Ideal Stabilization

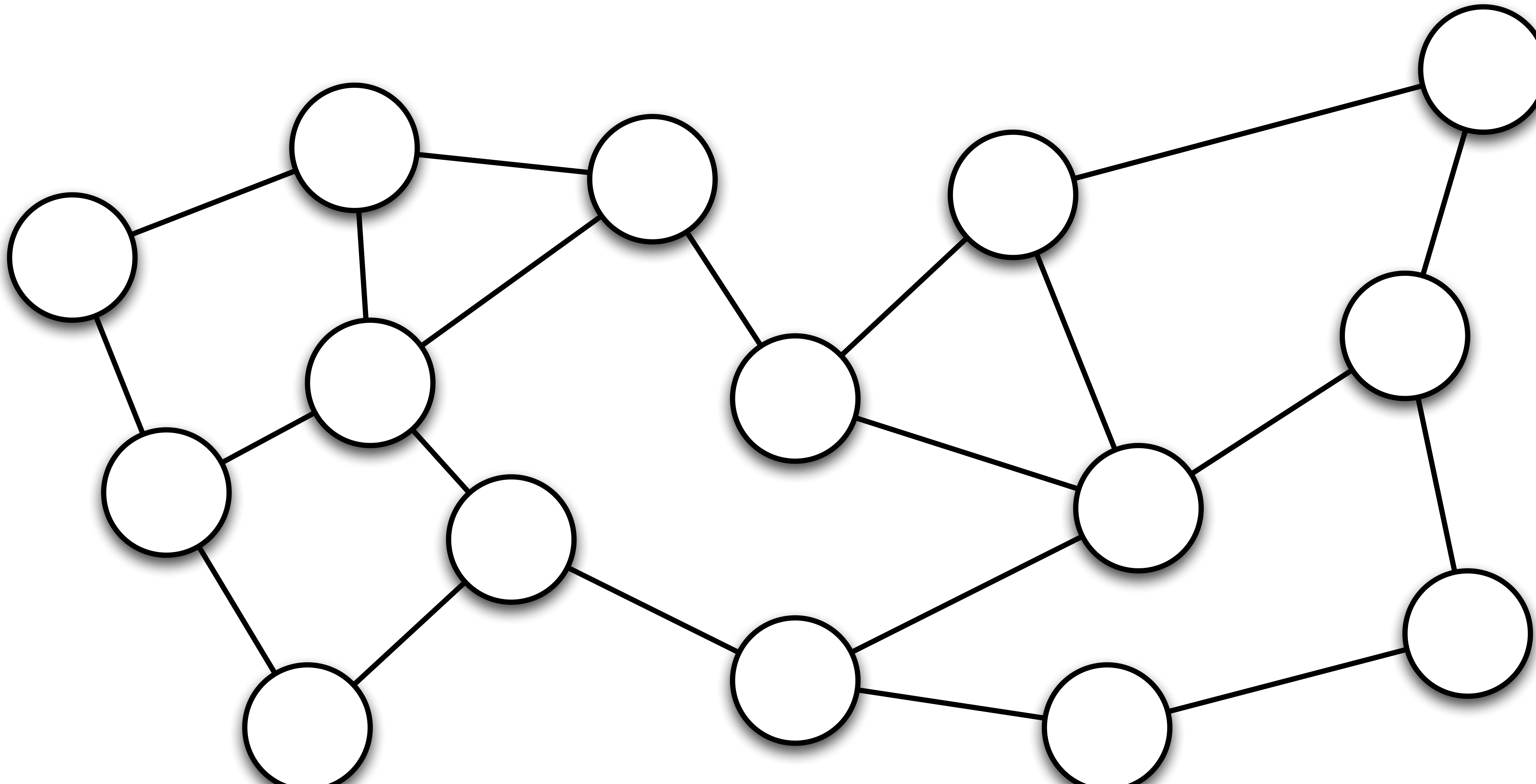
Mikhail Nesterenko
Kent State University

Sébastien Tixeuil
UPMC Sorbonne Universités & IUF

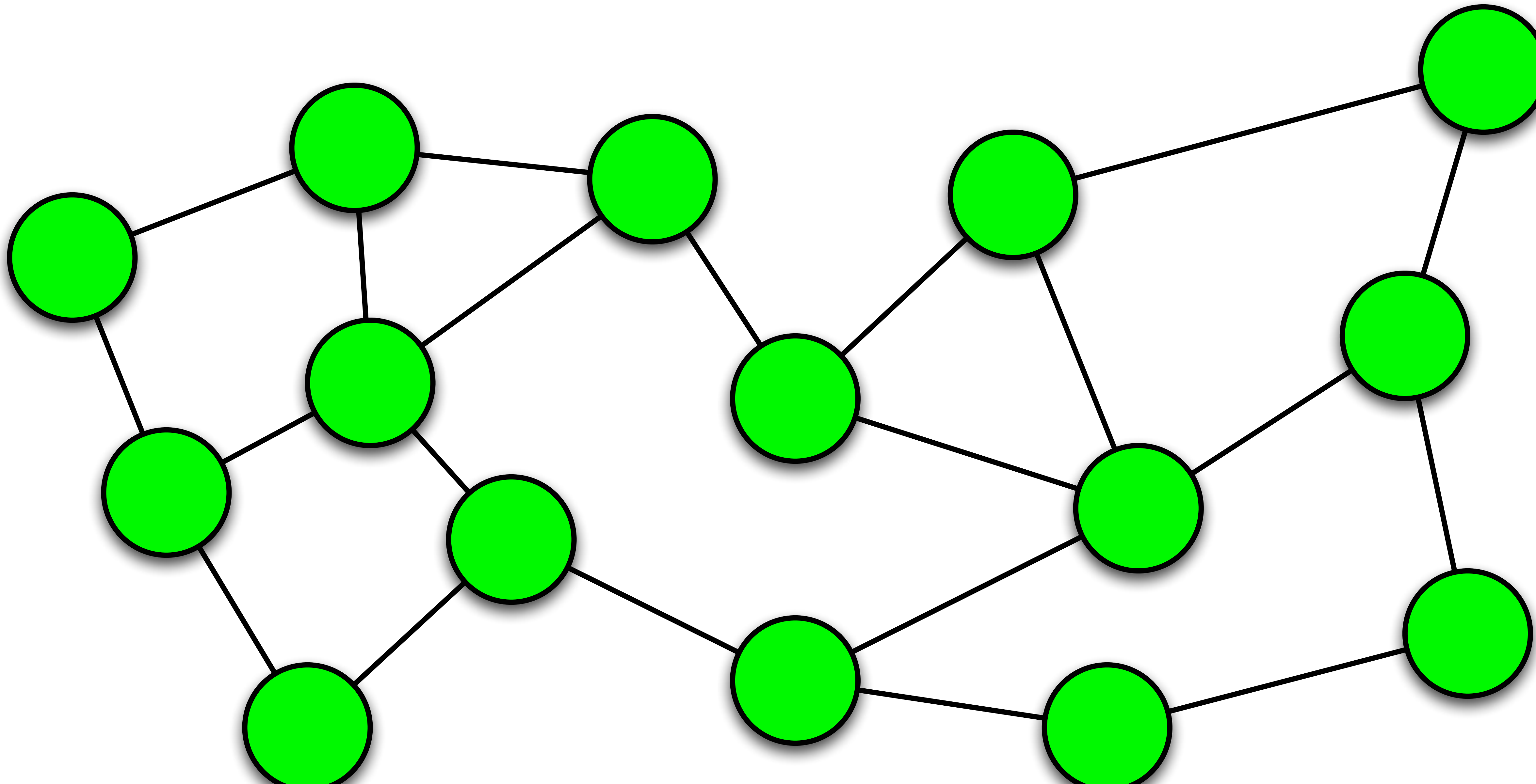
AINA 2011, Singapore, 24 March 2011

Motivation

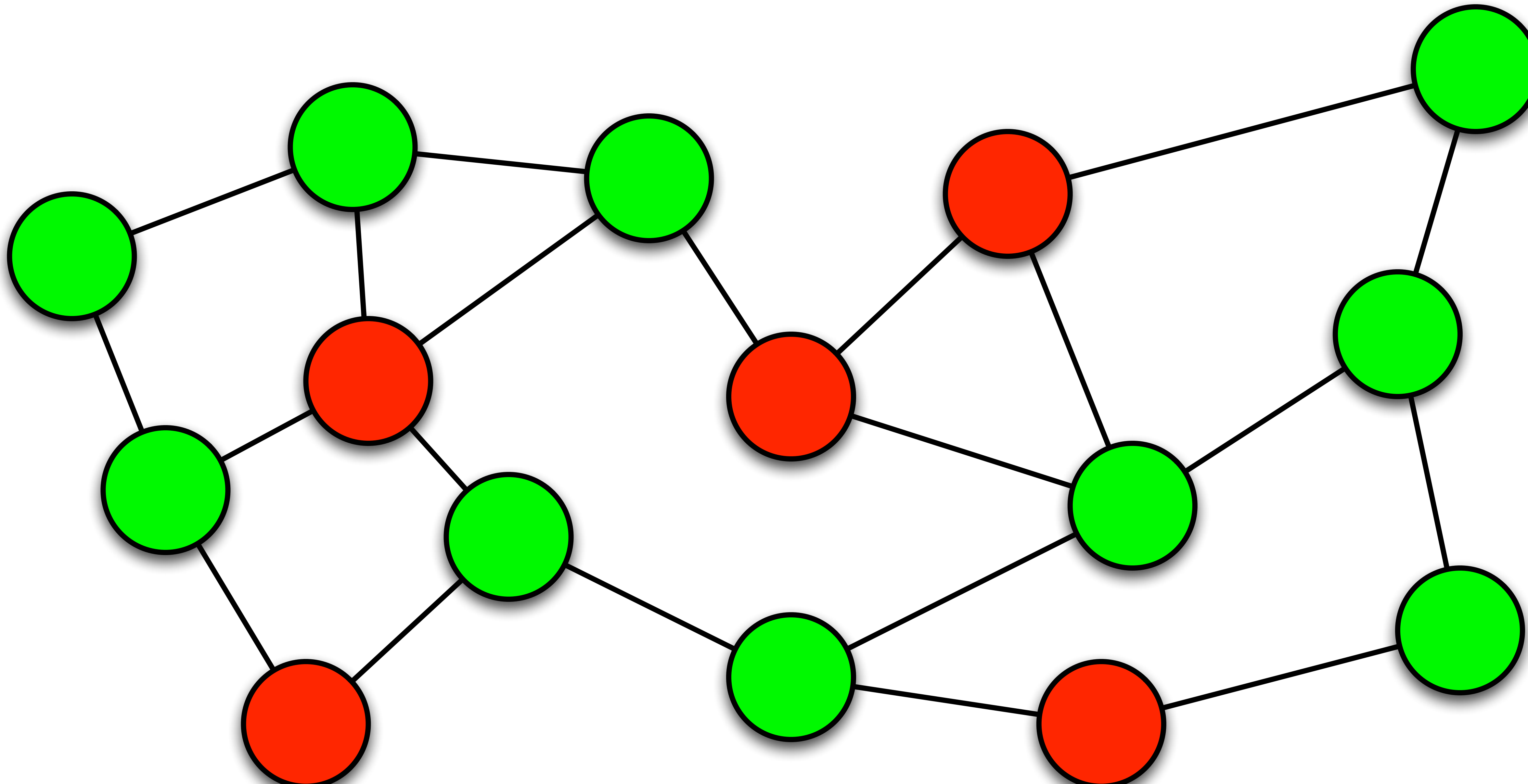
Distributed System



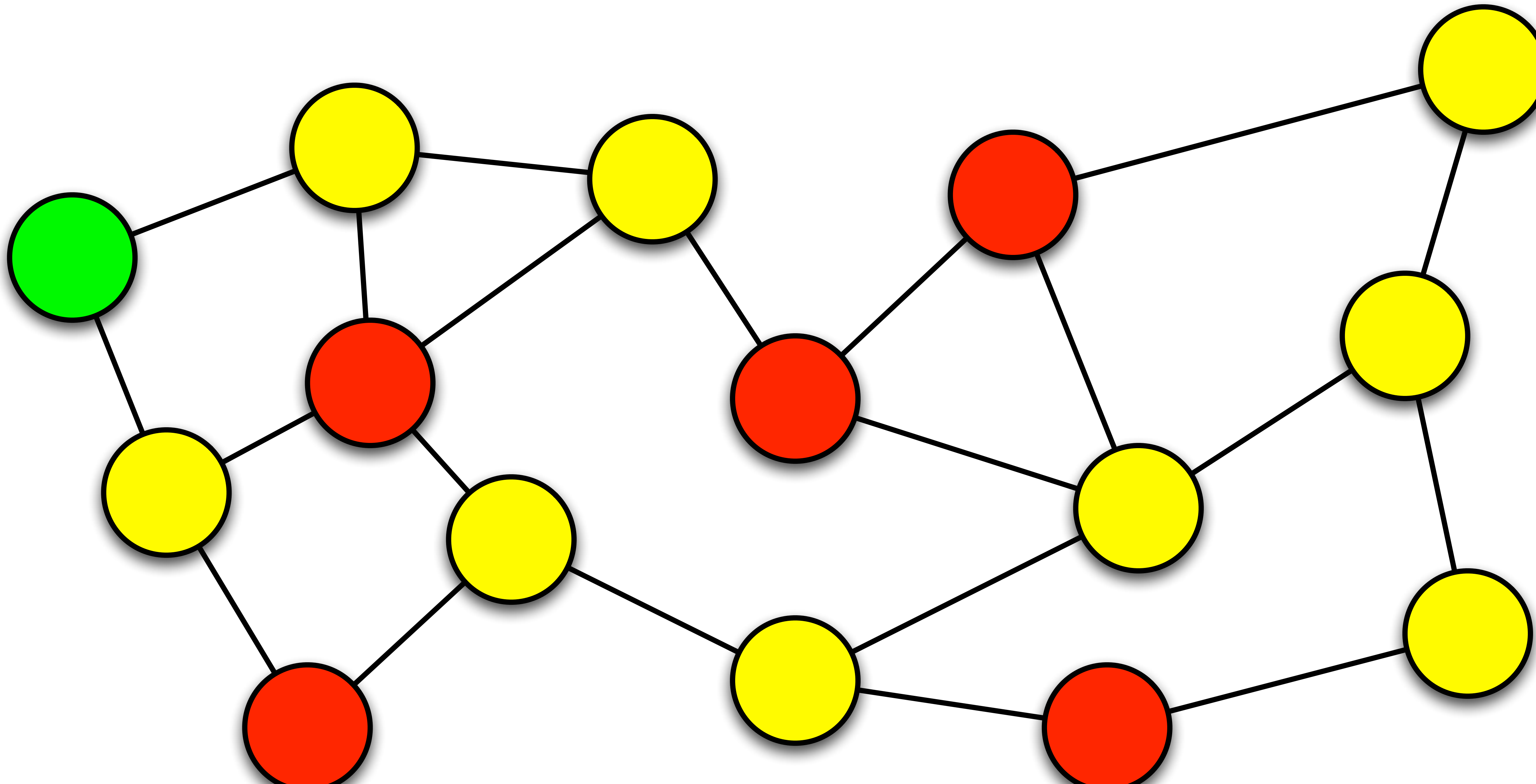
Legitimate State



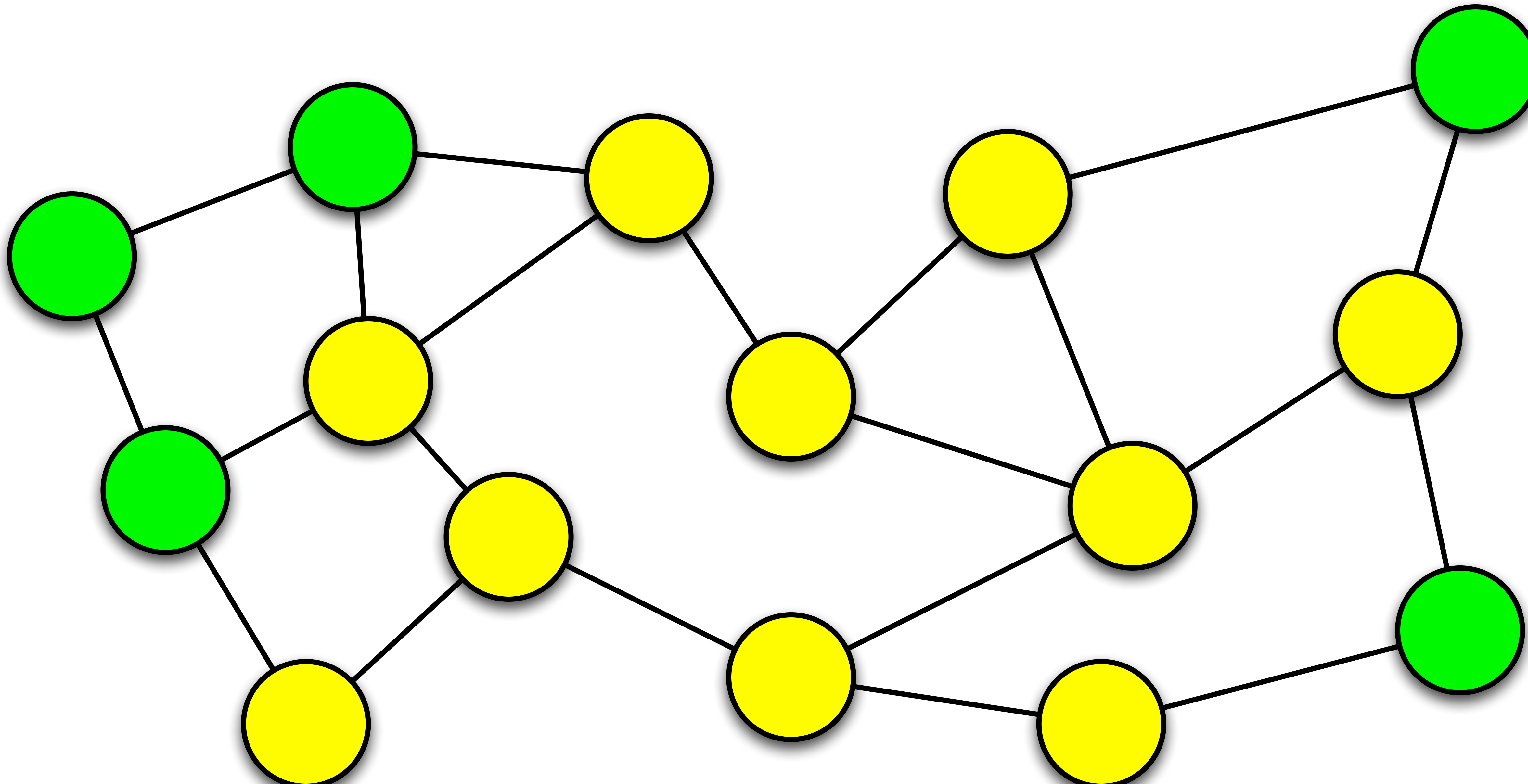
Transient Faults



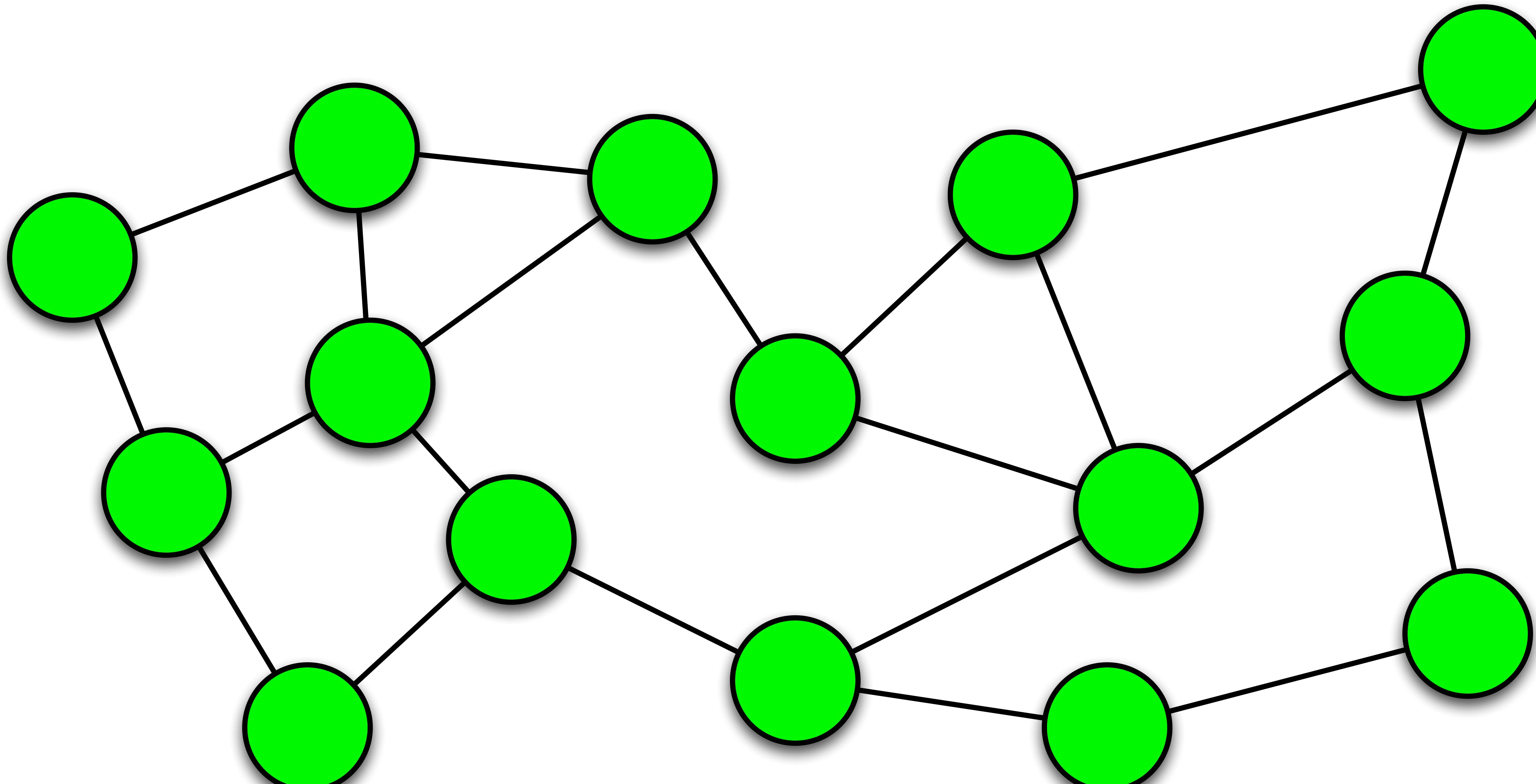
Recovery



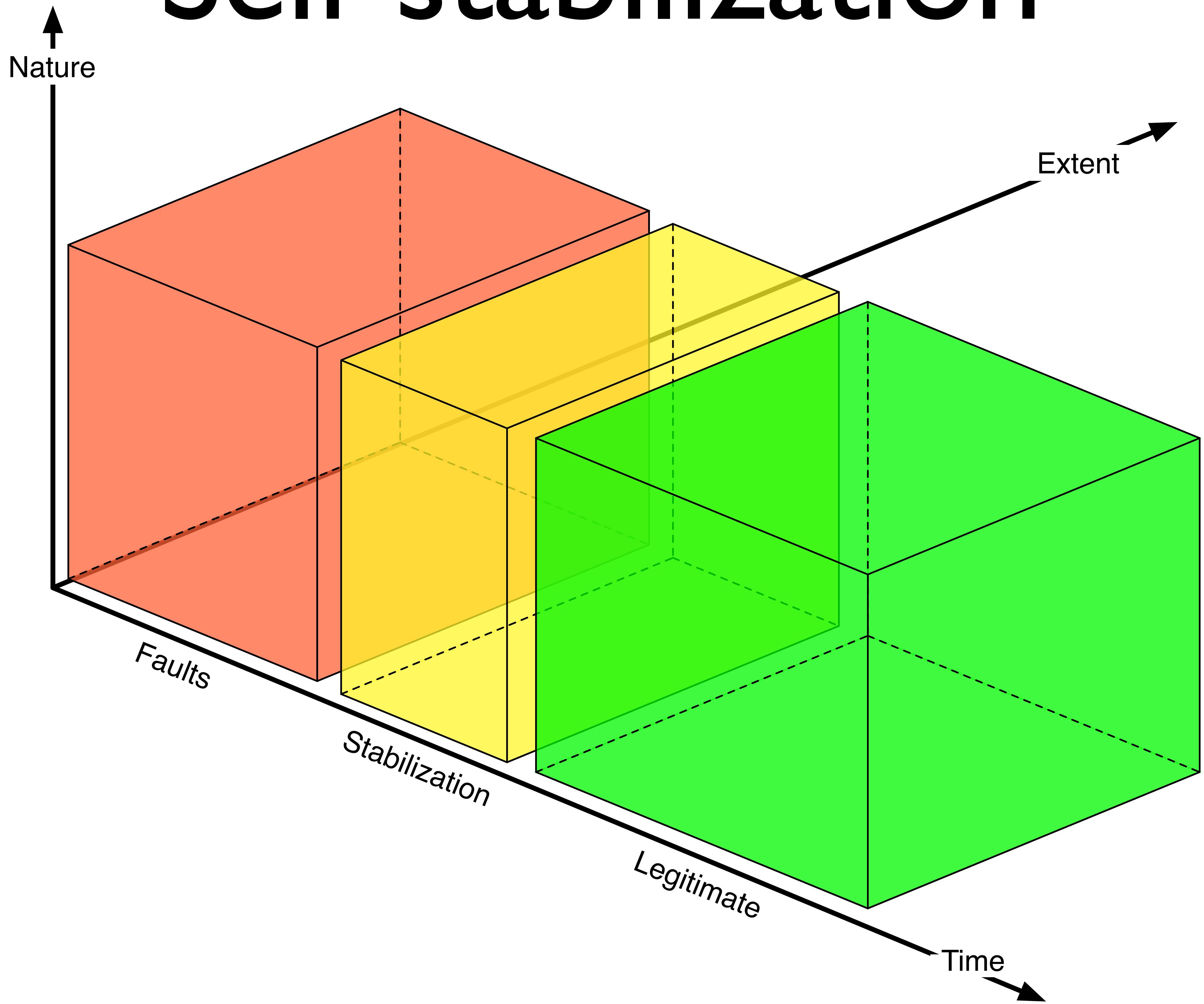
Recovery



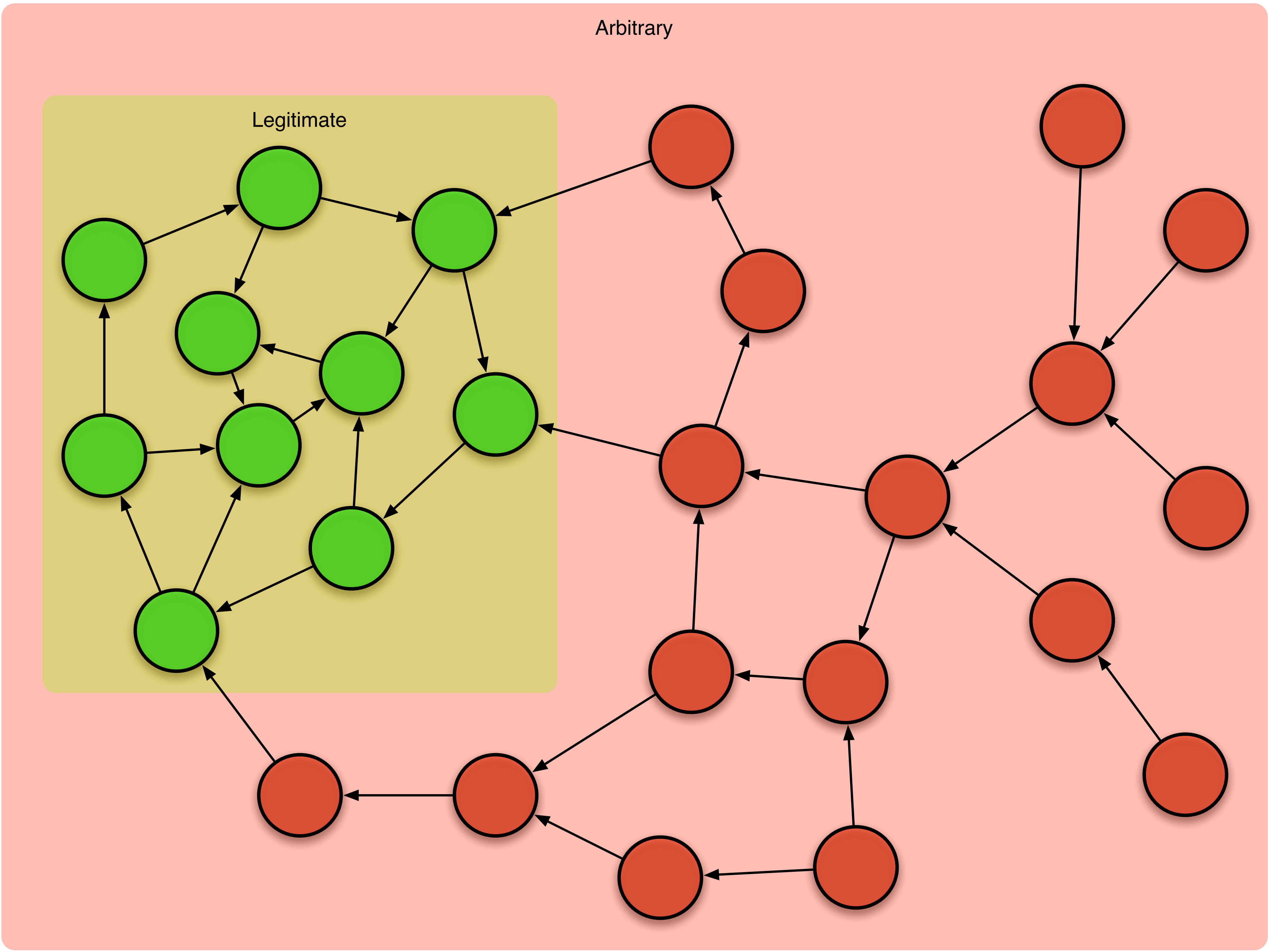
Legitimate State



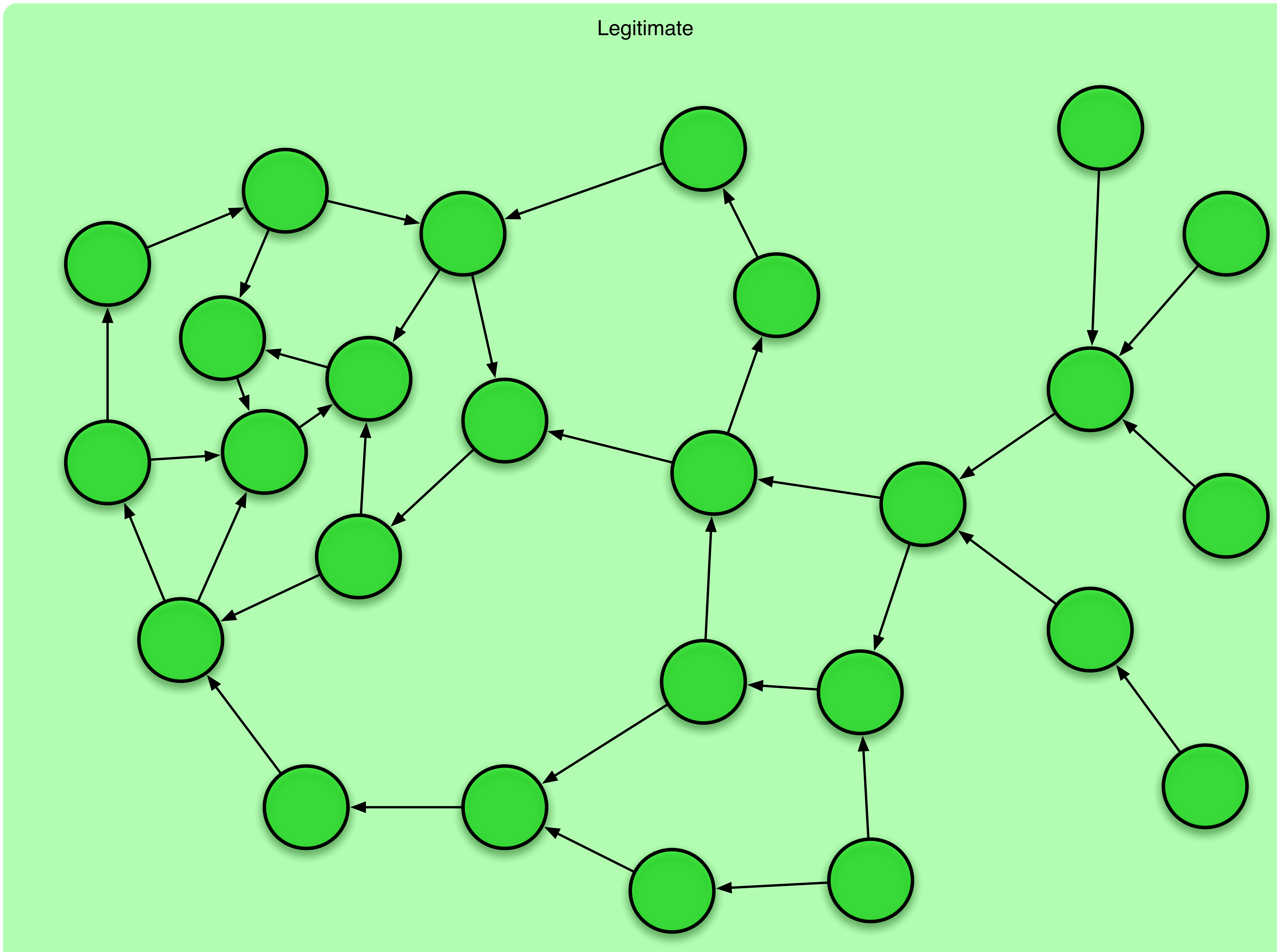
Self-stabilization



Self-stabilization



Ideal Stabilization

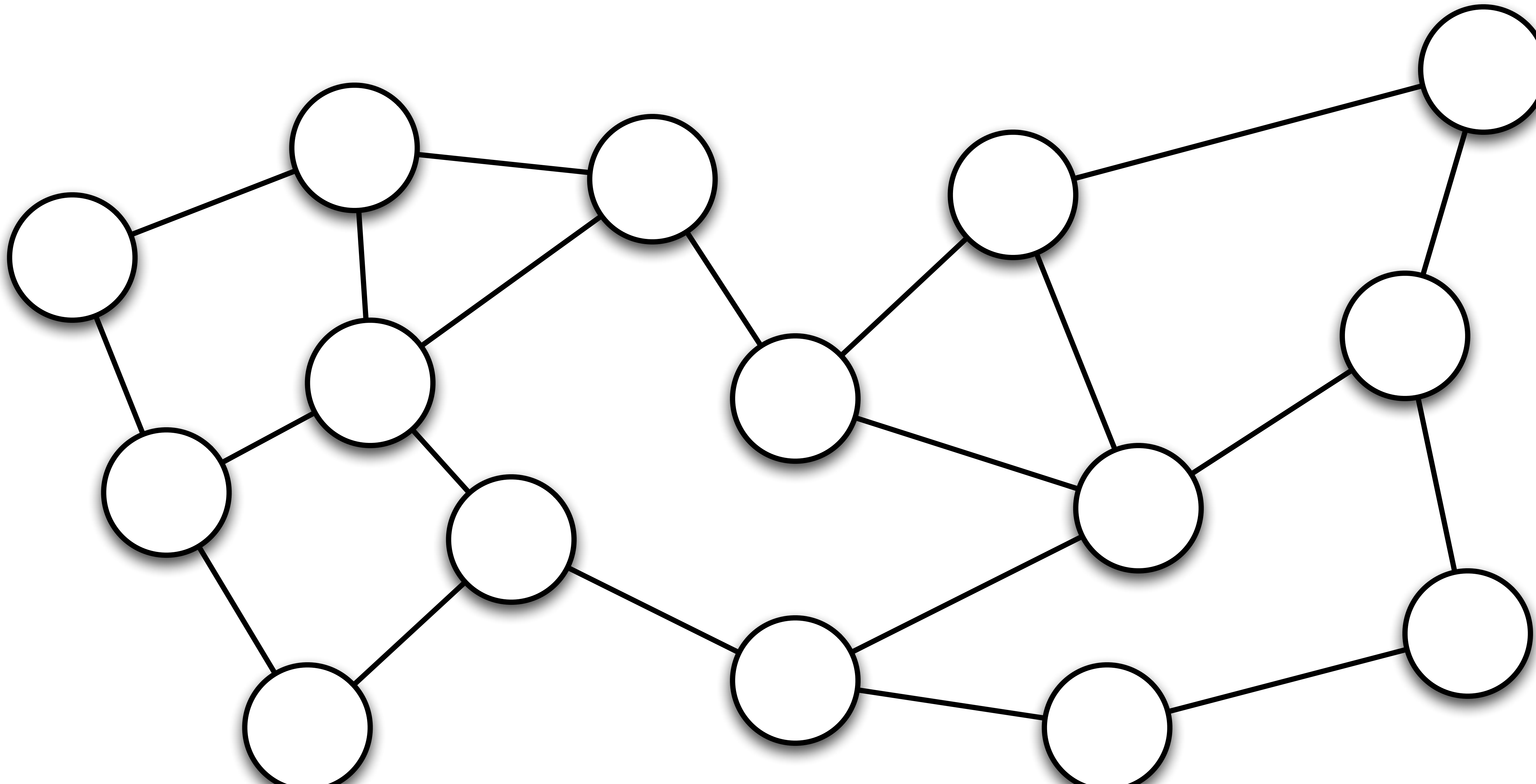


Questions

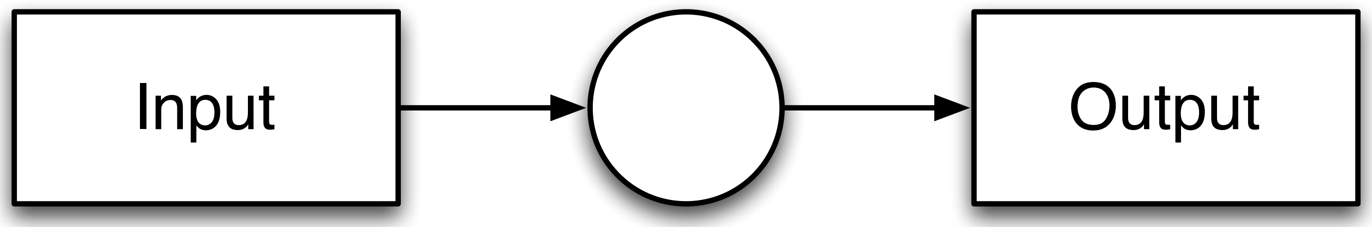
- Existence ?
- Construction ?
- Composition ?
- Implementation *vs.* Specification ?
- Proof techniques ?

Model

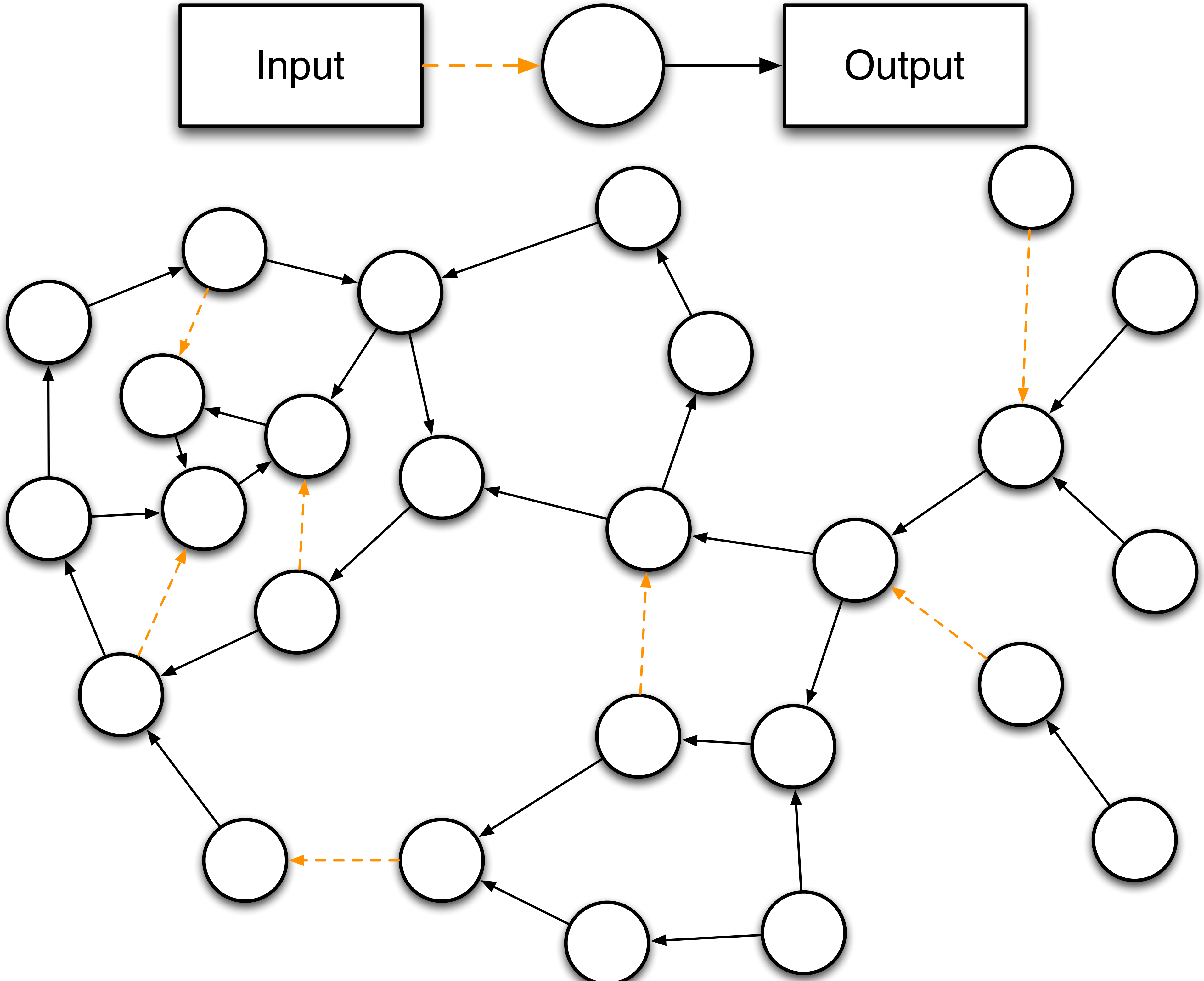
Distributed System



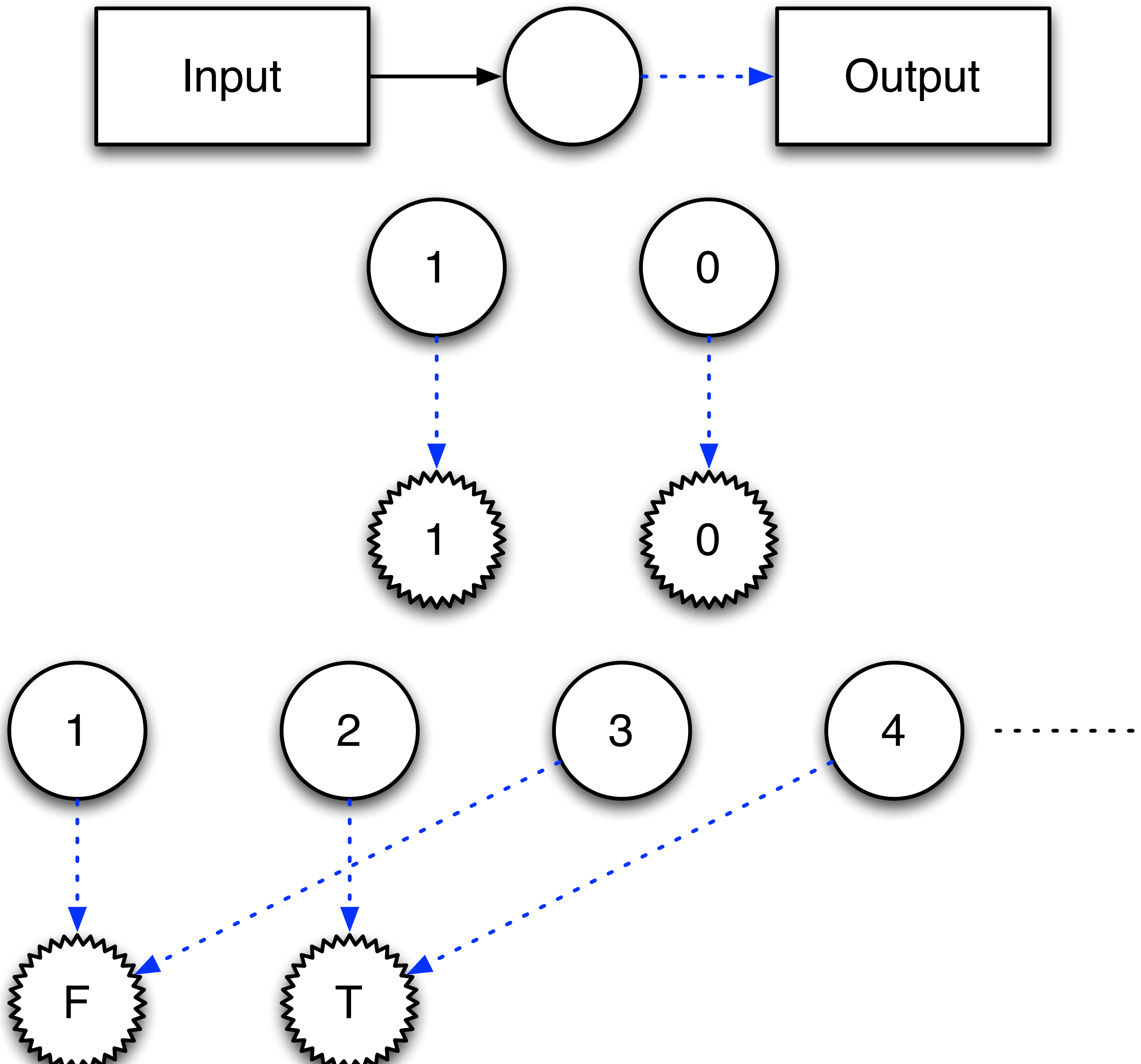
Model



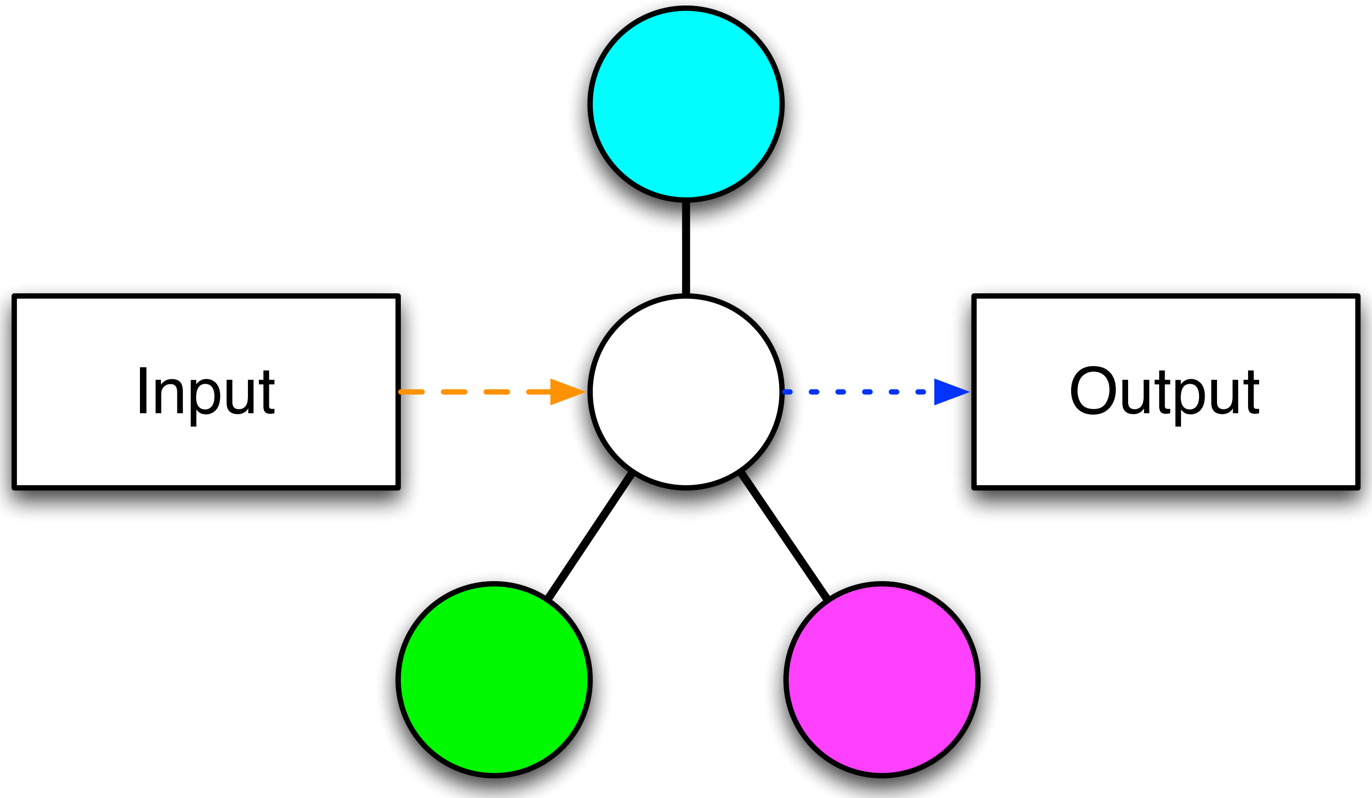
Model



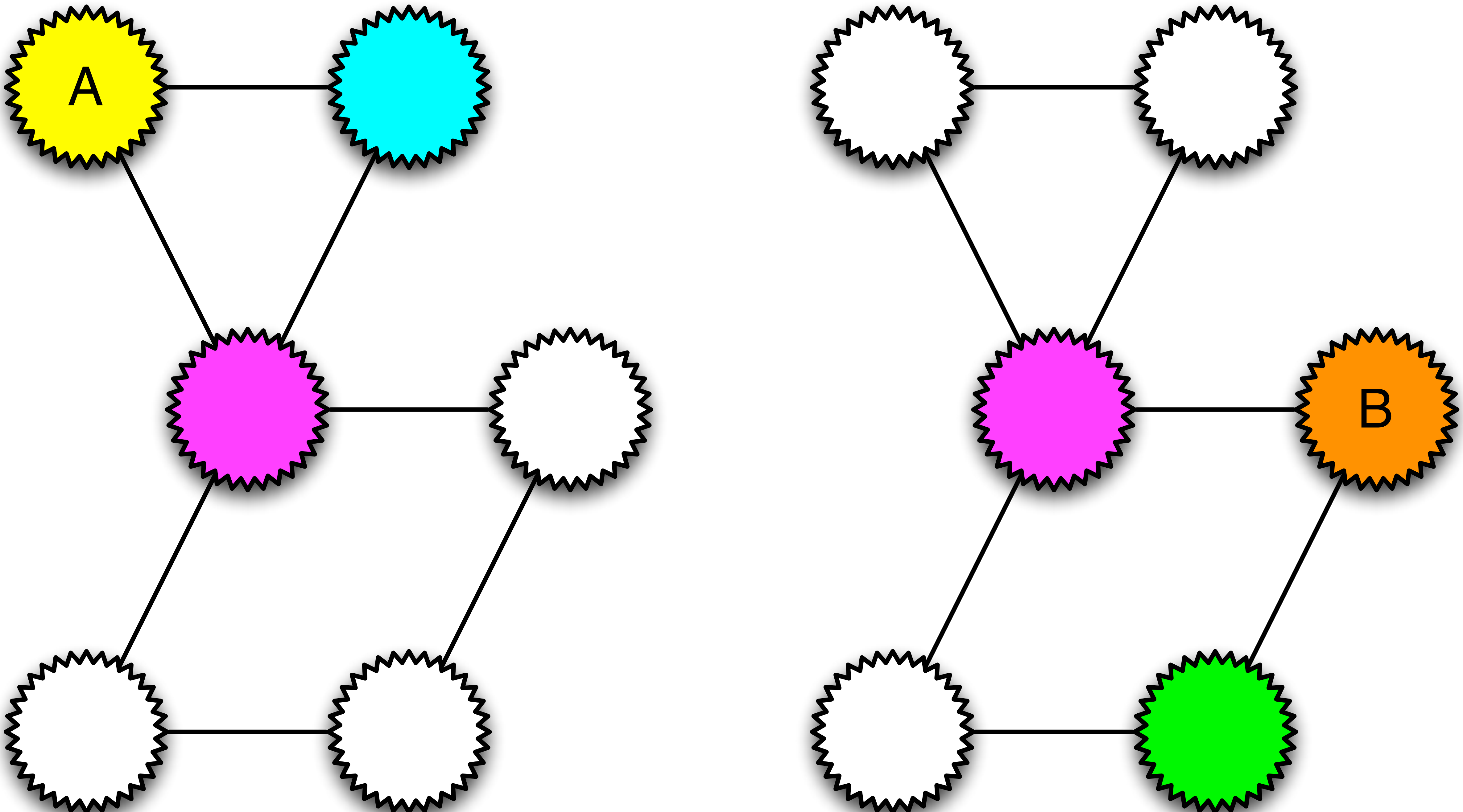
Model



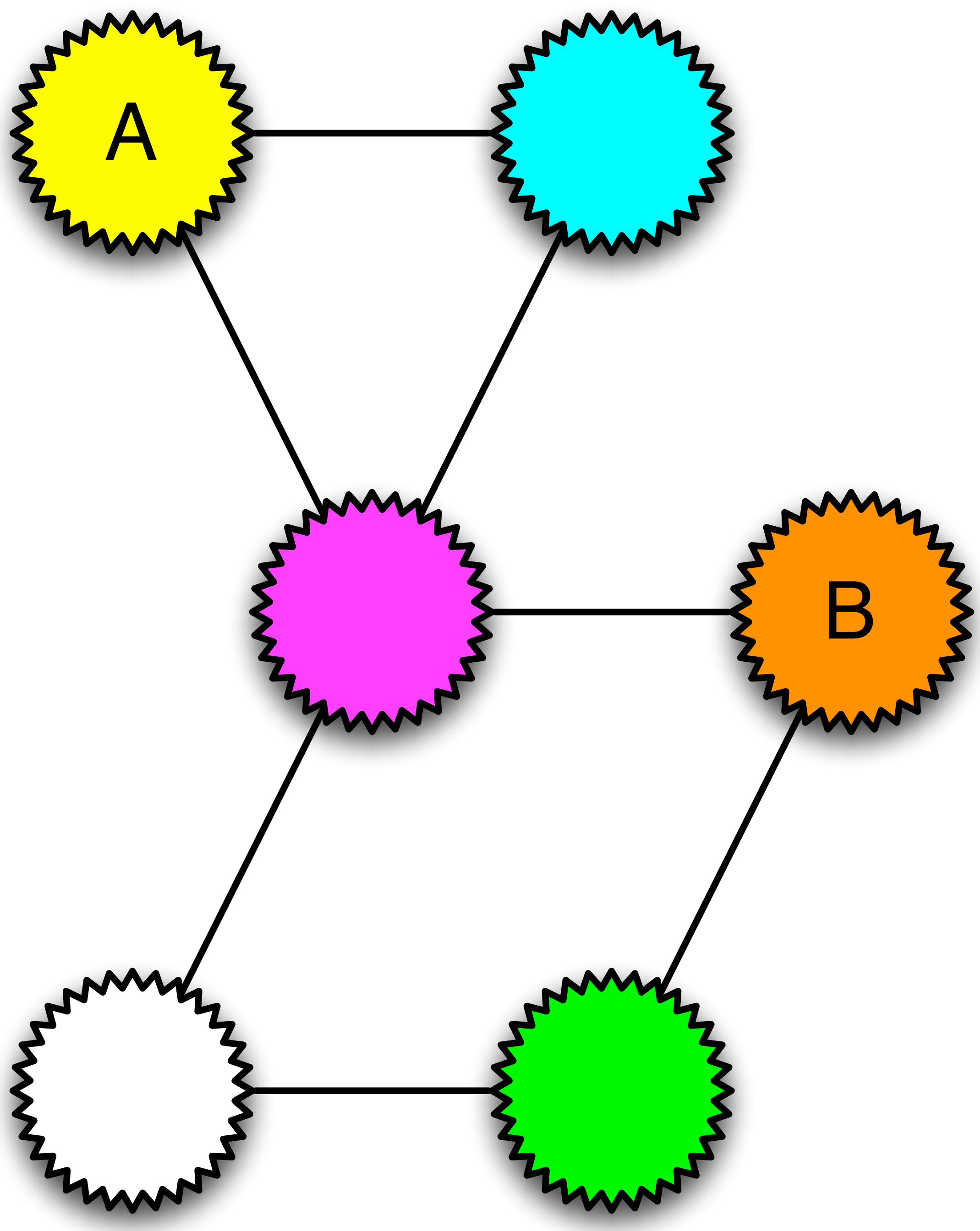
Model



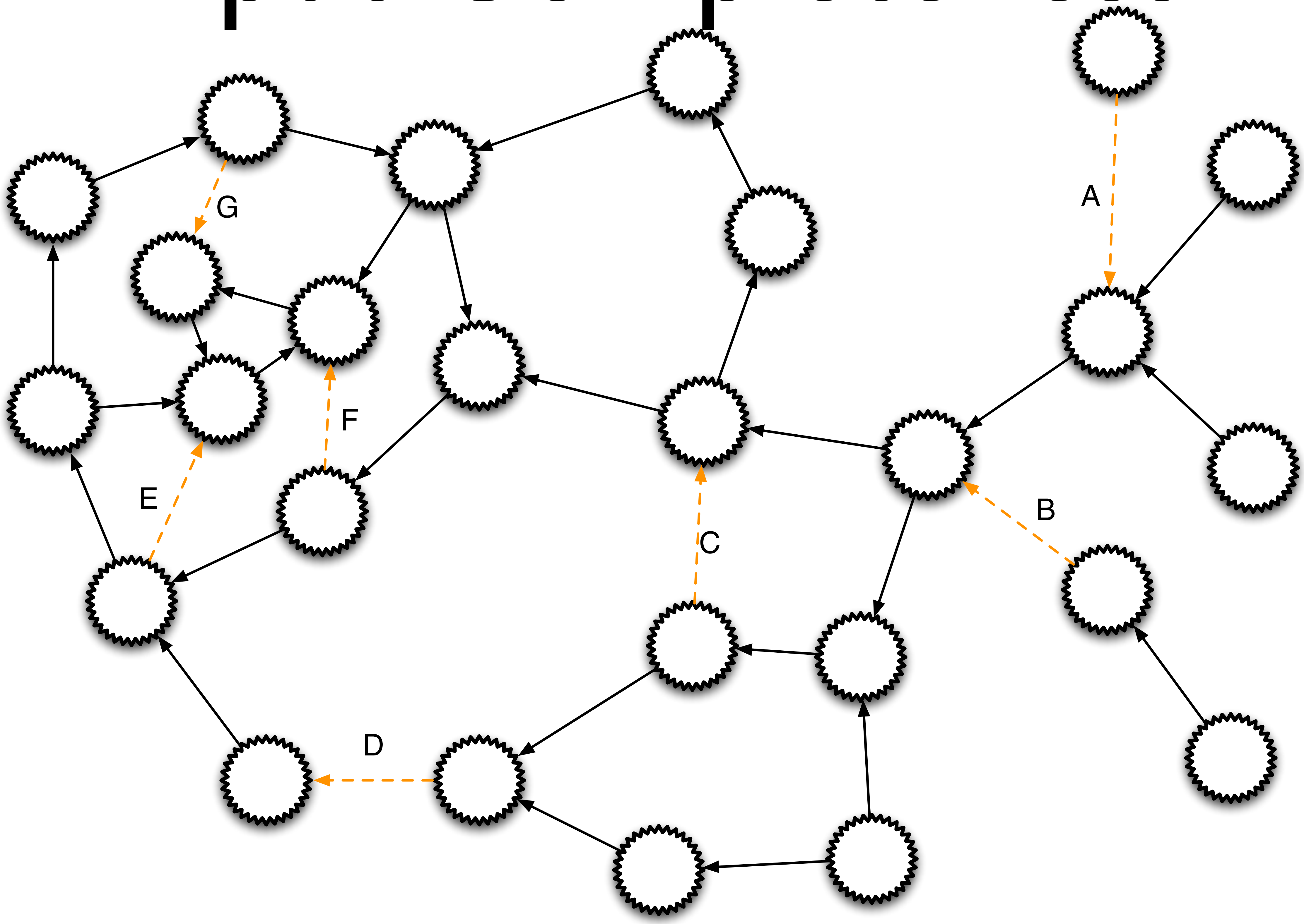
Merge Symmetry



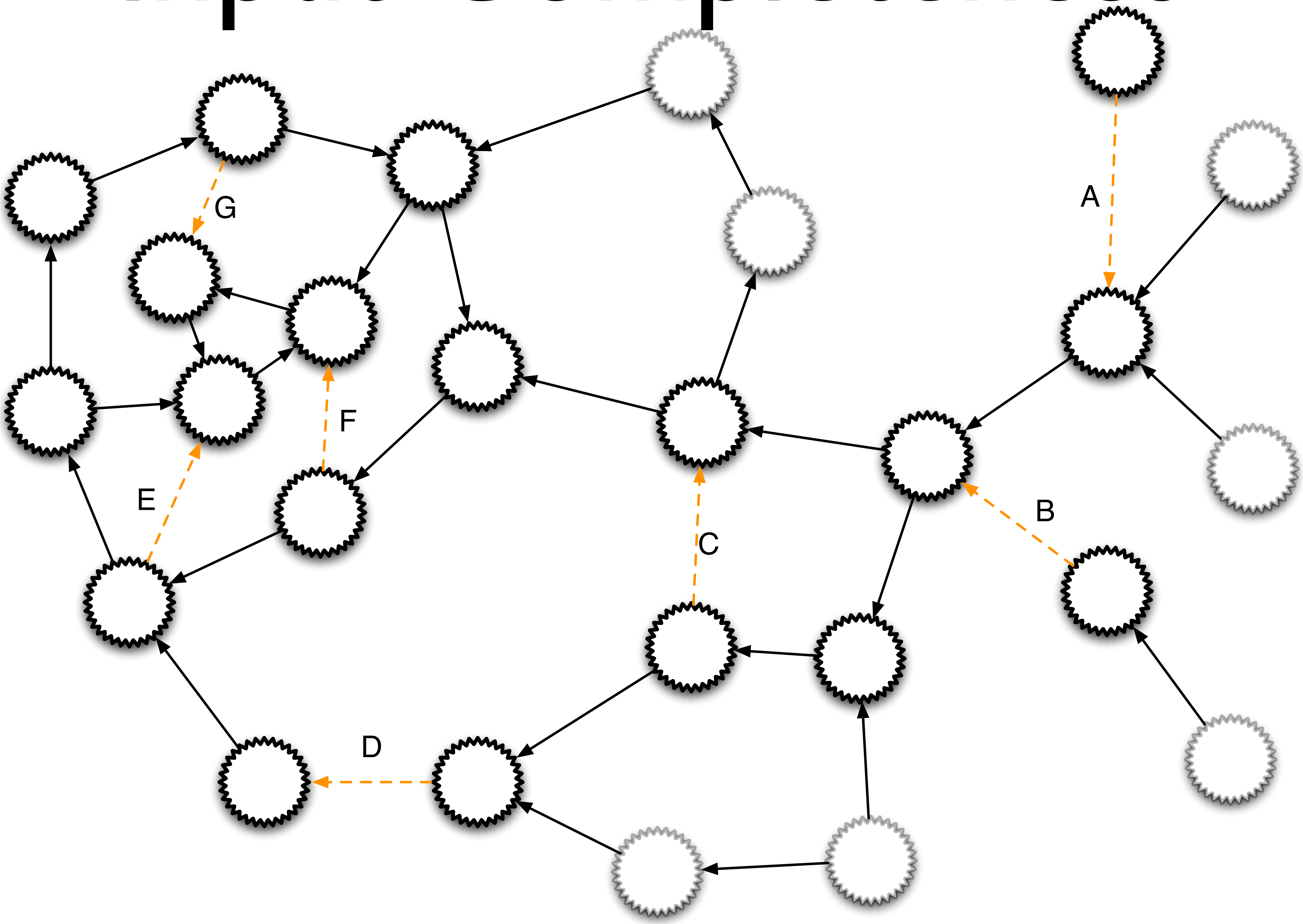
Merge Symmetry



Input Completeness

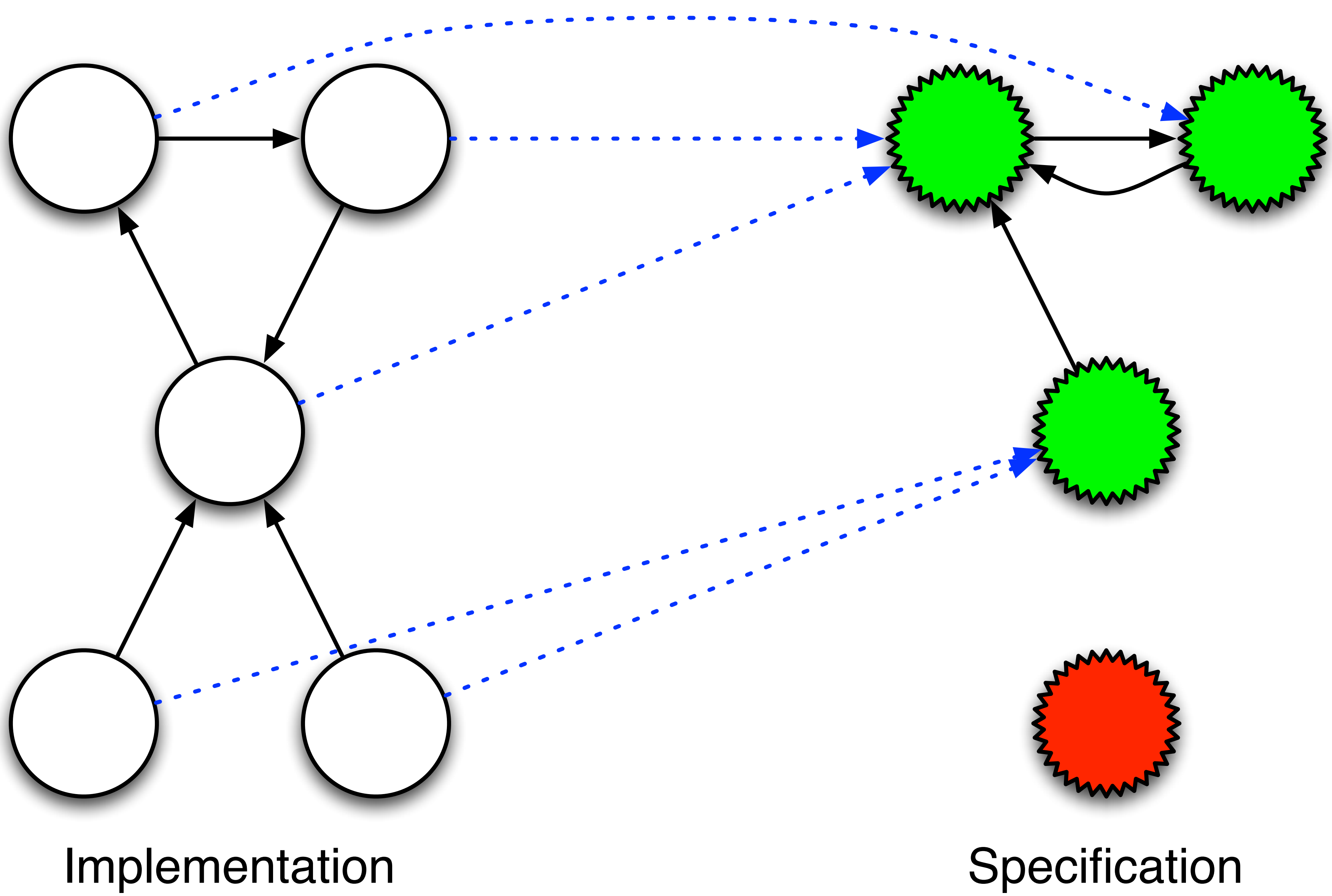


Input Completeness



Ideal Stabilization to *Non-ideal* Specifications

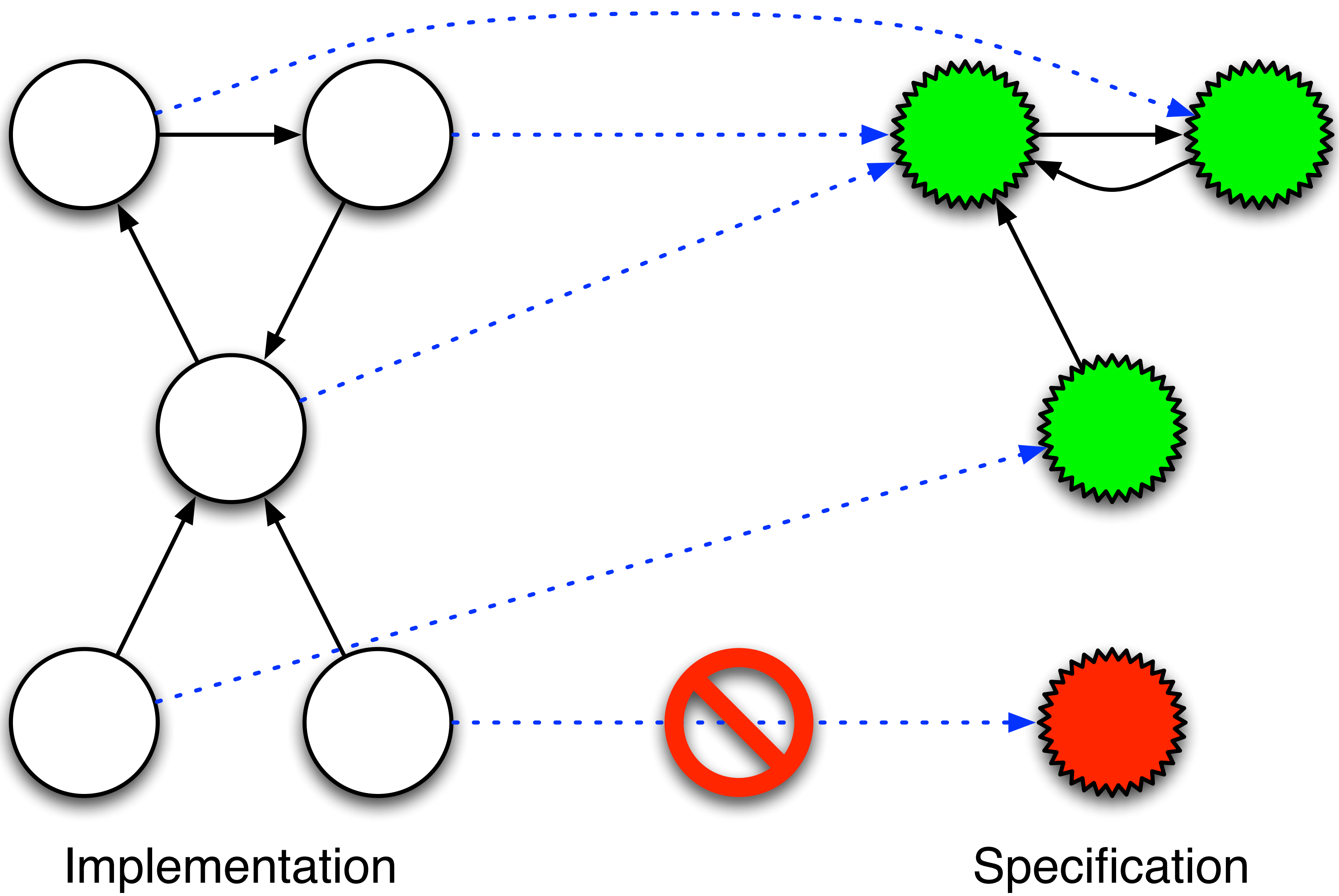
State Displacement



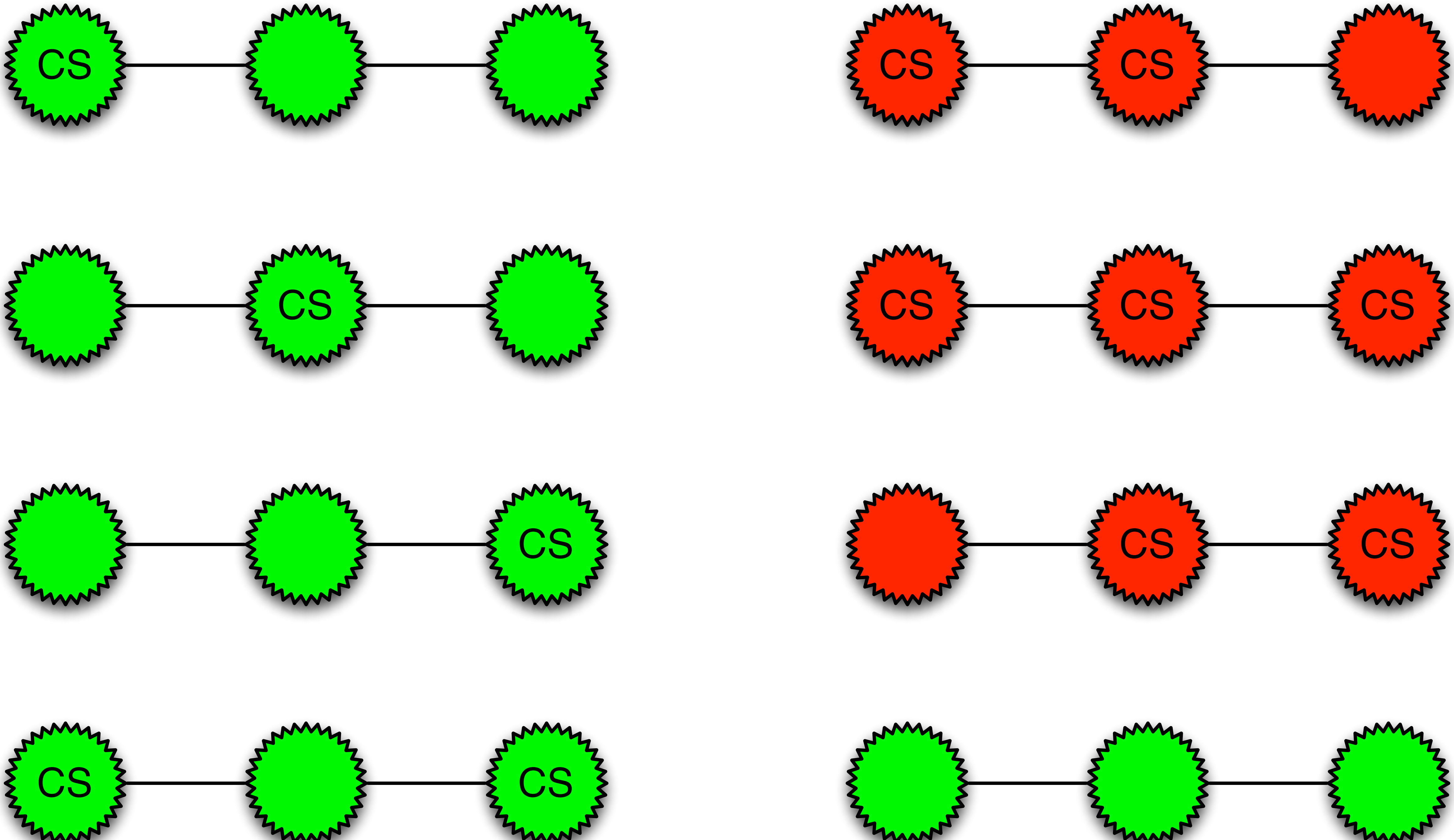
A Necessary Condition

- **Ideal stabilization** may be *possible only if* the specification contains an *input-complete* subset of sequences such that every disallowed specification state contains at least one process whose projection does not occur in the subset.

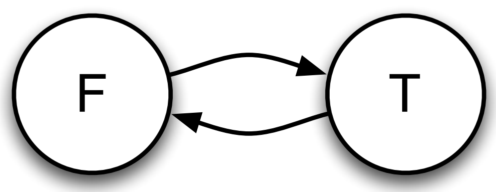
A Necessary Condition



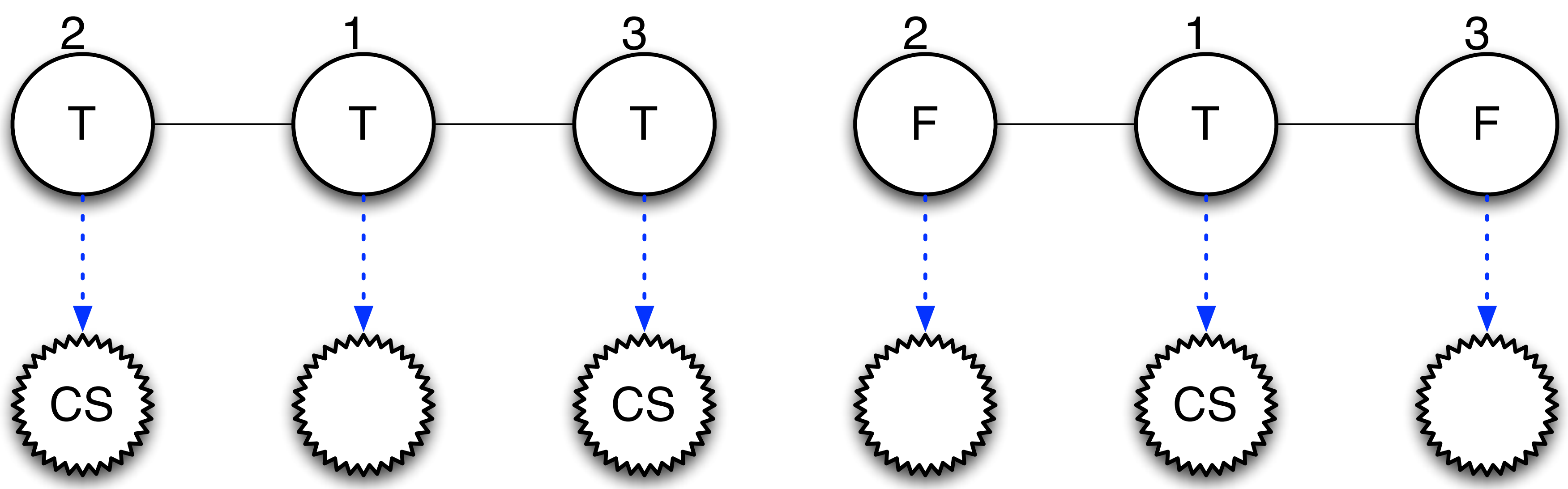
Conflict Managers

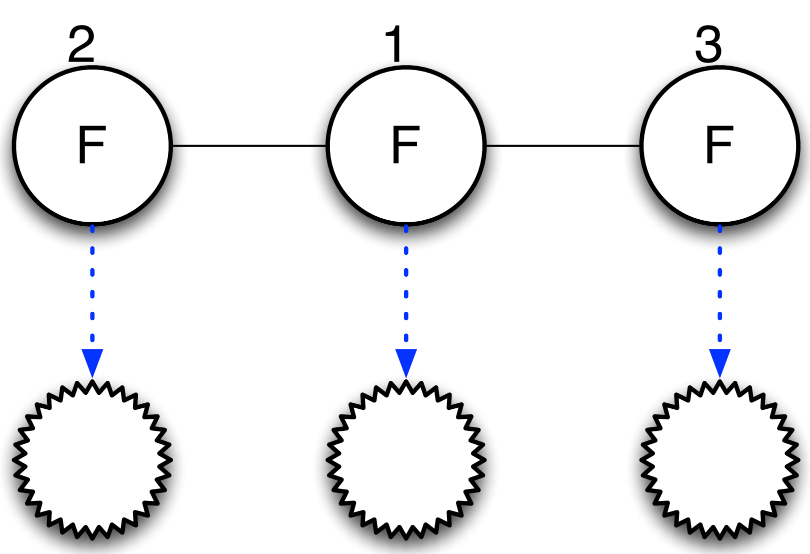


Conflict Managers



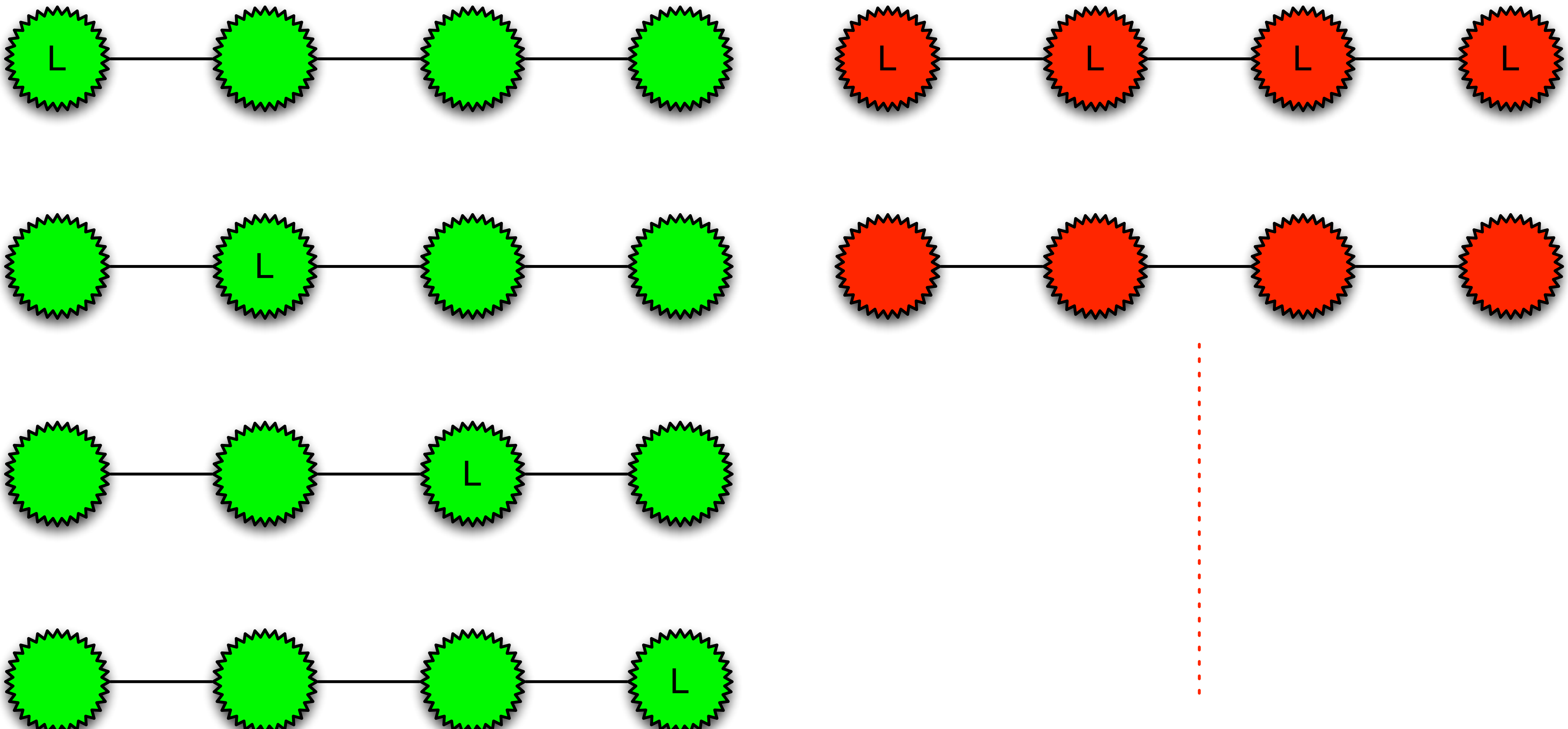
Program



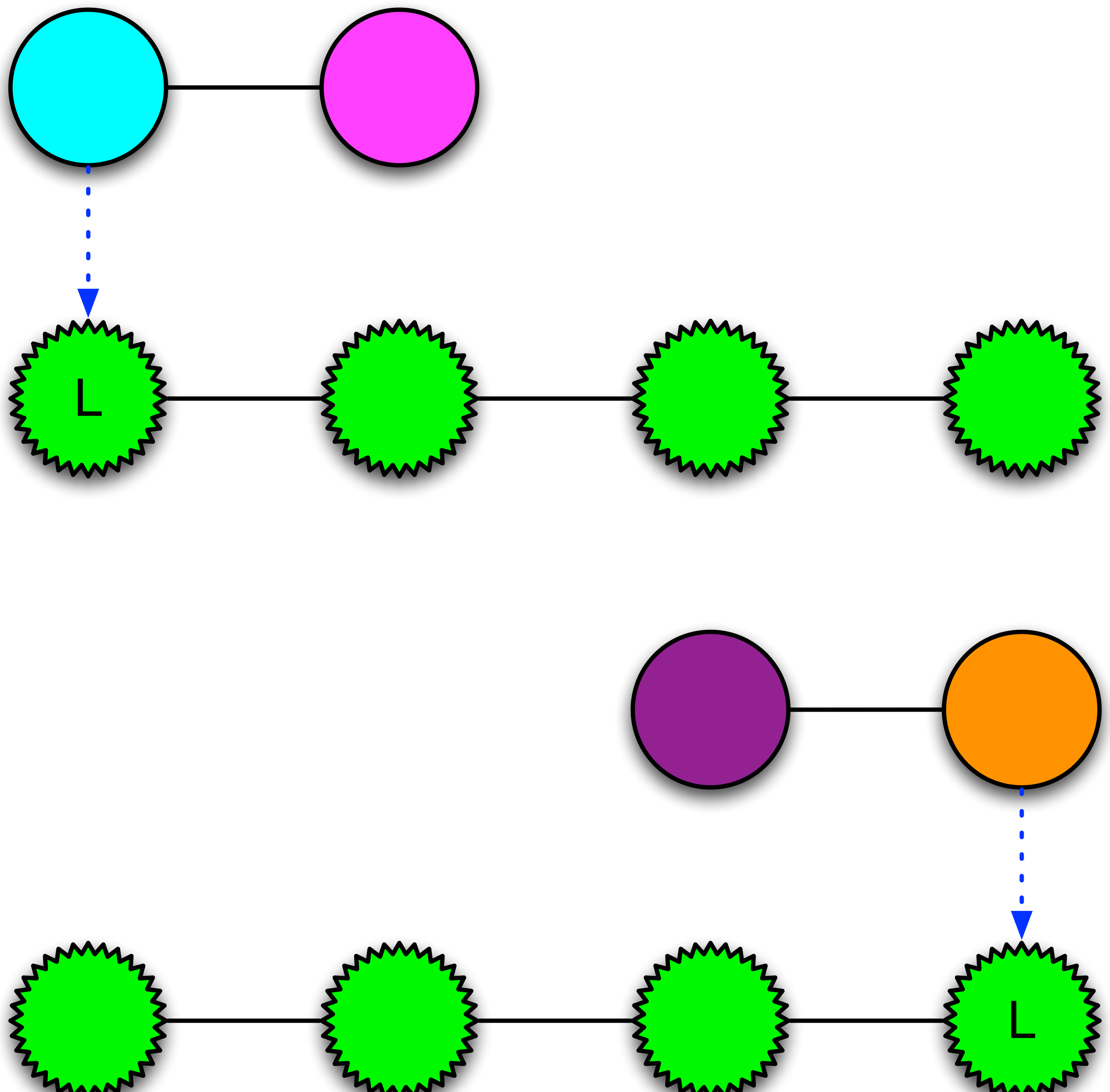


Mappings

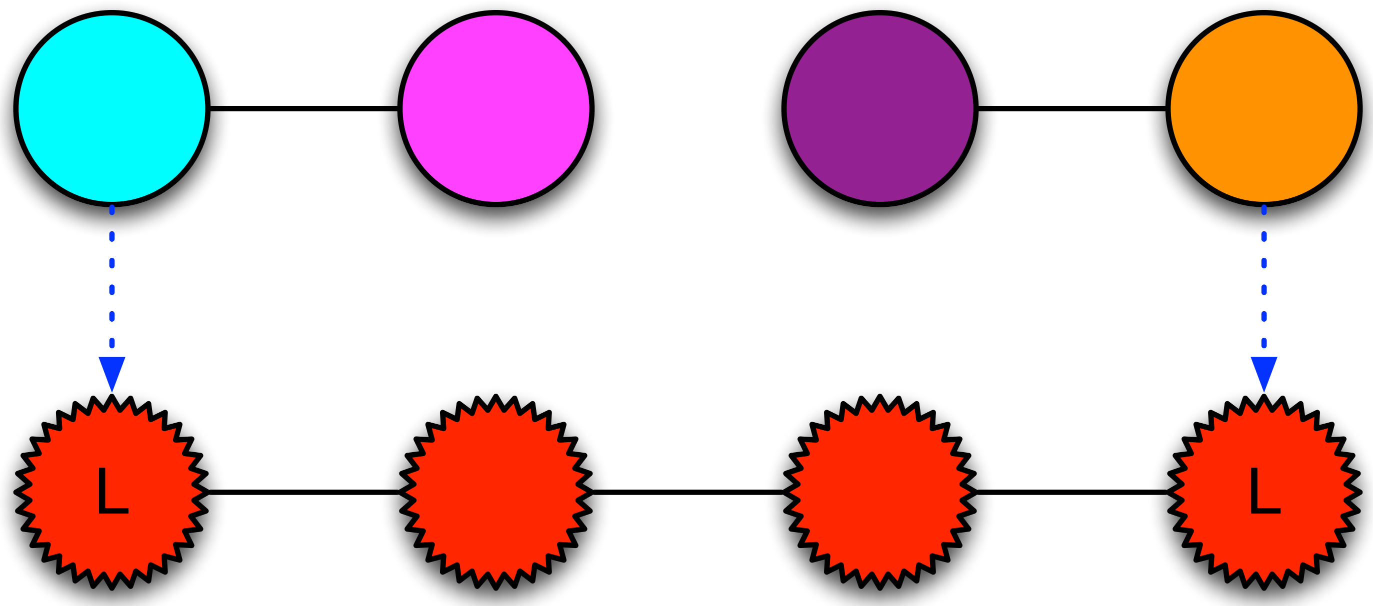
Leader Election



Leader Election

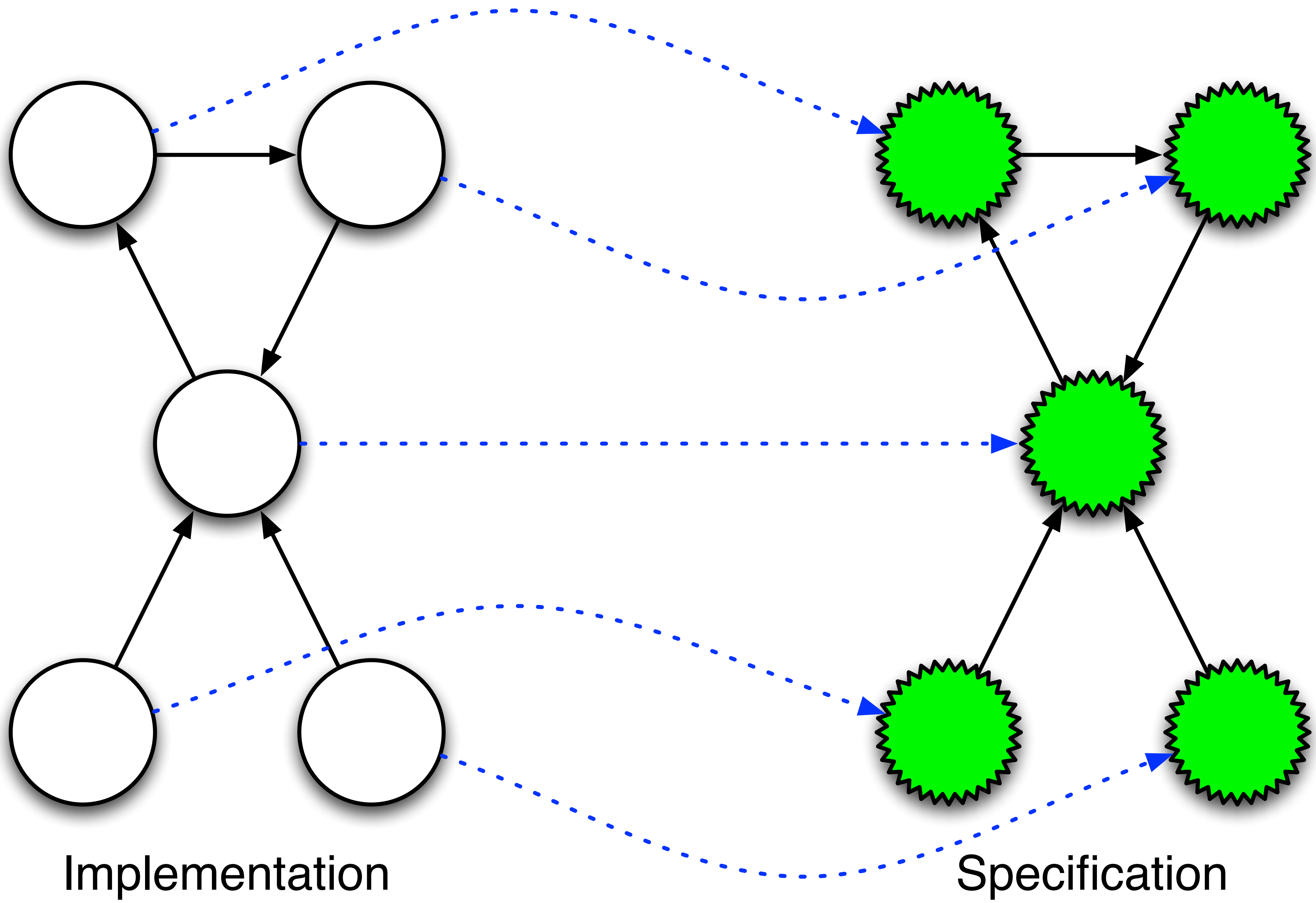


Leader Election



Stabilization to Ideal Specifications

Ideal Specifications

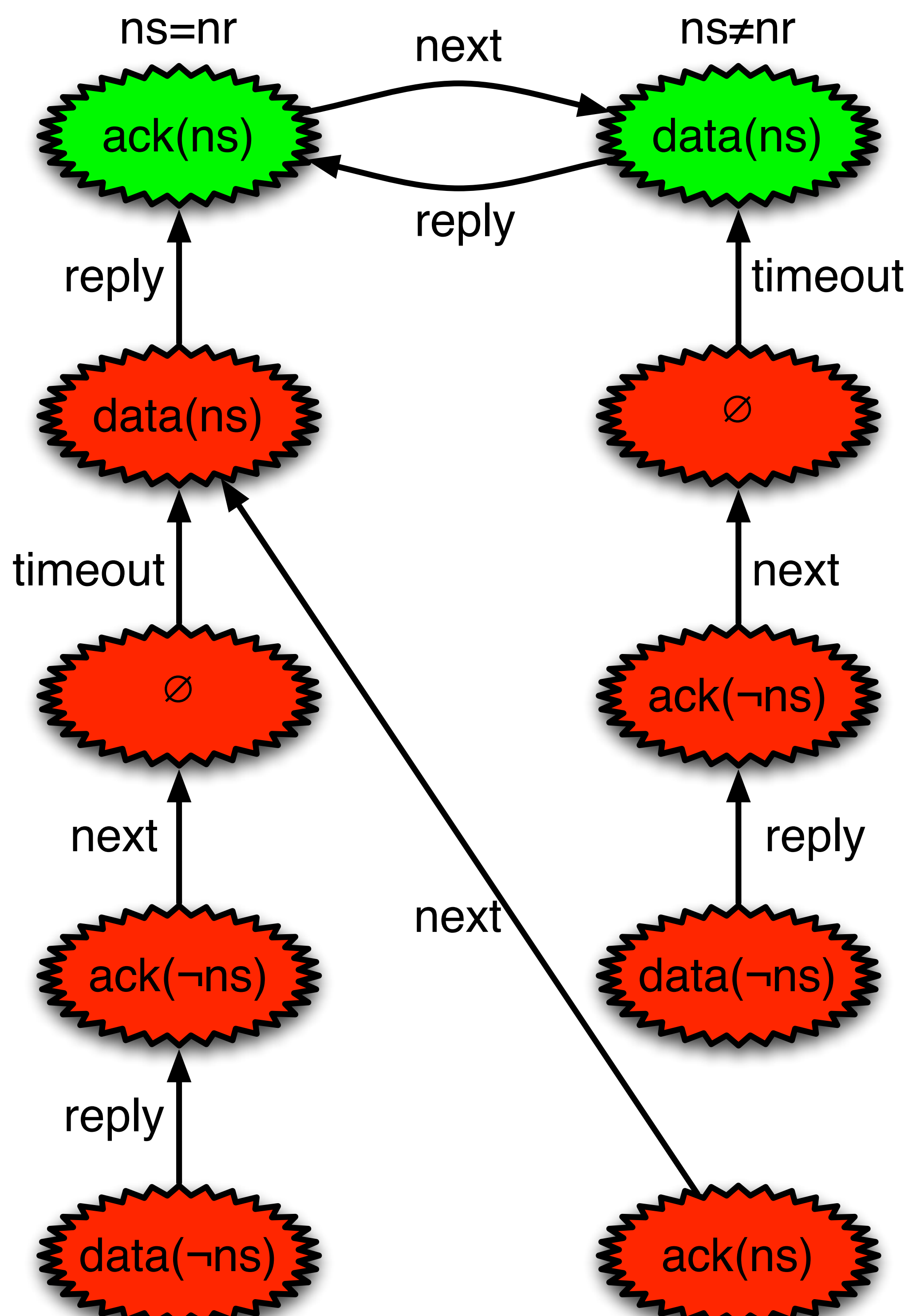


Alternating Bit Protocol

$$
\begin{array}{ll}
\textit{next}: & \textbf{receive}\ ack(nm) \longrightarrow \\
& \qquad \textbf{if}\ nm = ns\ \textbf{then} \\
& \qquad\qquad ns := \neg ns \\
& \qquad\qquad \textbf{send}\ data(ns) \\
\textit{timeout}: & \textbf{timeout}() \longrightarrow \textbf{send}\ data(ns) \\
\textit{reply}: & \textbf{receive}\ data(nm) \longrightarrow \\
& \qquad \textbf{if}\ nm \neq nr\ \textbf{then} \\
& \qquad\qquad nr := nm \\
& \qquad \textbf{send}\ ack(nm)
\end{array}
$$

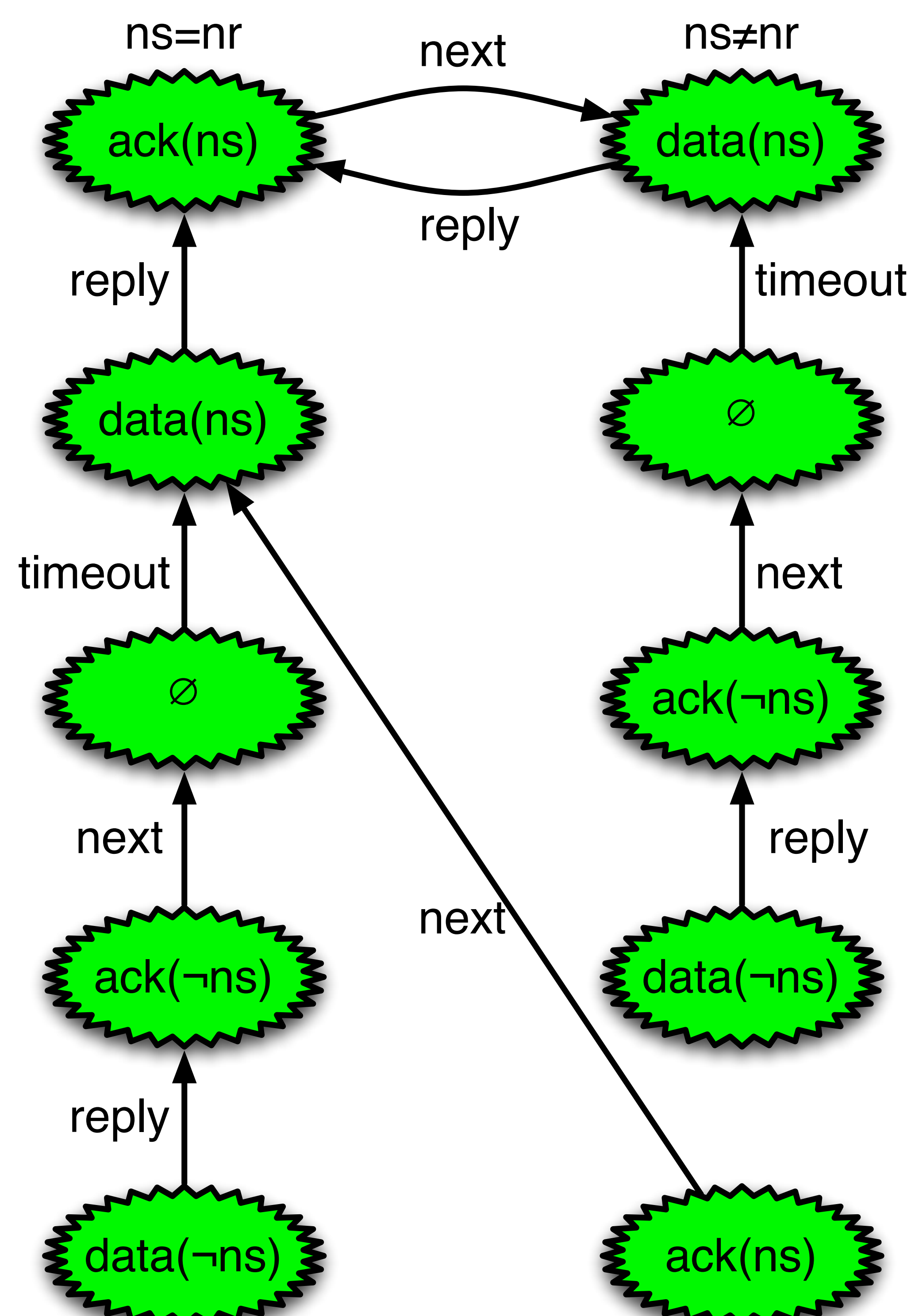
Alternating Bit Protocol

$$
\begin{array}{lll}
\textit{next}: & \textbf{receive } ack(nm) \longrightarrow & \\
 & \quad \textbf{if } nm = ns \textbf{ then} & \\
 & \qquad ns := \neg ns & \\
 & \qquad \textbf{send } data(ns) & \\
\textit{timeout}: & \textbf{timeout}() \longrightarrow \textbf{send } data(ns) & \\
\textit{reply}: & \textbf{receive } data(nm) \longrightarrow & \\
 & \quad \textbf{if } nm \neq nr \textbf{ then} & \\
 & \qquad nr := nm & \\
 & \quad \textbf{send } ack(nm) &
\end{array}
$$

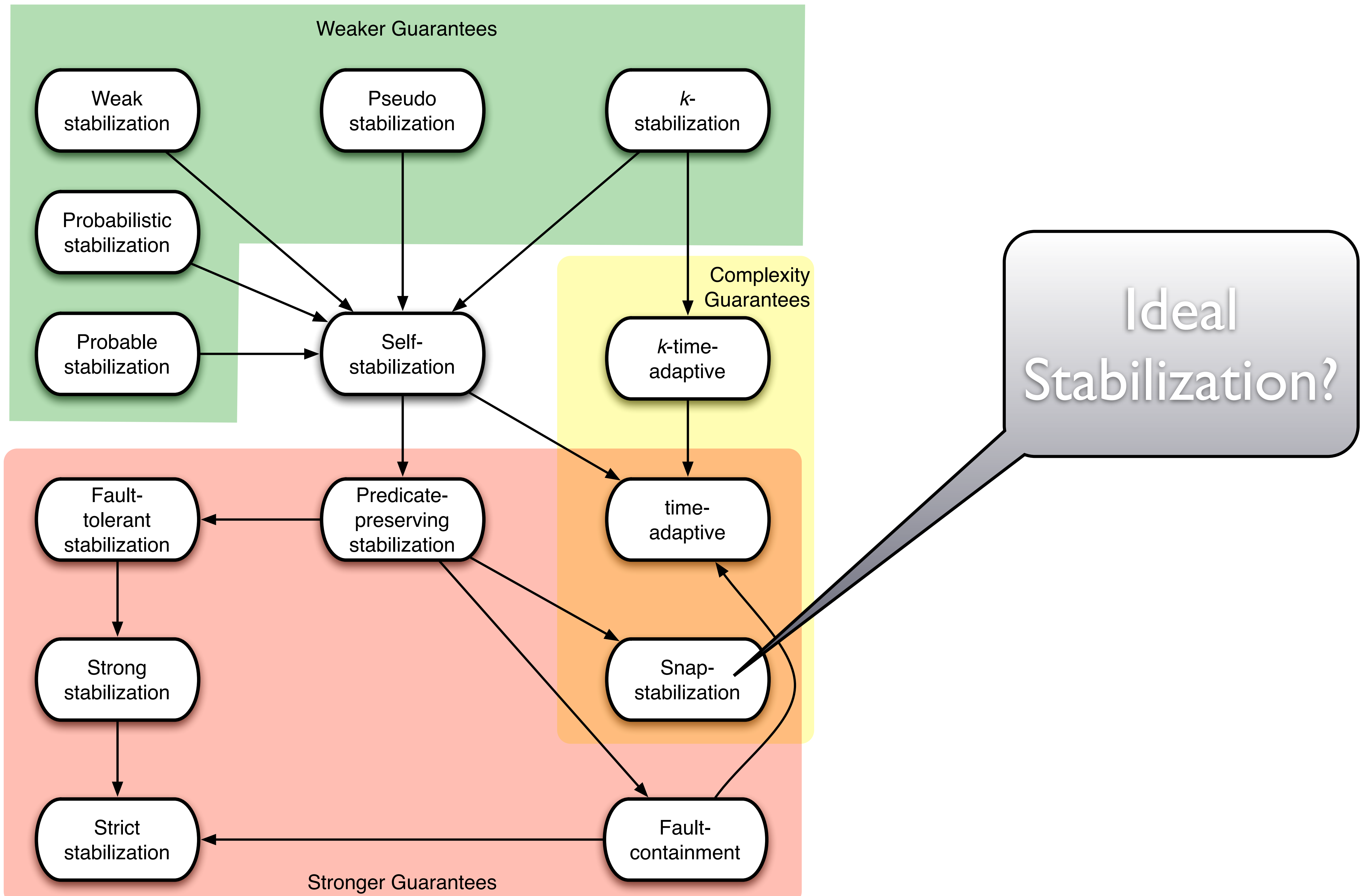


Alternating Bit Protocol

$$
\begin{array}{lll}
\textit{next}: & \textbf{receive } ack(nm) \longrightarrow & \\
 & \quad \textbf{if } nm = ns \textbf{ then} & \\
 & \quad\quad ns := \neg ns & \\
 & \quad\quad \textbf{send } data(ns) & \\
\textit{timeout}: & \textbf{timeout}() \longrightarrow \textbf{send } data(ns) & \\
\textit{reply}: & \textbf{receive } data(nm) \longrightarrow & \\
 & \quad \textbf{if } nm \neq nr \textbf{ then} & \\
 & \quad\quad nr := nm & \\
 & \quad \textbf{send } ack(nm) &
\end{array}
$$

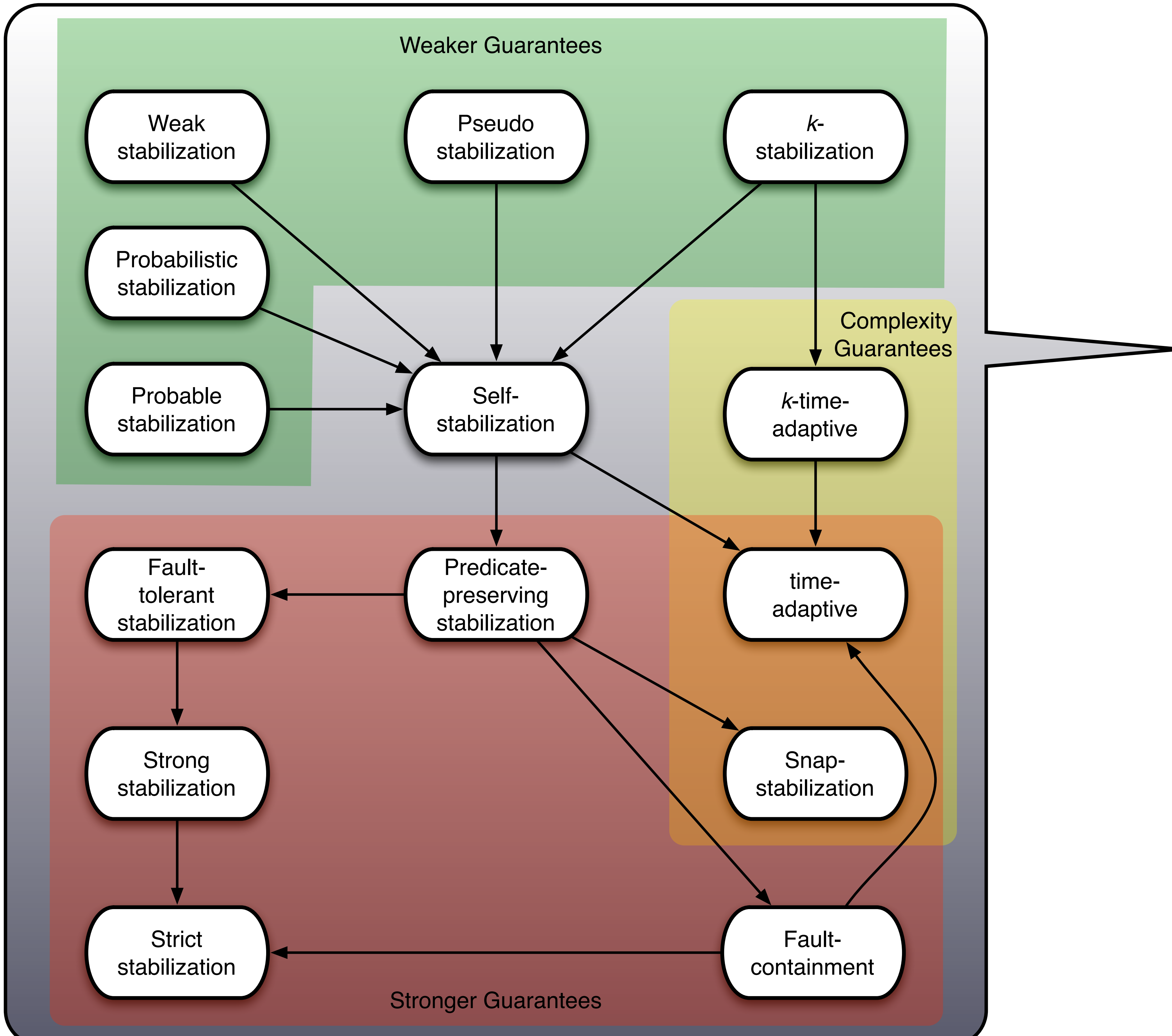


Conclusion



Weaker Guarantees
Weak stabilization
Pseudo stabilization
k-stabilization
Probabilistic stabilization
Probable stabilization
Self-stabilization
Complexity Guarantees
k-time-adaptive
Fault-tolerant stabilization
Predicate-preserving stabilization
time-adaptive
Strong stabilization
Snap-stabilization
Strict stabilization
Fault-containment
Stronger Guarantees
Ideal Stabilization?

Conclusion



Ideal Stabilization

Ideal Stabilization

- New way of **reasoning** about distributed fault-tolerance
- Abitrary degree of precision when **specifying** the system behavior after transient faults occur
- **Composition** is easy
- **Assertional** *vs.* operational proofs

Thank You