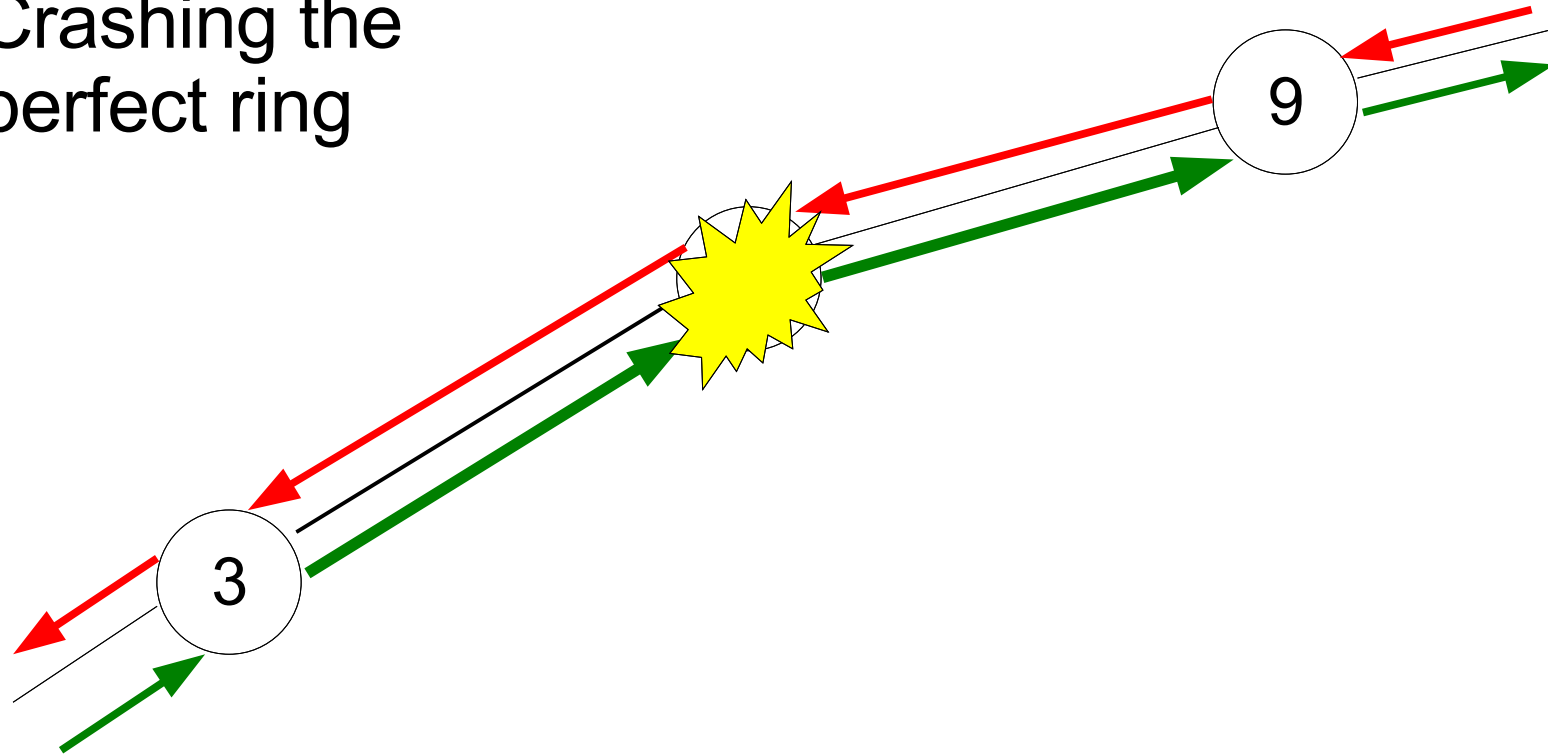
The crashing peer

Perfect ring



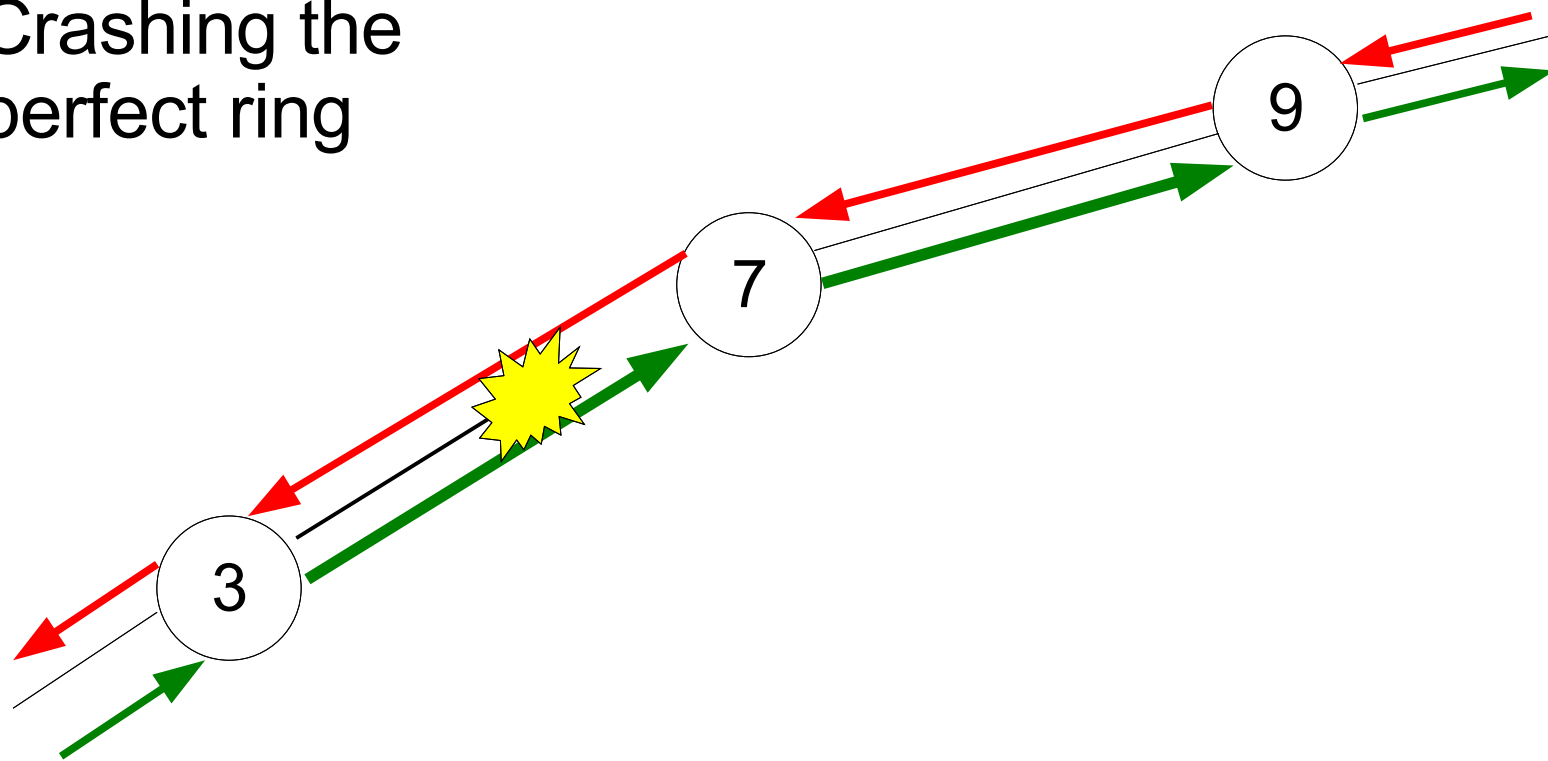
The crashing peer

Crashing the perfect ring



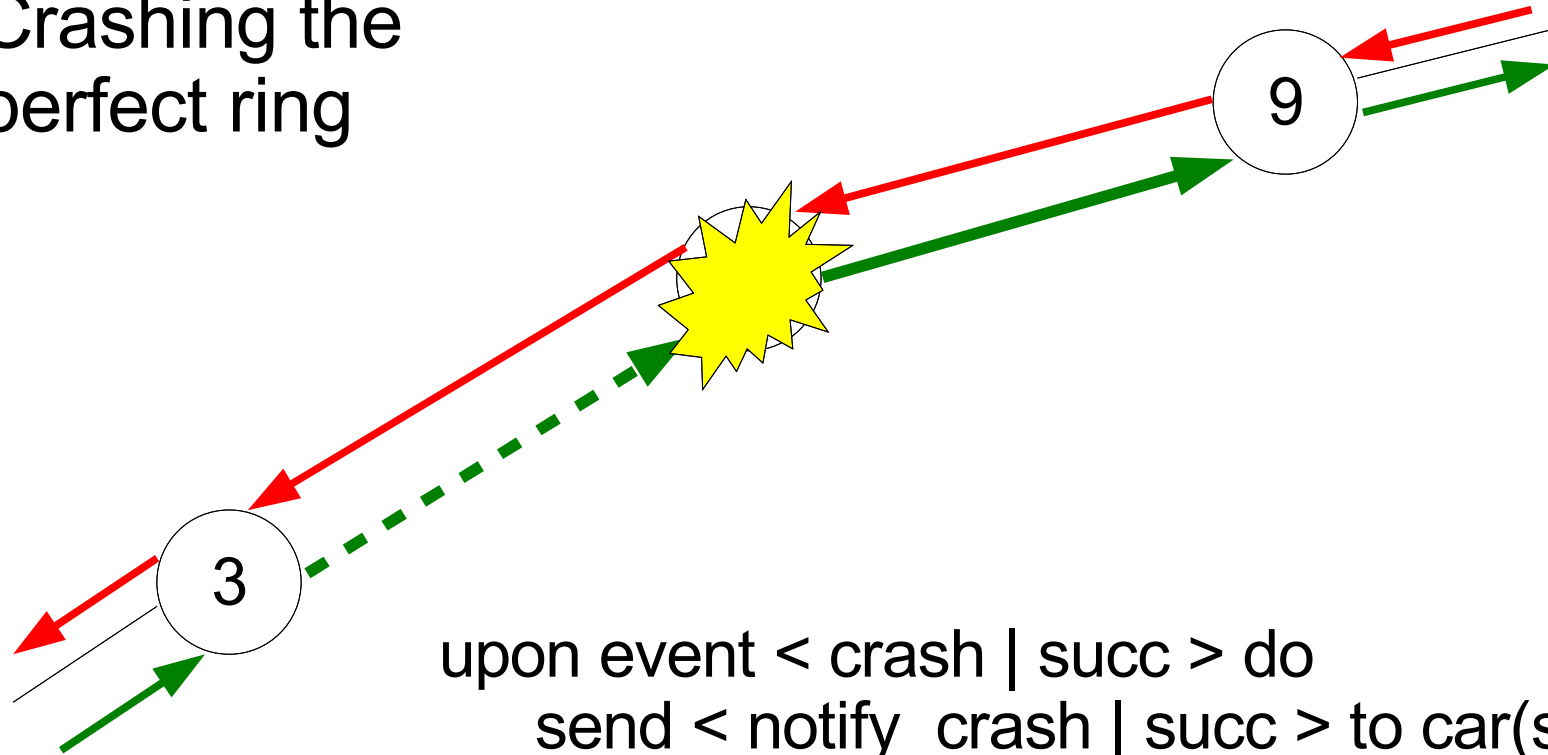
The crashing link

Crashing the perfect ring



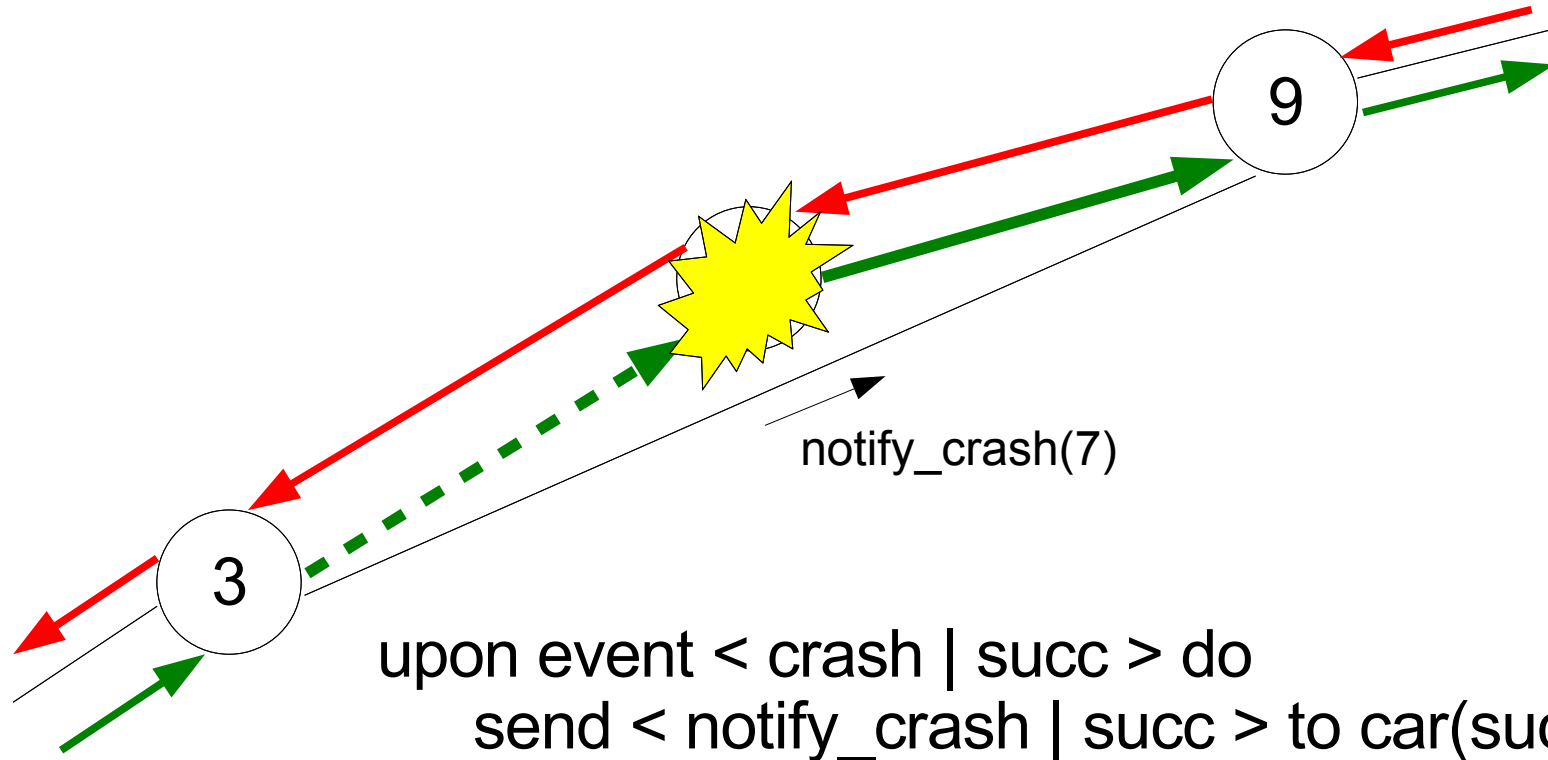
The crashing peer

Crashing the
perfect ring



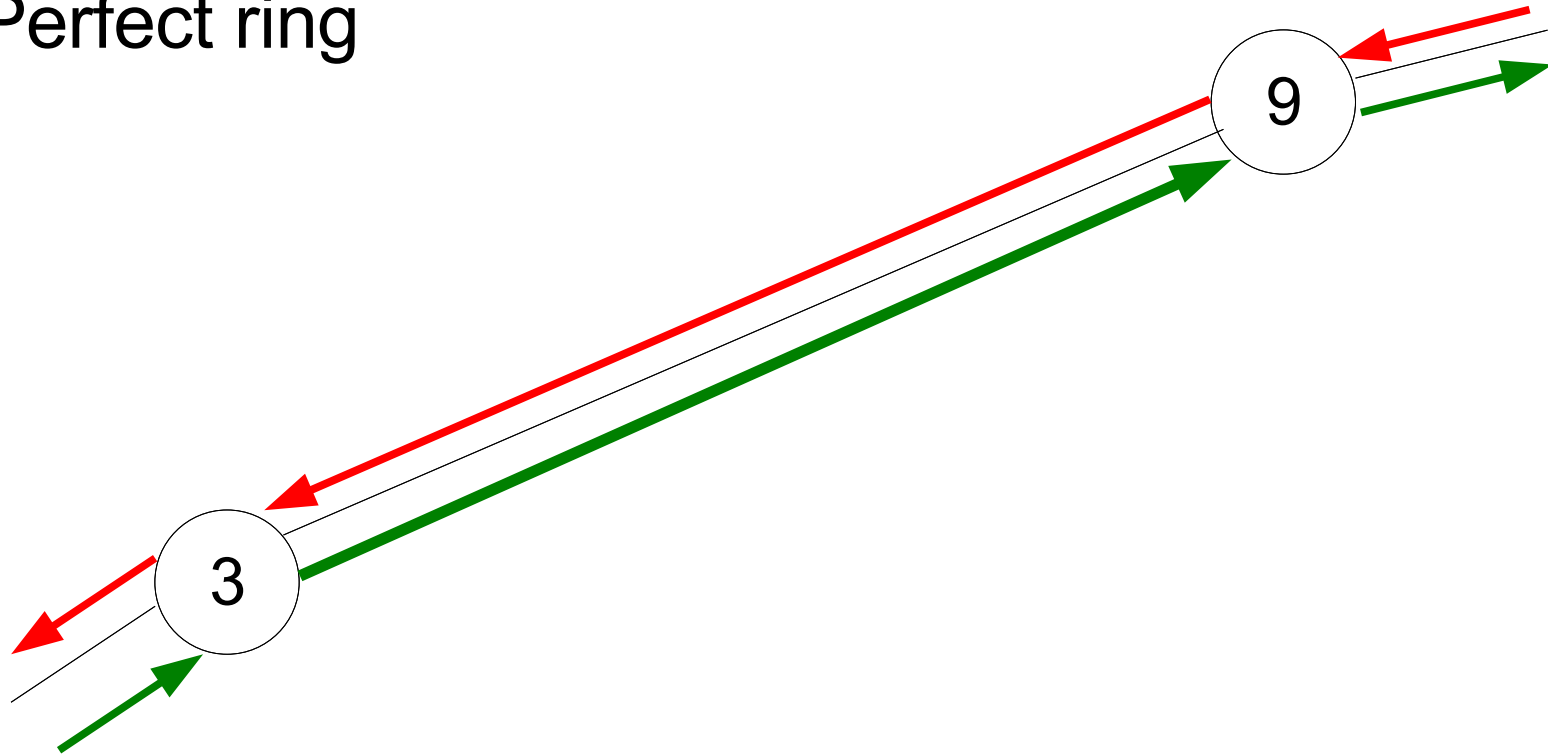
```
upon event < crash | succ > do  
  send < notify_crash | succ > to car(succList)
```

The crashing peer

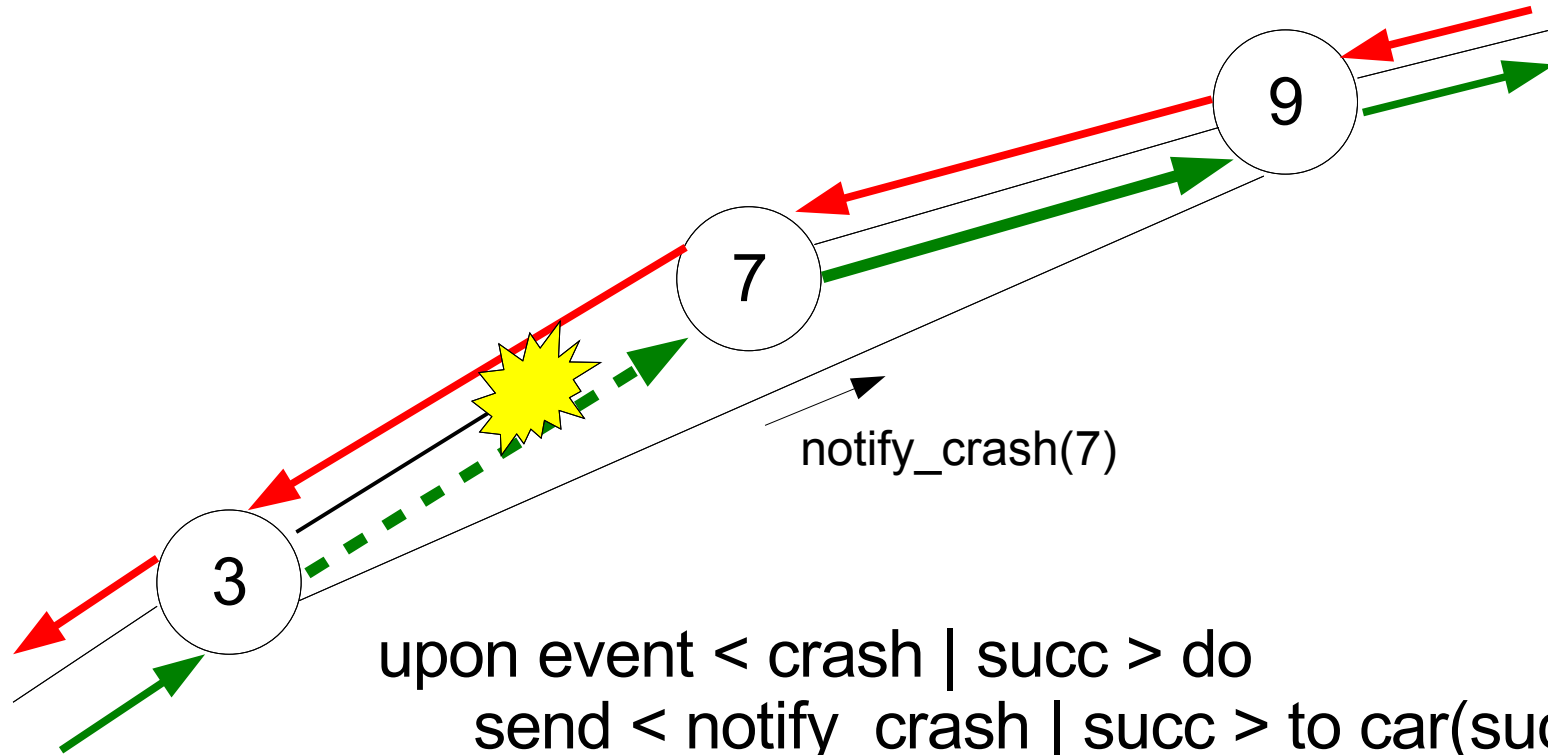


The crashing peer

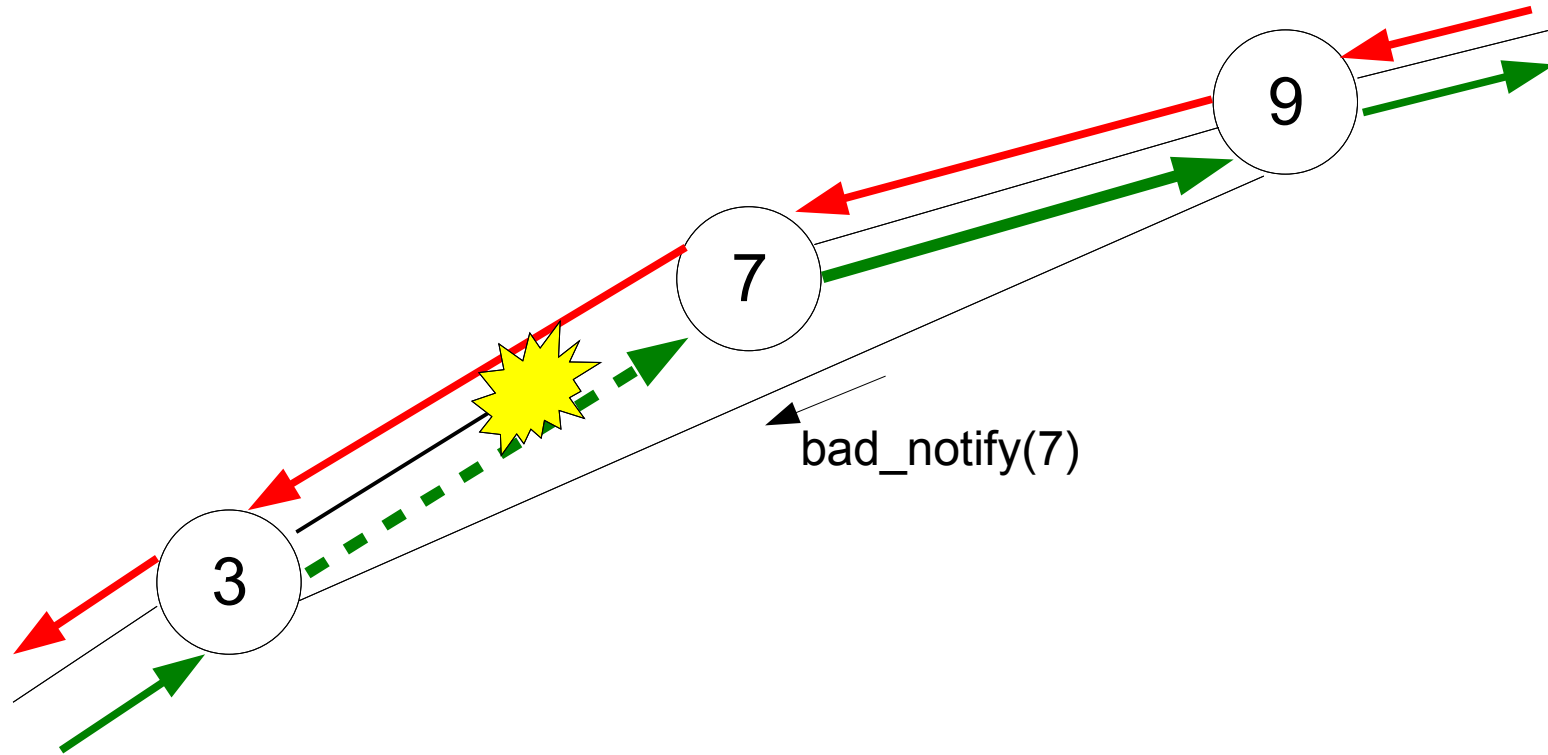
Perfect ring



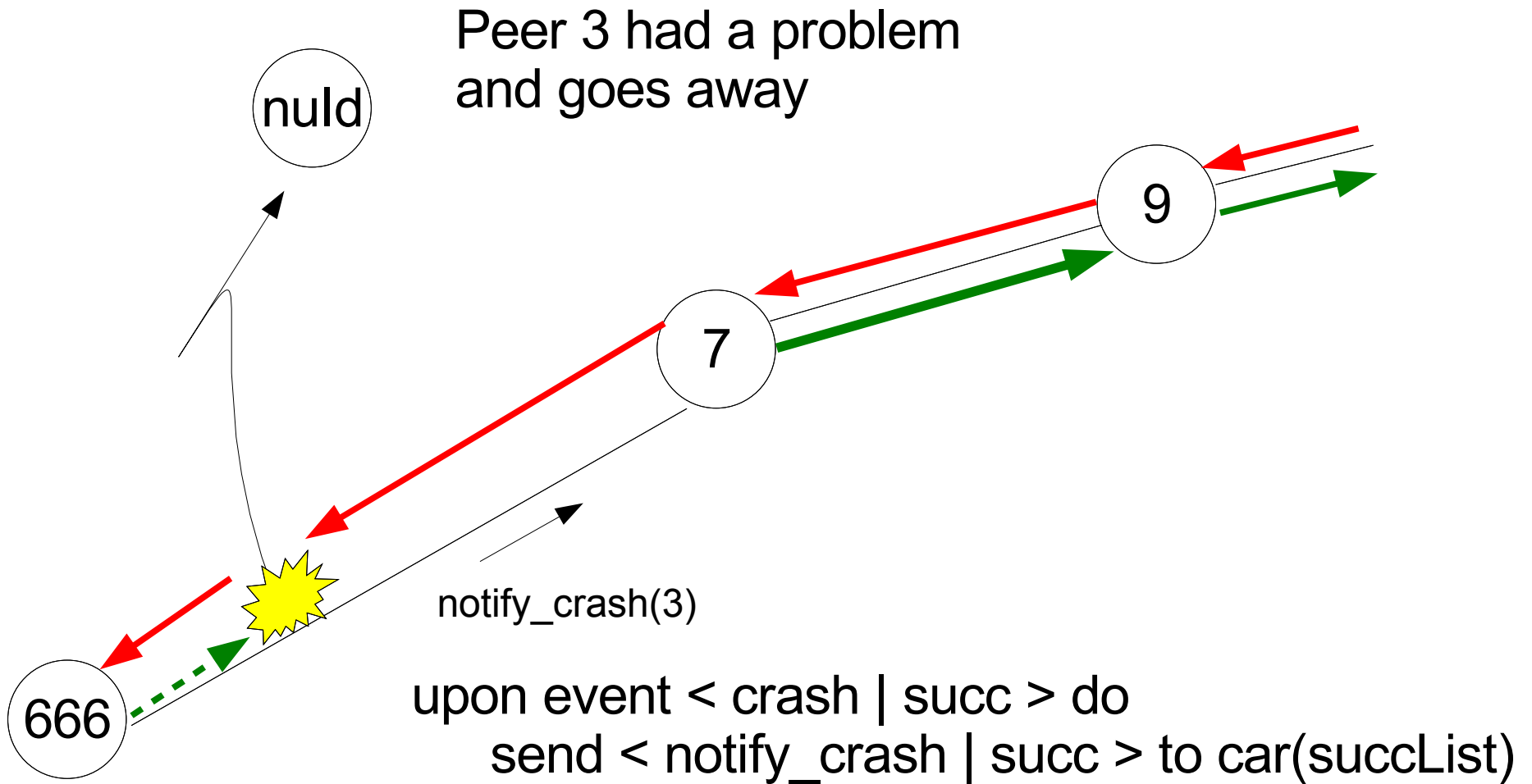
The crashing link



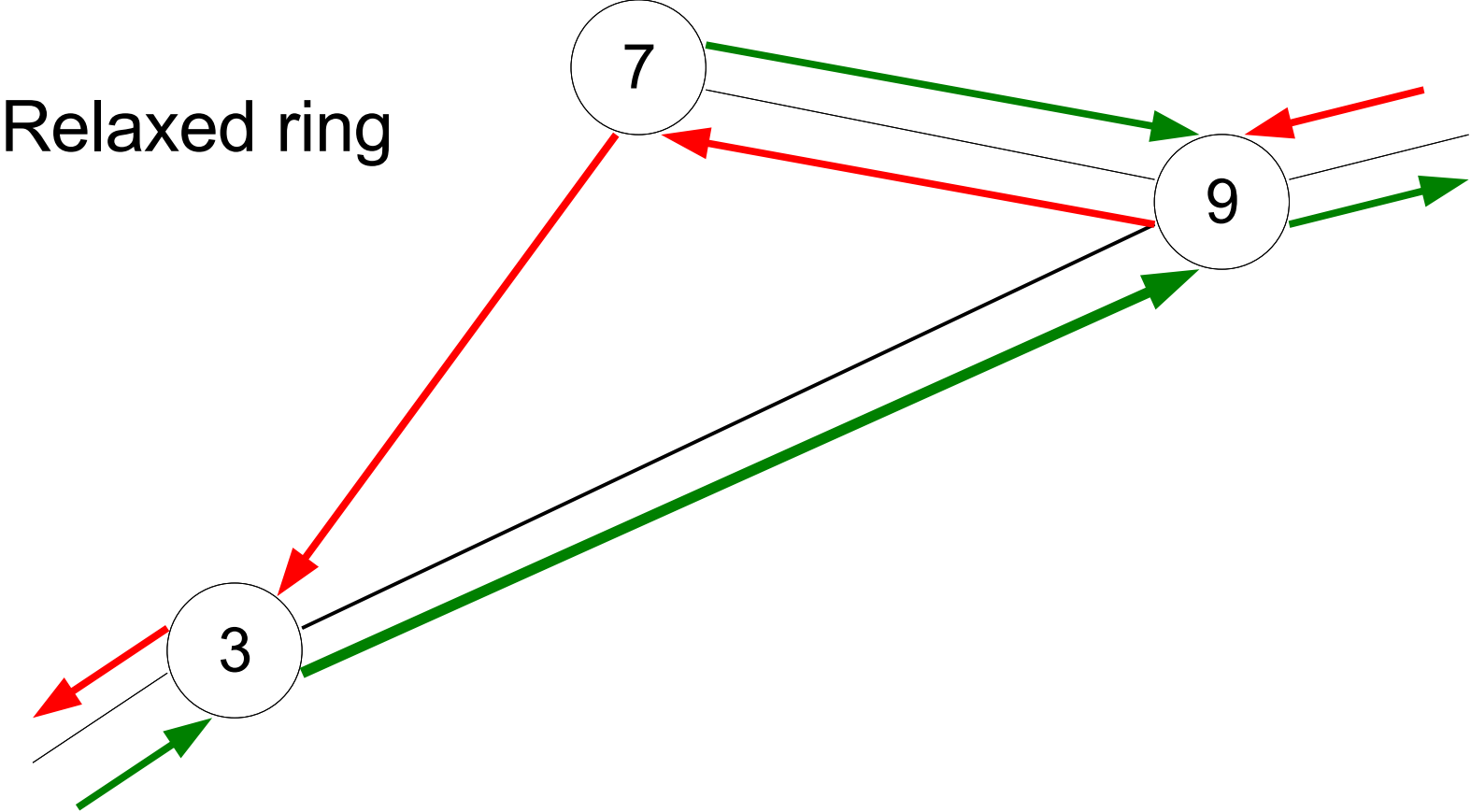
The crashing link



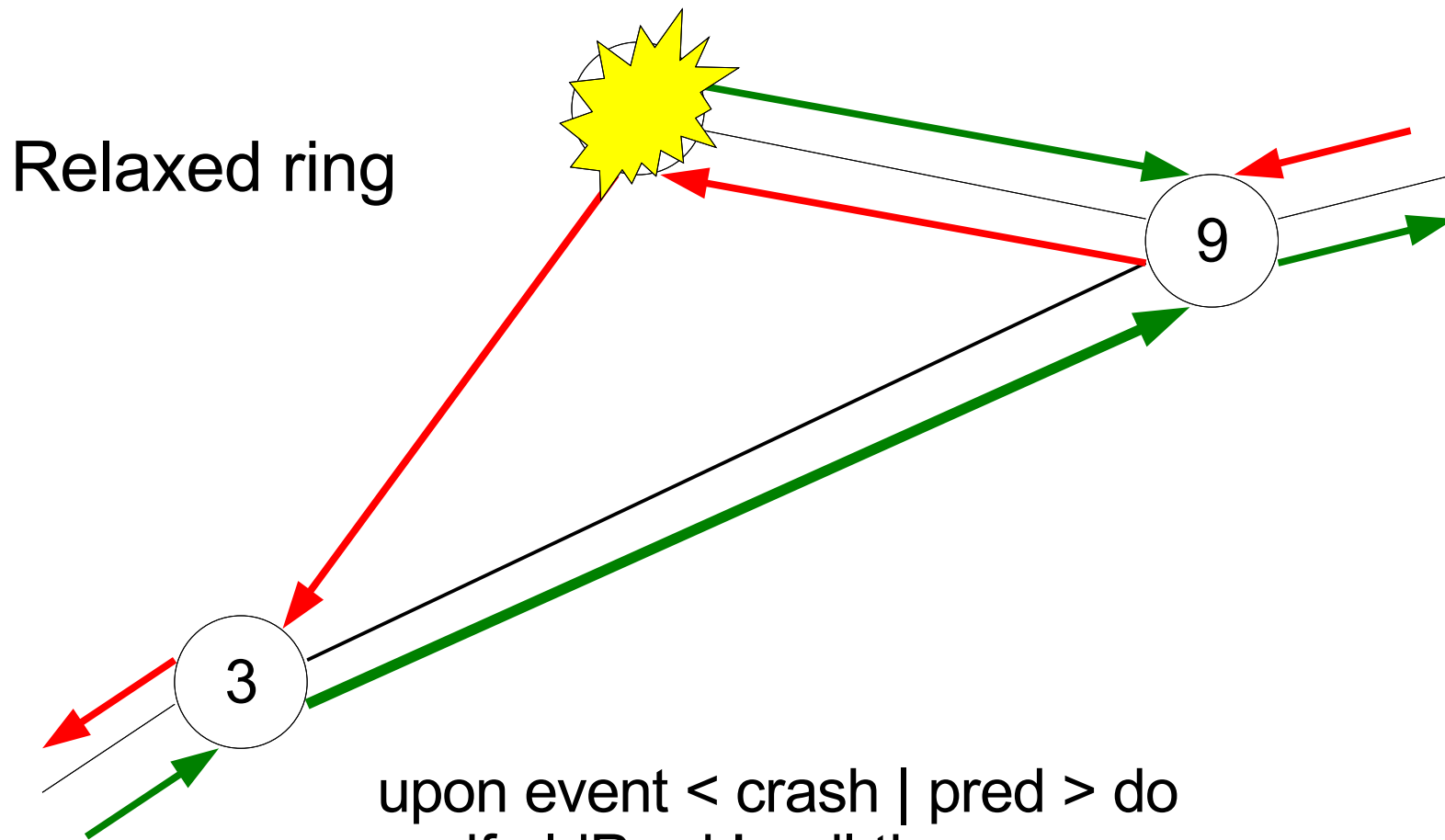
The crashing peer



The crashing peer

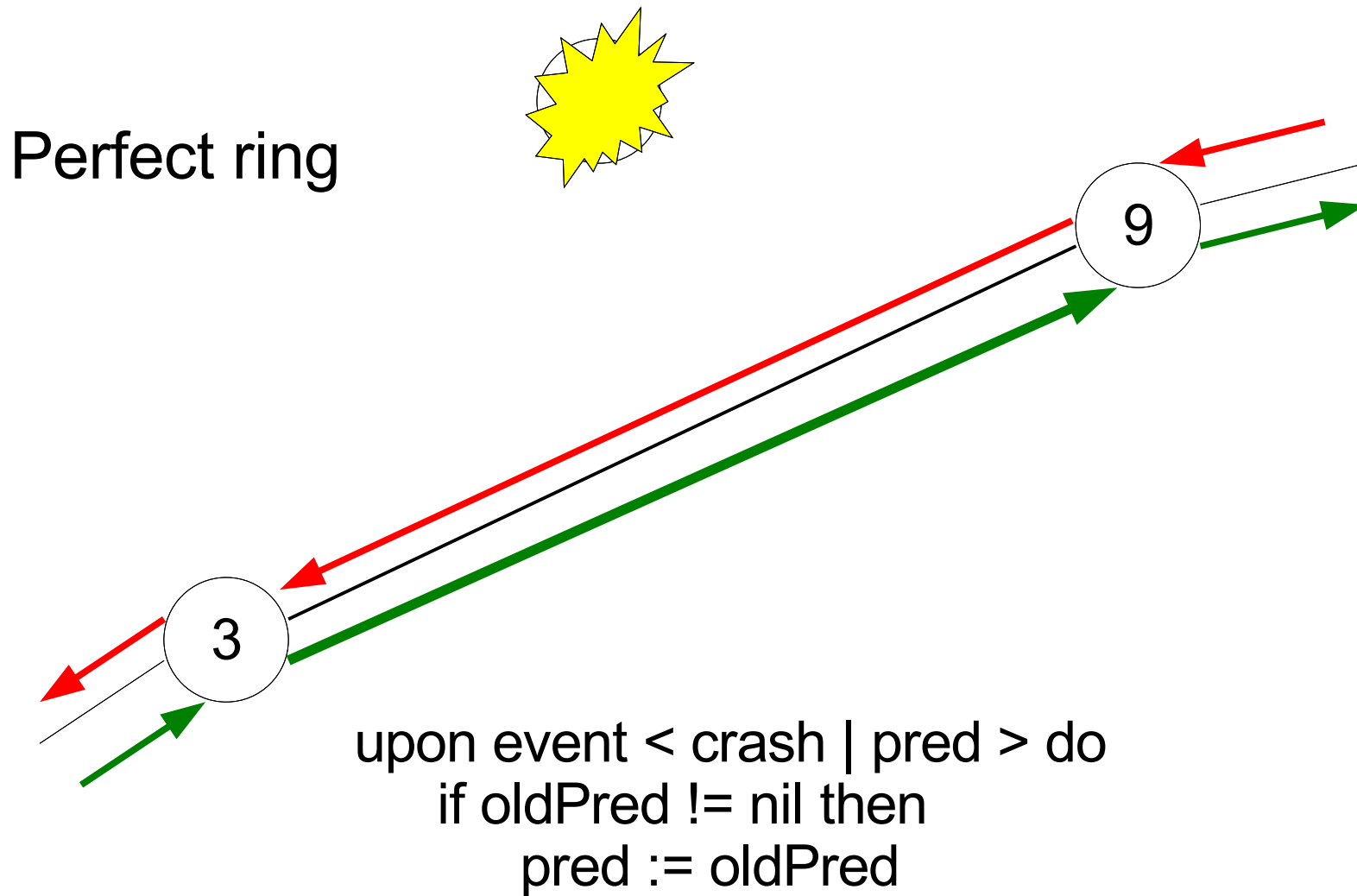


The crashing peer



```
upon event < crash | pred > do  
  if oldPred != nil then  
    pred := oldPred
```

The crashing peer



Message

- You are in the same ring as your successor
- First correctness, then efficient routing
- The predecessor links form a perfect ring

Should I defy Murphy's law?