MAKING NETWORKS ROBUST TO COMPONENT FAILURE

A Dissertation Outline presented by Daniel Gyllstrom
University of Massachusetts Amherst USA
Advisor: Jim Kurose

2/8/13

Talk Outline

Talk Outline

- thesis introduction

Talk Outline

- thesis introduction
- describe 3 technical chapters

Talk Outline

- thesis introduction
- describe 3 technical chapters
  - smart grid sensor failure

Talk Outline

- thesis introduction
- describe 3 technical chapters
  - smart grid sensor failure
  - failed communication links in a smart grid

Talk Outline

- thesis introduction
- describe 3 technical chapters
  - smart grid sensor failure
  - failed communication links in a smart grid
  - malicious nodes injecting false ~~routing~~ state

Talk Outline

- thesis introduction
- describe 3 technical chapters
  - smart grid sensor failure
  - failed communication links in a smart grid
  - malicious nodes injecting false routing state
- outline for future work and conclusions

Thesis Problem Statement

How can networks -- the Internet and networked cyber-physical systems -- be made more robust to component failure?

3 Component Failure Problems

3 Component Failure Problems

1. failure of critical power grid sensors

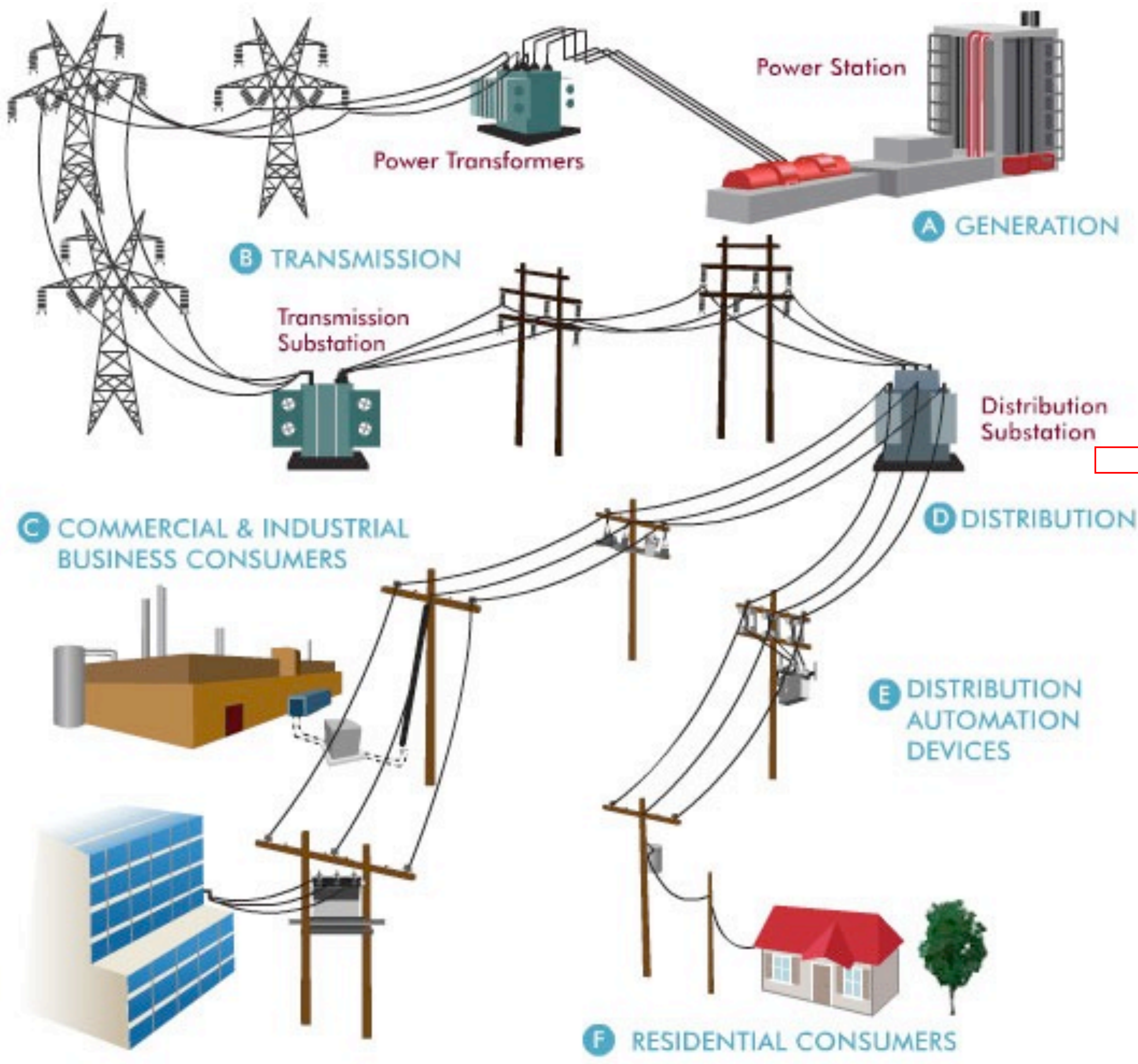
3 Component Failure Problems

1. failure of critical power grid sensors
2. link failures in a power grid communication network

3 Component Failure Problems

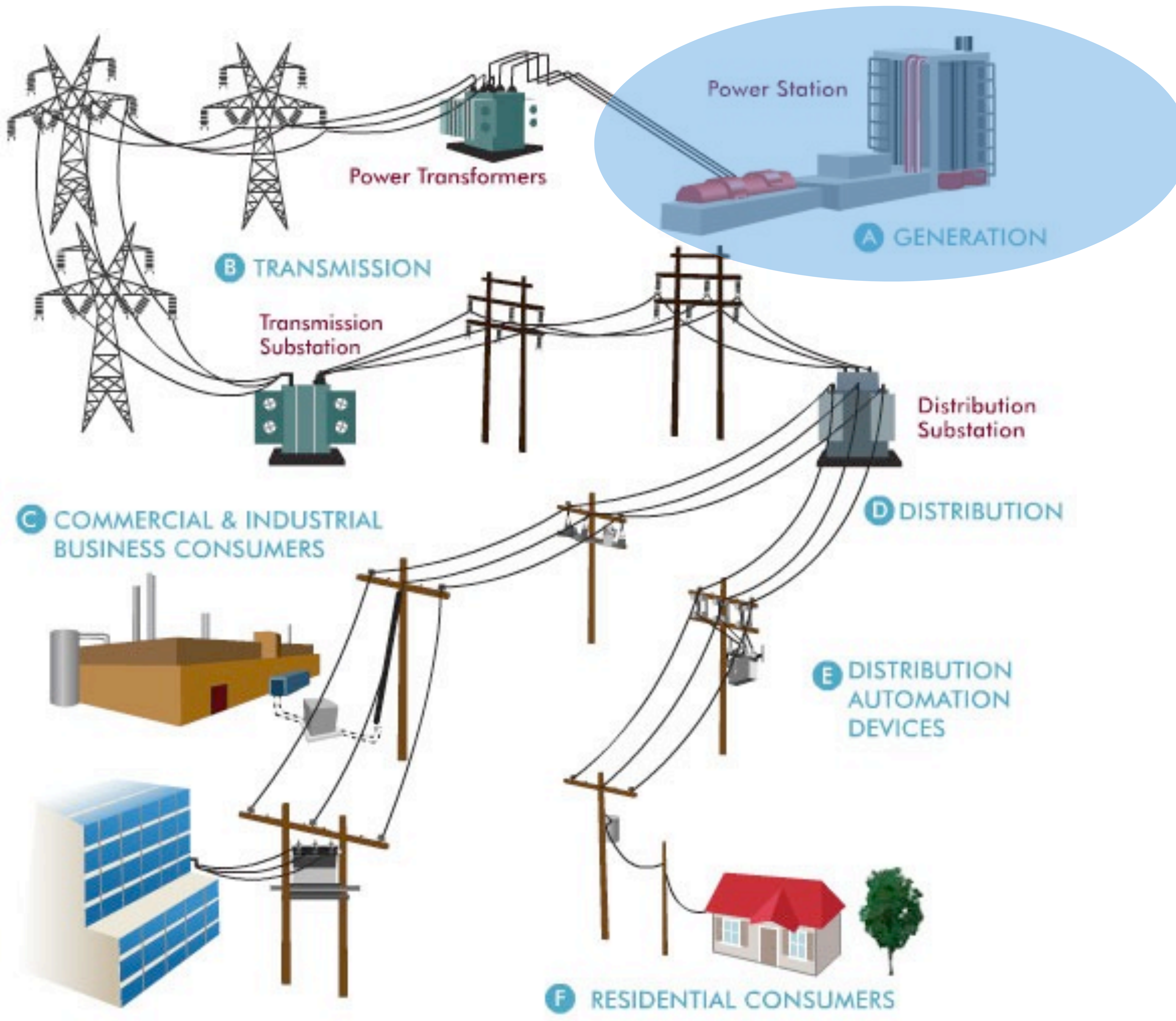
1. failure of critical power grid sensors
2. link failures in a power grid communication network
3. router failure in traditional communication networks

Power and Smart Grid Definition

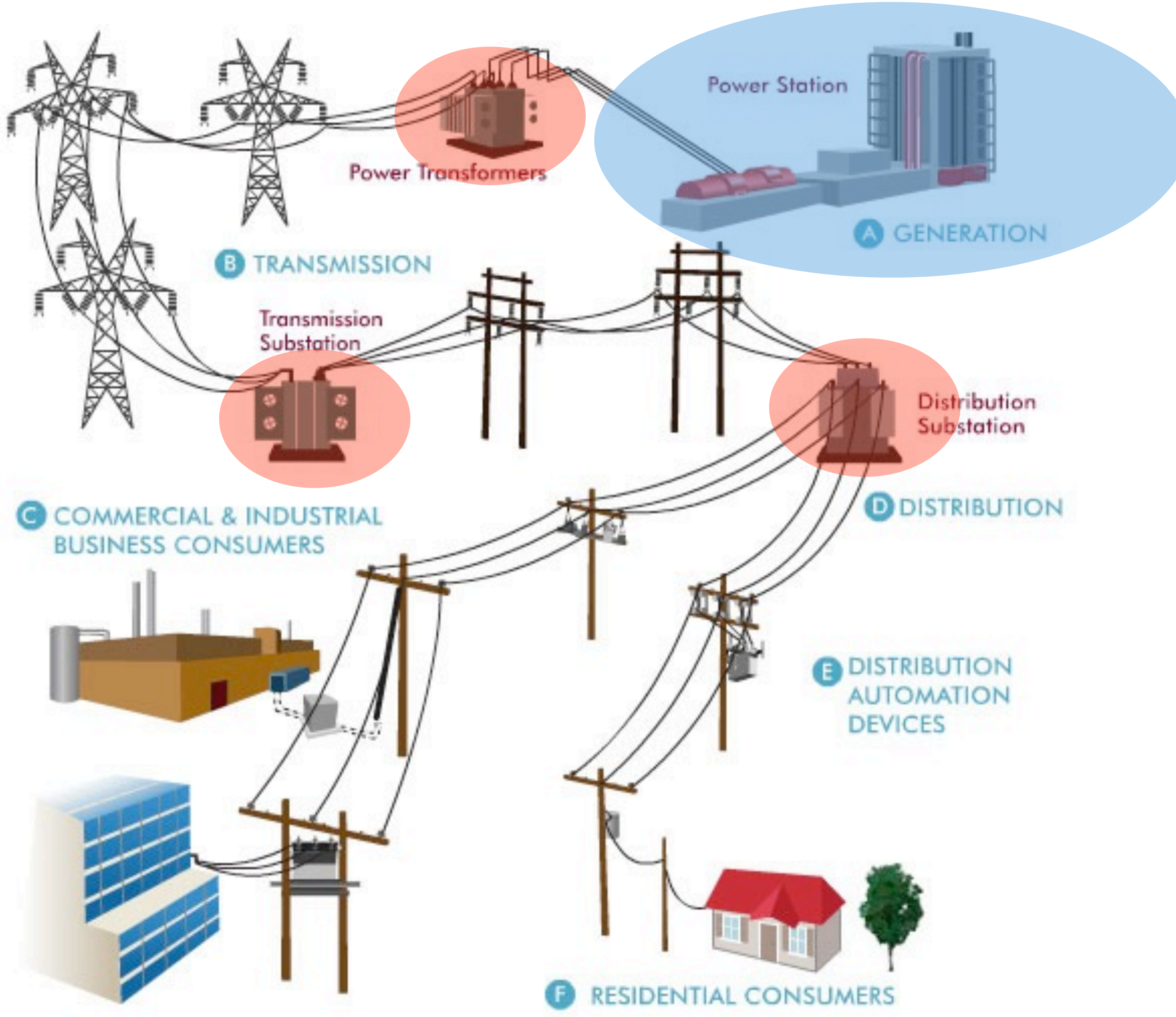


THINK OF A BUS AS A NODE")

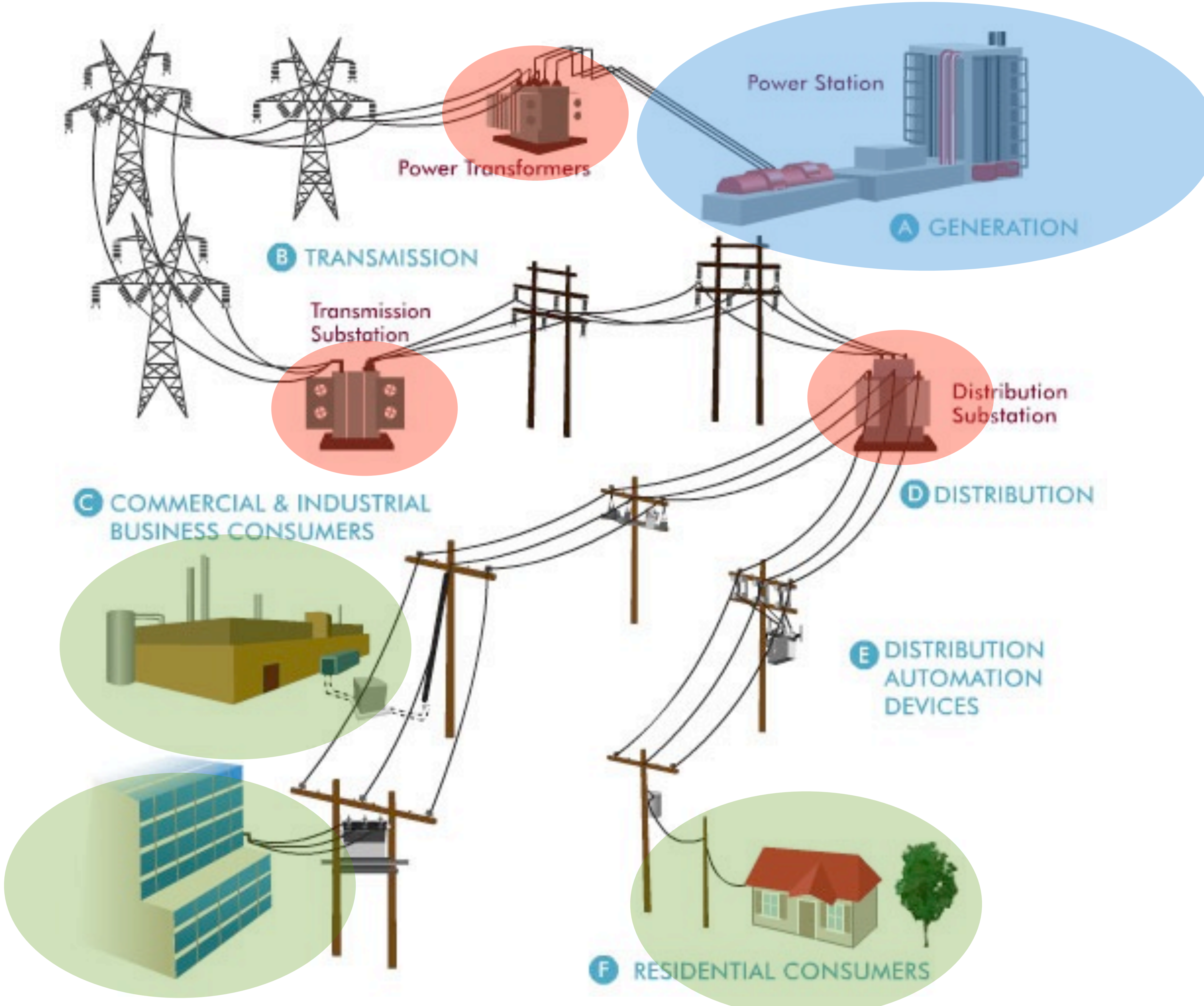
Power and Smart Grid Definition



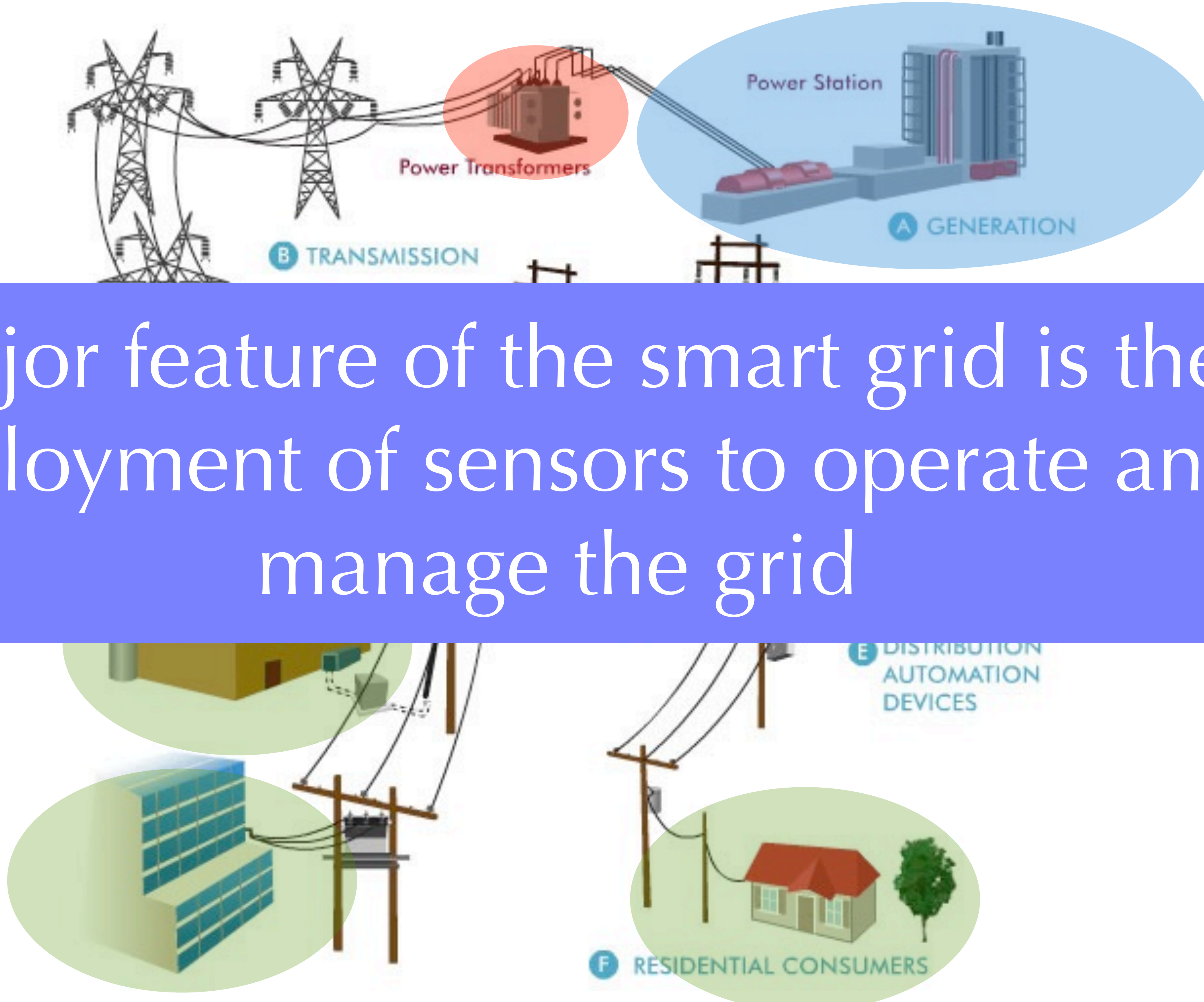
Power and Smart Grid Definition



Power and Smart Grid Definition



Power and Smart Grid Definition



major feature of the smart grid is the deployment of sensors to operate and manage the grid

PMU: Smart Grid Sensor

- Phasor Measurement Unit (PMU)
- high frequency voltage and current measurements
  - measures the “pulse” of the power grid

Ch 1+2: Smart Grid Failures

placement and failure

- Ch 1: PMU sensor failure
- Ch 2: link failures in communication network used to disseminate PMU measurements

Ch 1+2: Smart Grid Failures

- Ch 1: PMU sensor failure
- Ch 2: link failures in communication network used to disseminate PMU measurements

these failures can cause critical errors in smart grid applications used to operate the grid

theguardian

India blackouts leave 700 million without power

Power cuts plunge 20 of India's 28 states into darkness as energy suppliers fail to meet growing demand

Helen Pidd in Delhi
The Guardian, Tuesday 31 July 2012 10.48 EDT

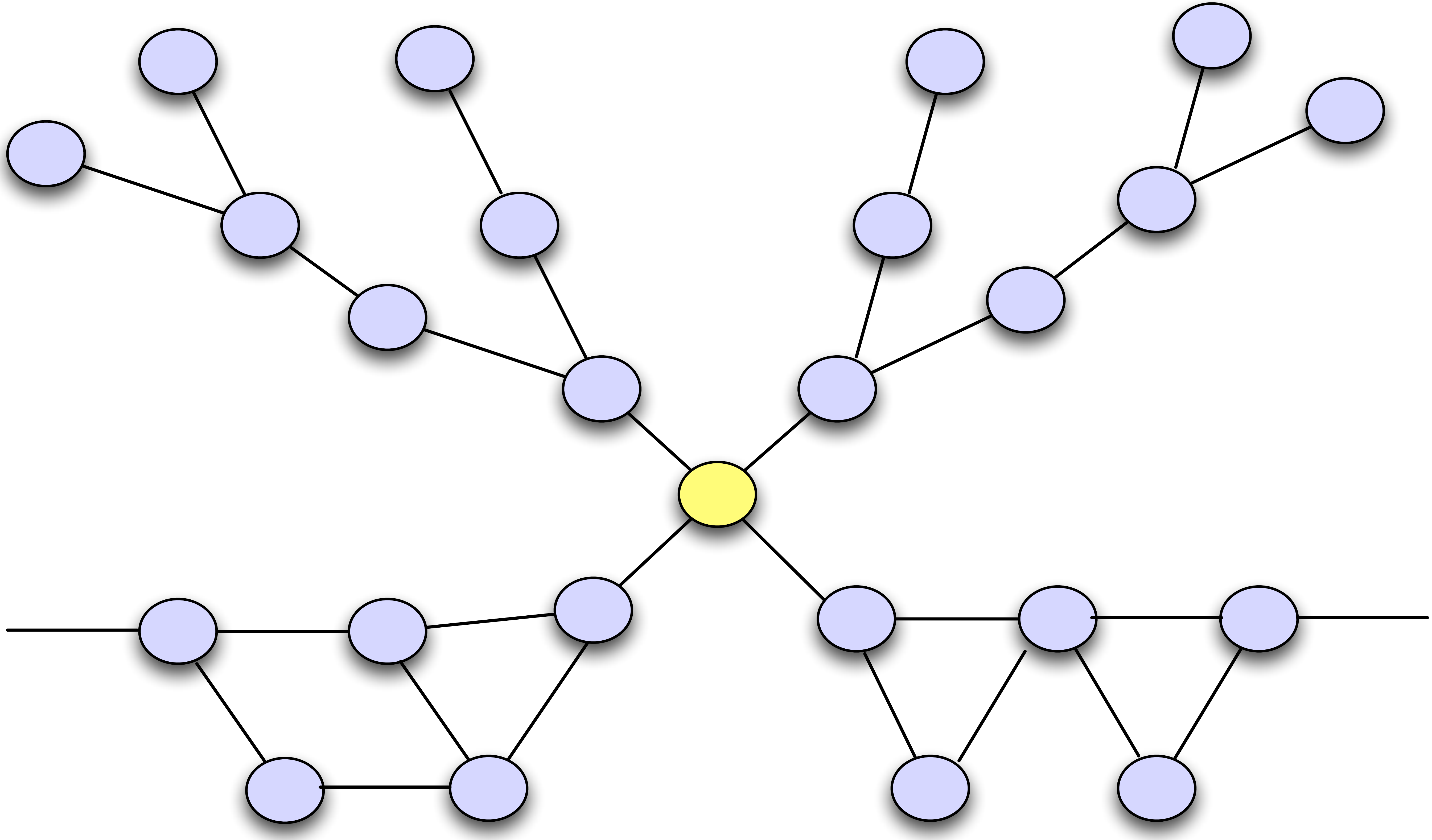
More than 700 million people in India have been left without power in the world's worst blackout of recent times, leading to fears that protests and even riots could follow if the country's electricity supply continues to fail to meet growing demand.

Twenty of India's 28 states were hit by power cuts, along with the capital, New Delhi, when three of the country's five electricity grids failed at lunchtime.

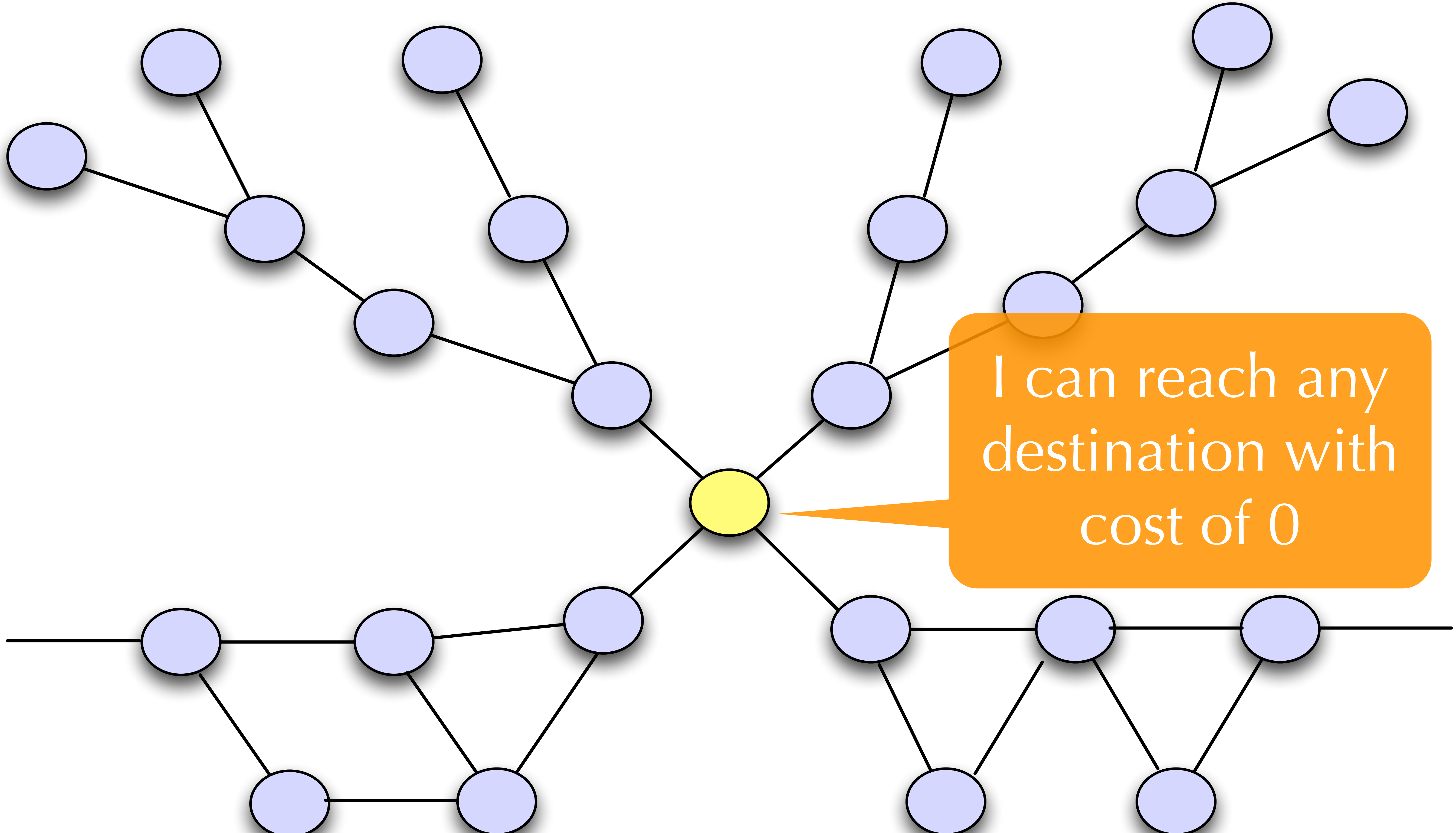
As engineers struggled for hours to fix the problem, hundreds of trains failed, leaving passengers stranded along thousands of miles of track from Kashmir in the north to Nagaland on the eastern border with Burma.

Traffic lights went out, causing jams in New Delhi, Kolkata and other cities. Surgical operations were cancelled across the country, with nurses at one hospital just outside Delhi having to operate life-saving equipment manually when back-up generators failed.

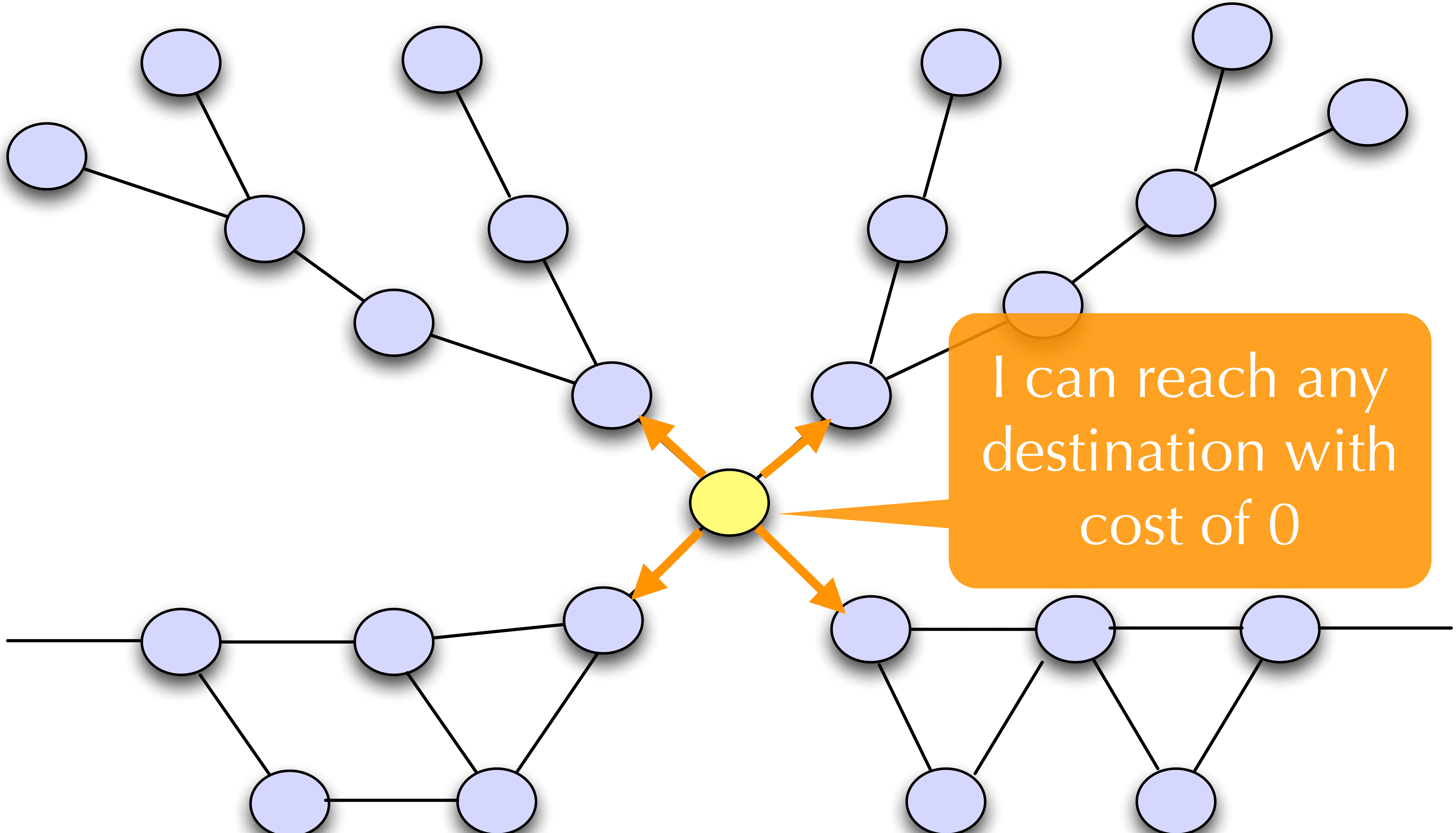
Ch 3: Network Router Failure



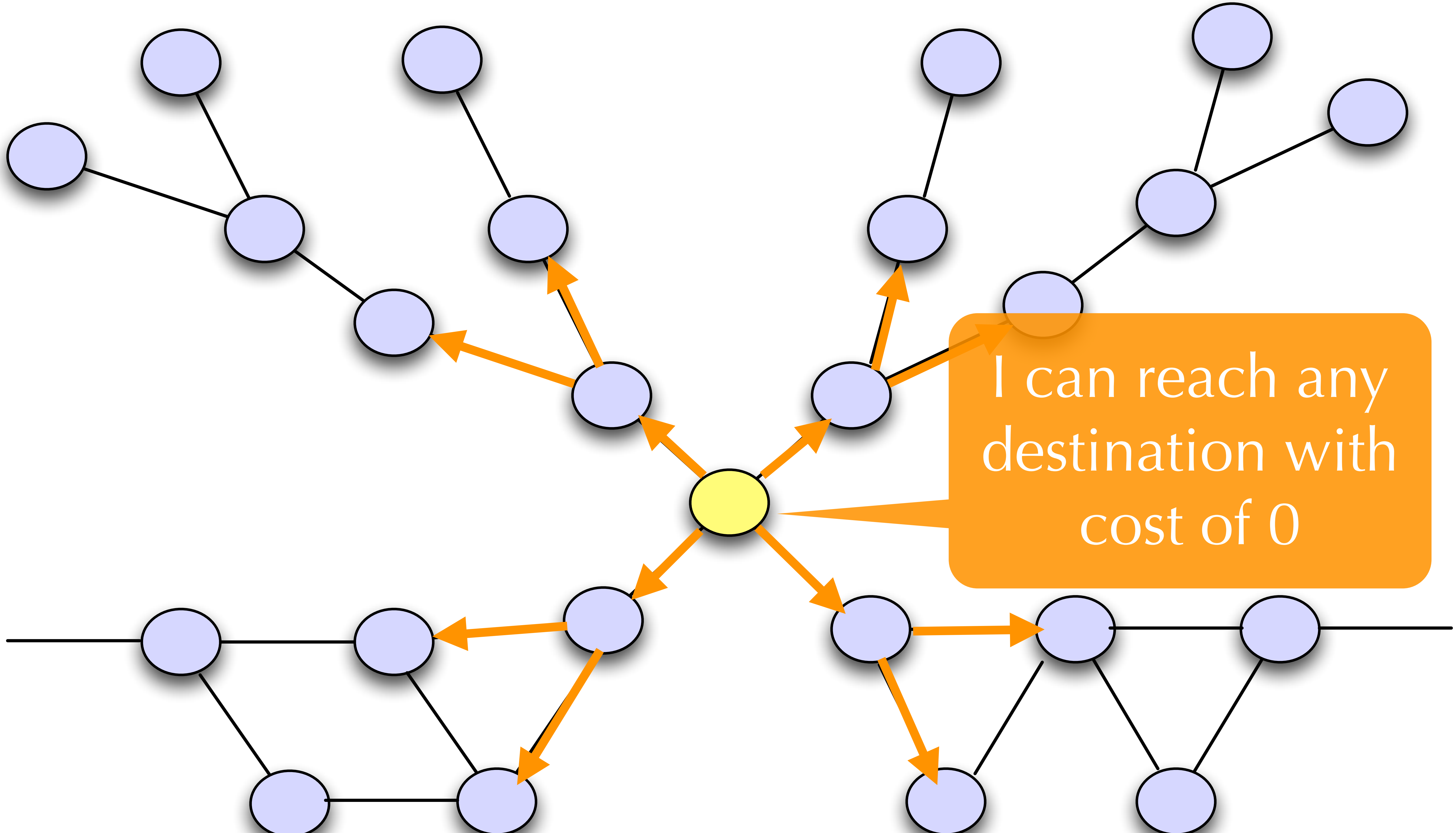
Ch 3: Network Router Failure



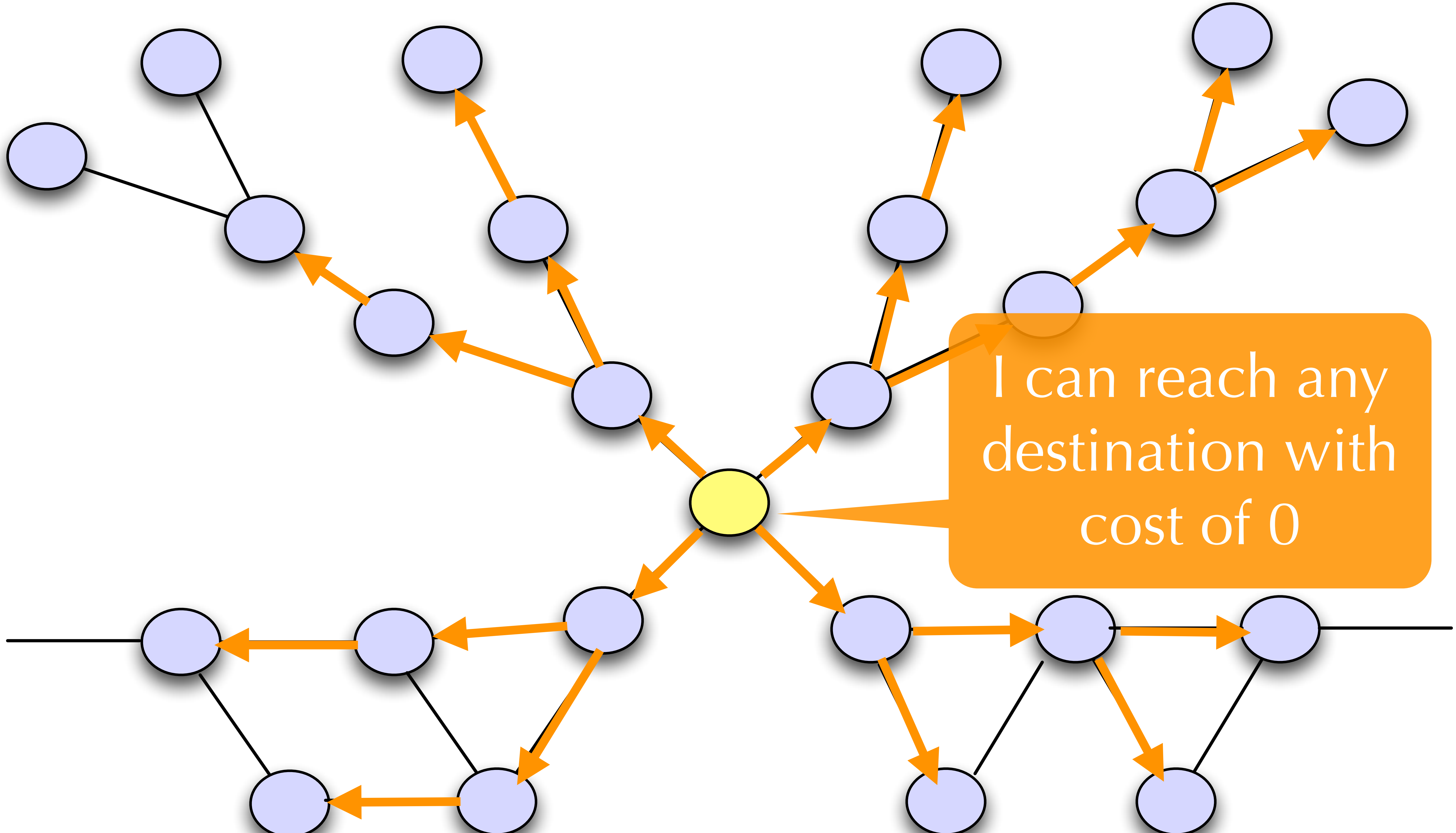
Ch 3: Network Router Failure



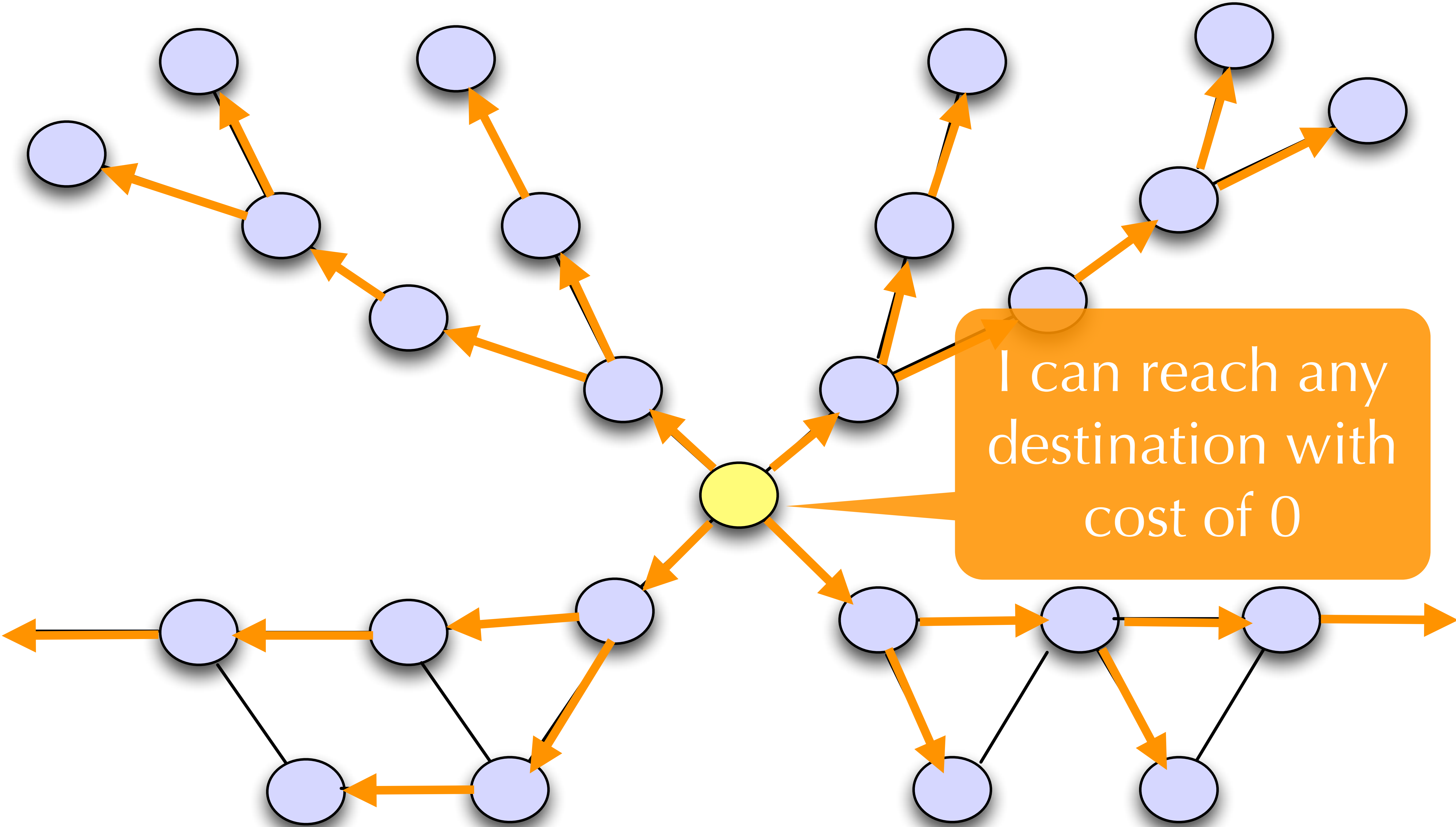
Ch 3: Network Router Failure



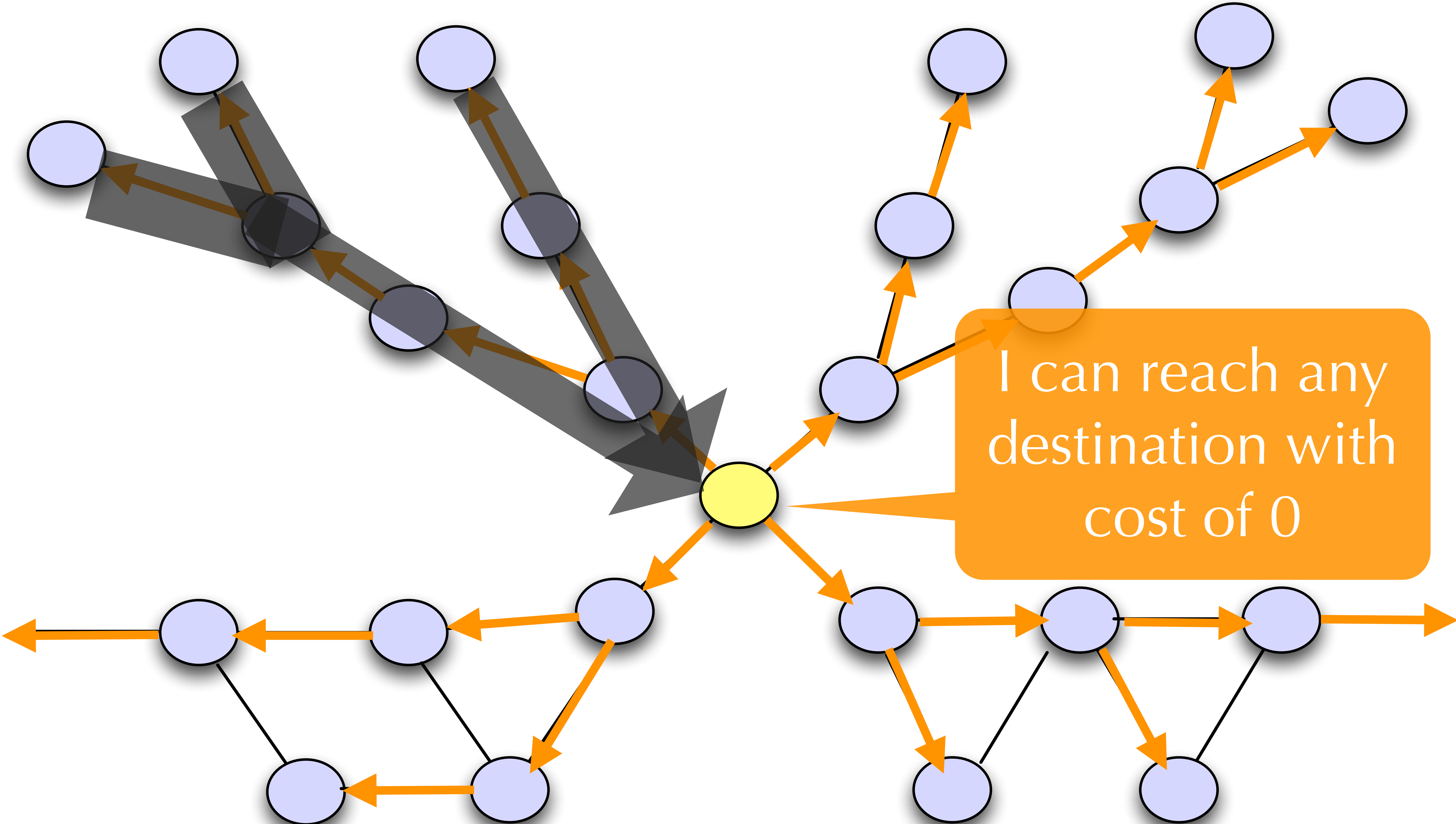
Ch 3: Network Router Failure



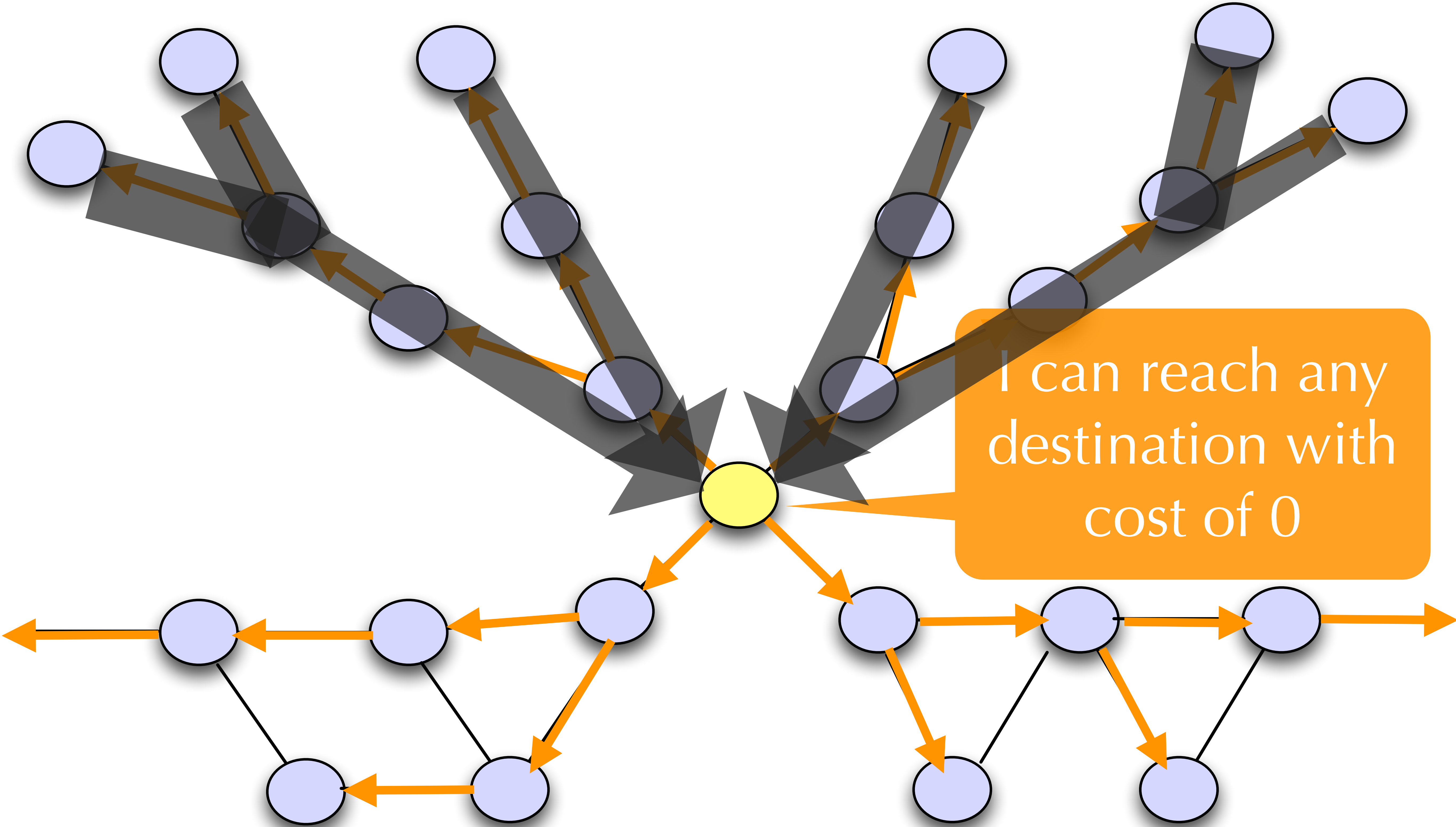
Ch 3: Network Router Failure



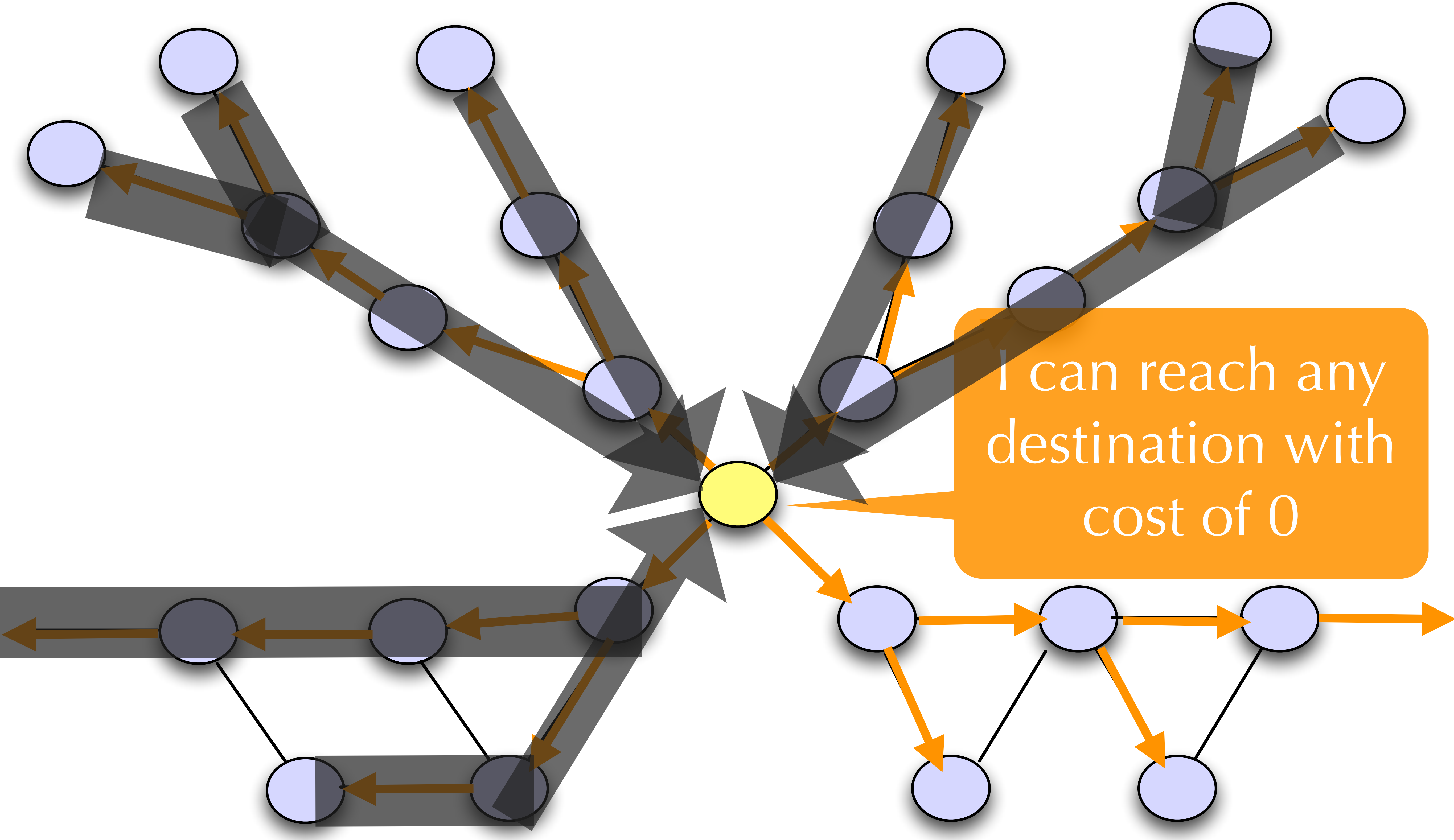
Ch 3: Network Router Failure



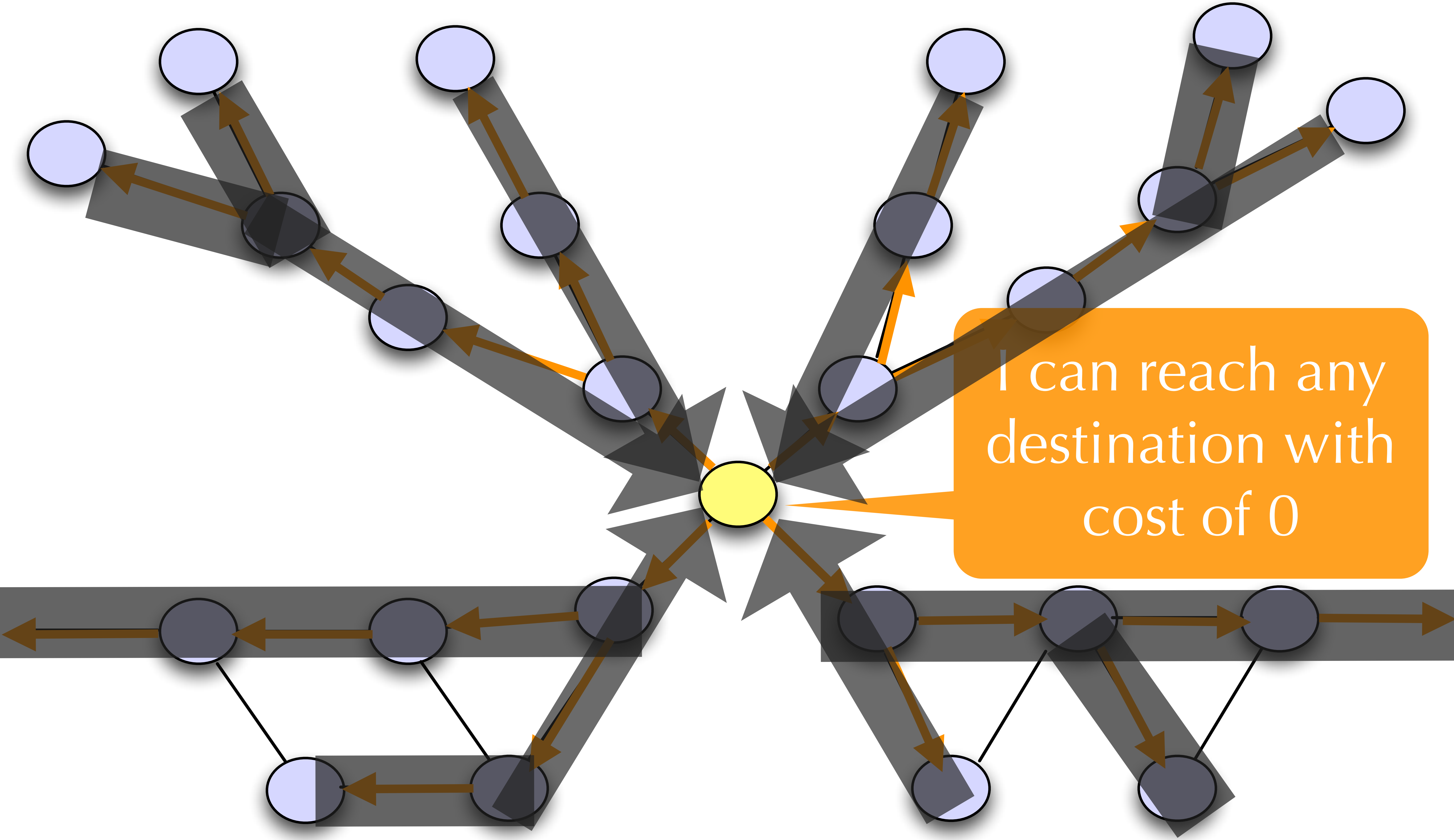
Ch 3: Network Router Failure



Ch 3: Network Router Failure



Ch 3: Network Router Failure



Reviews News Download CNET TV How To Deals

CNET › News › Communications

April 25, 1997 7:00 PM PDT

Router glitch cuts Net access

By CNET News.com Staff
Staff Writer, CNET News

Related Stories

Net blackout hits some regions
April 25, 1997

Software blamed for AOL blackout
February 5, 1997

WorldNet service restored
November 9, 1996

Web gets an Olympian workout
July 13, 1996

What started out as a router glitch at a small Internet service provider in Virginia today triggered a major outage in Internet access across the country, lasting more than two hours in some places.

The problem started this morning at 8:30 a.m. PT when MAI Network Services, an ISP headquartered in a McLean, Virginia, unwittingly passed some bad router information from one of its customers onto Sprint, one of the largest Internet backbone operators in North America. Because Sprint's backbone is used by so many other smaller ISPs, the router problem was echoed, causing temporary network outages across the country and, perhaps, internationally.

The outage underscored the fragility of the infrastructure that underlies the global network and how easily a problem with one small ISP can be amplified throughout the Internet. Even so, the Net displayed a remarkable resilience that seems to disprove its doomsayers, who have predicted that the network is on the verge of collapse.

"This particular thing was a confluence of two or three things happening--human error, bug, and some policy problems--that all came together on the same day," said Jack Rickard, publisher of *BoardWatch* magazine.

"There are probably a hundred guys in back rooms keeping this stuff together, just barely," Ricard said of the Internet.

Automated Recovery Is Needed

- automated recovery needed to reduce
  - short-term disruption
  - increase long-term network survivability

Automated Recovery Is Needed

- automated recovery needed to reduce
  - short-term disruption
  - increase long-term network survivability

thesis designs algorithms to make networks robust to these component failures

Unifying The 3 Subproblems

- each problem considers network robustness in the face of component failure
- our solutions
  - preplanned recovery for smart grid apps where reliability is key
  - on-demand recovery for distributed network algorithms

Talk Outline

- thesis introduction
- placement of smart grid sensors to enable measurement error detection
- recovery from failed communication links in a smart grid
- recovery from malicious nodes injecting false routing state
- outline for future work and conclusions

Talk Outline

- thesis introduction
- placement of smart grid sensors to enable measurement error detection
- recovery from failed communication links in a smart grid
- recovery from malicious nodes injecting false routing state
- outline for future work and conclusions

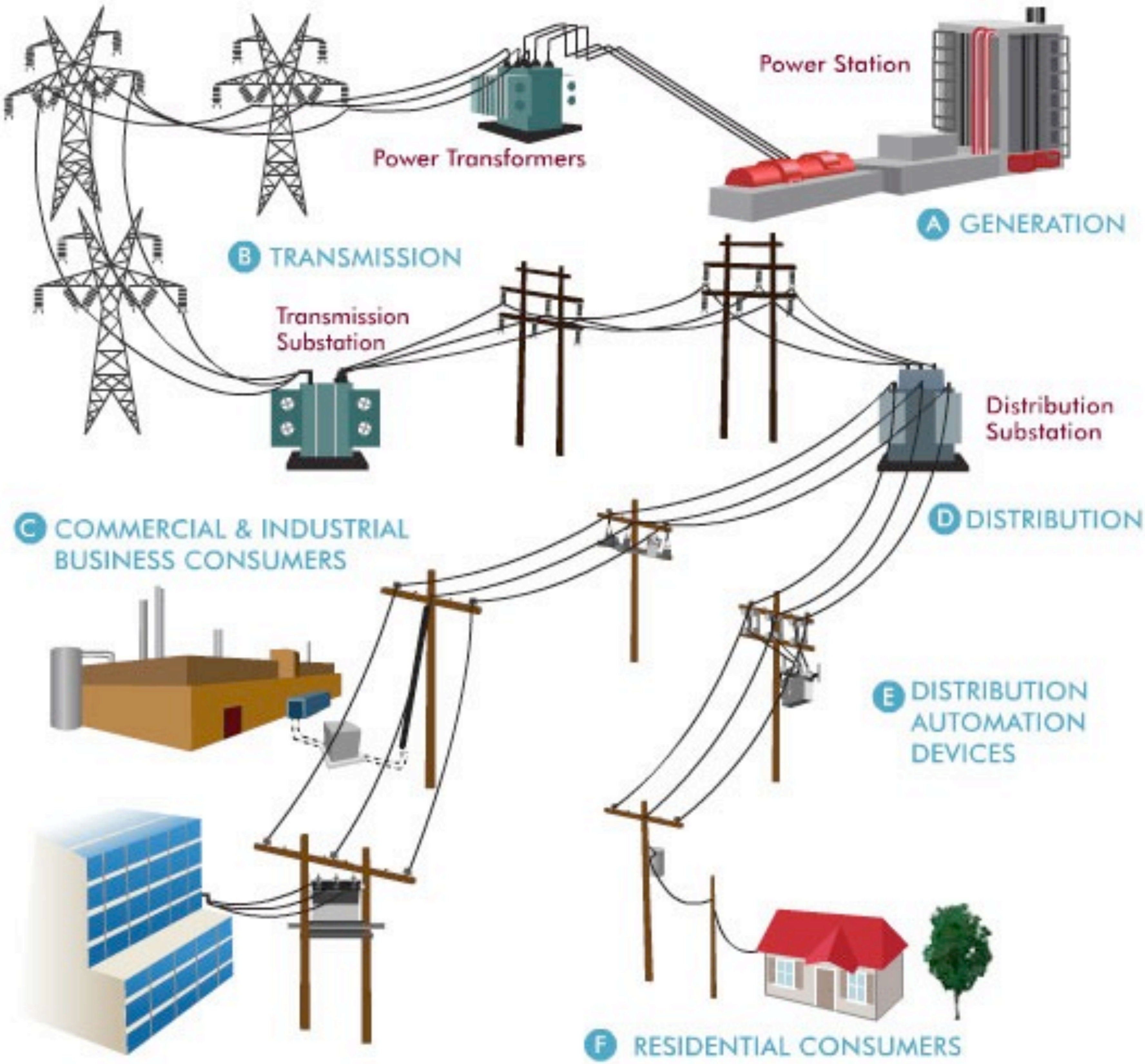
Talk Outline

- thesis introduction
- placement of smart grid sensors to enable measurement error detection
- recovery from failed communication links in a smart grid
- recovery from malicious nodes injecting false routing state
- outline for future work and conclusions

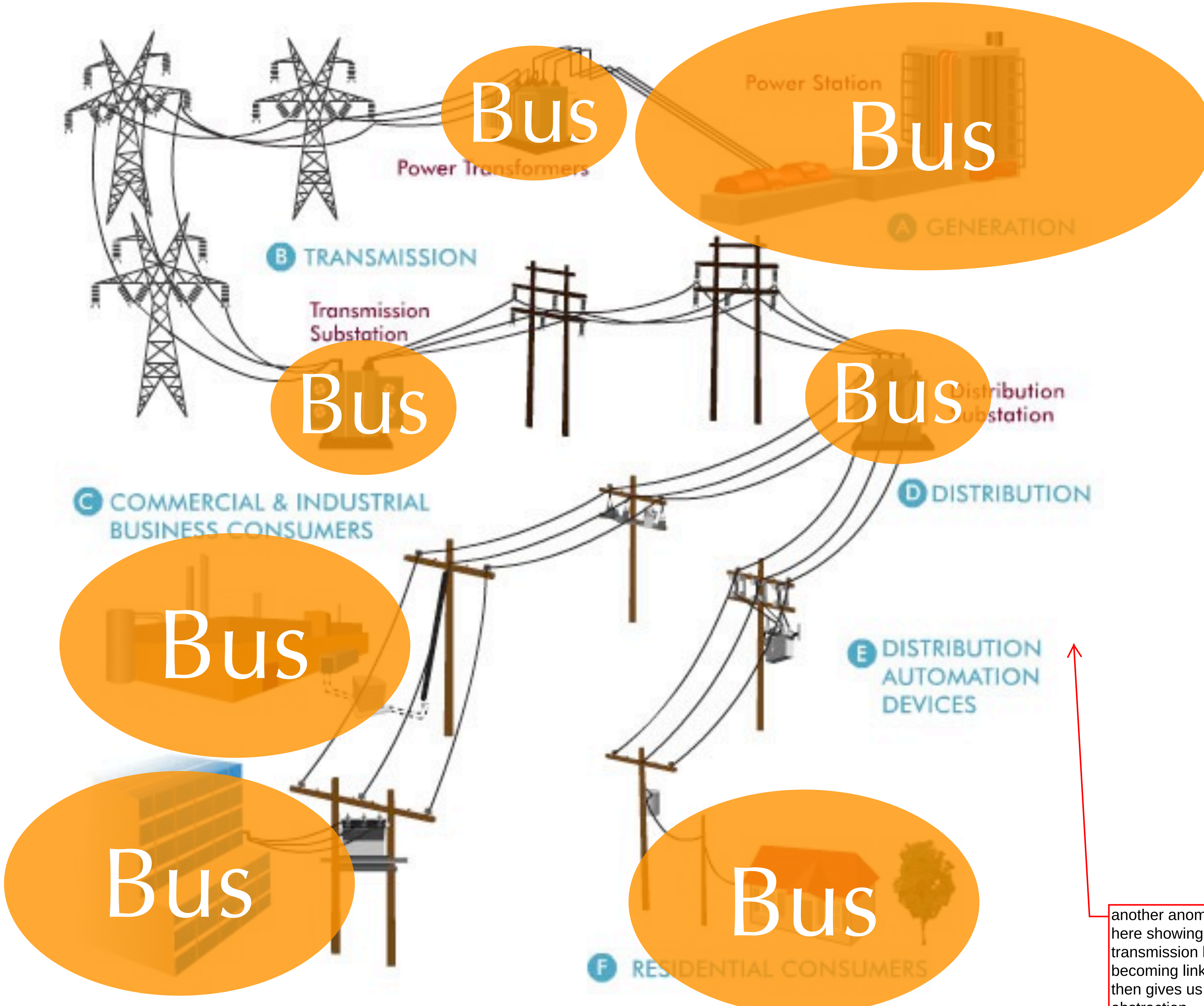
Talk Outline

- thesis introduction
- placement of smart grid sensors to enable measurement error detection
- recovery from failed communication links in a smart grid
- recovery from malicious nodes injecting false routing state
- outline for future work and conclusions

Refresher: Power Grid Definition

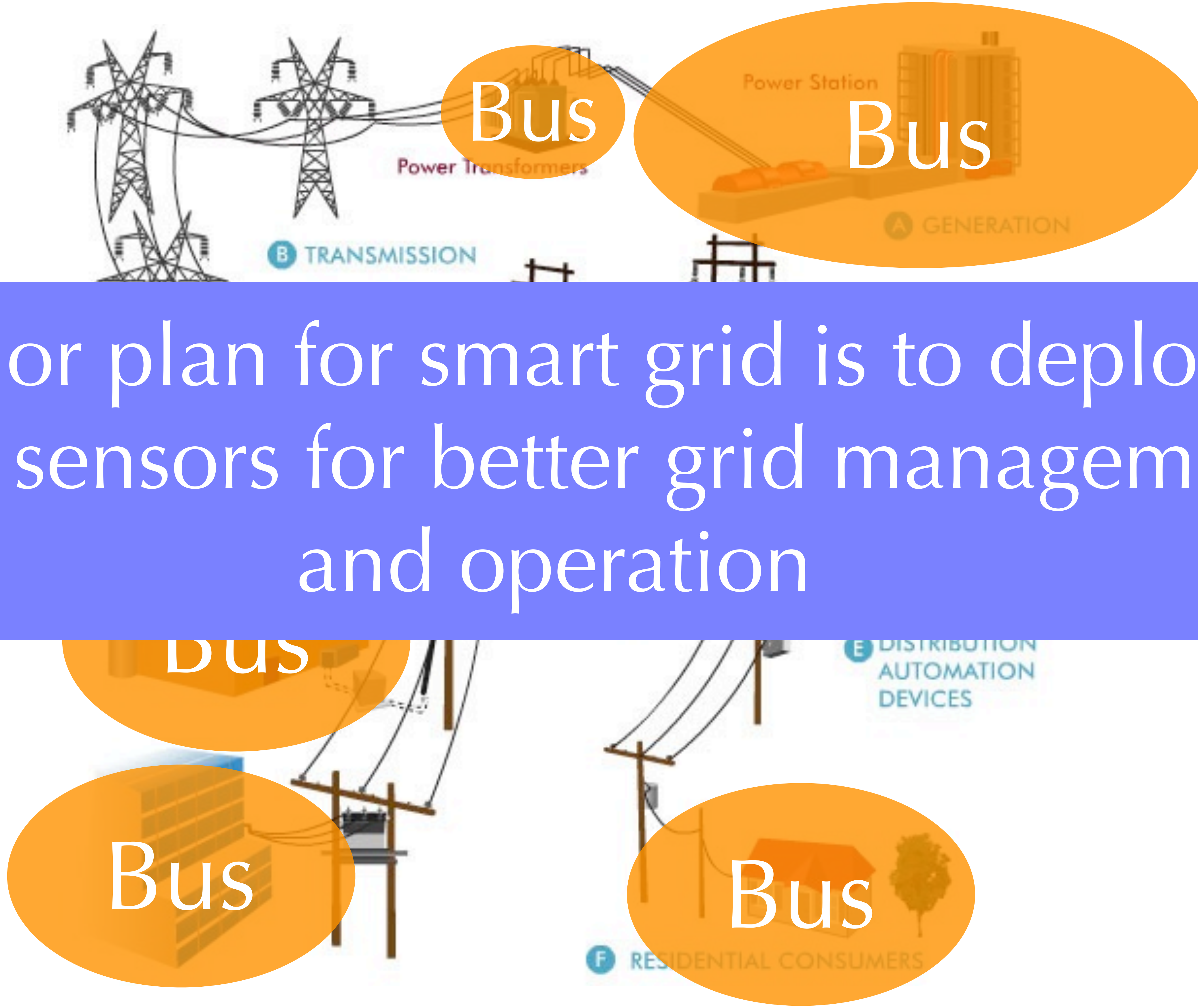


Refresher: Power Grid Definition



another anomiation here showing transmission lines becoming links. THis then gives us the graph abstraction

Refresher: Power Grid Definition



major plan for smart grid is to deploy PMU sensors for better grid management and operation

More on PMU Sensors

- deployed at system buses
- measure voltage and current of system bus and transmission lines
  - high sampling rate (60 samples/sec)
  - measurements synchronized with GPS clock
- instantaneous snapshot of grid

Grid Measurement + Monitoring

where to place PMUs to maximize observability?

D. Brueni and L. Heath. The PMU Placement Problem. SIAM Journal on Discrete Math, 2005

Grid Measurement + Monitoring

where to place PMUs to maximize observability?

- observability rule 1: if a PMU is placed at v then v and its neighbors are observed

D. Brueni and L. Heath. The PMU Placement Problem. SIAM Journal on Discrete Math, 2005

Grid Measurement + Monitoring

where to place PMUs to maximize observability?

- observability rule 1: if a PMU is placed at v then v and its neighbors are observed
- observability rule 2: if a node is observable and all neighbors except one are observed, then all neighbors are observable

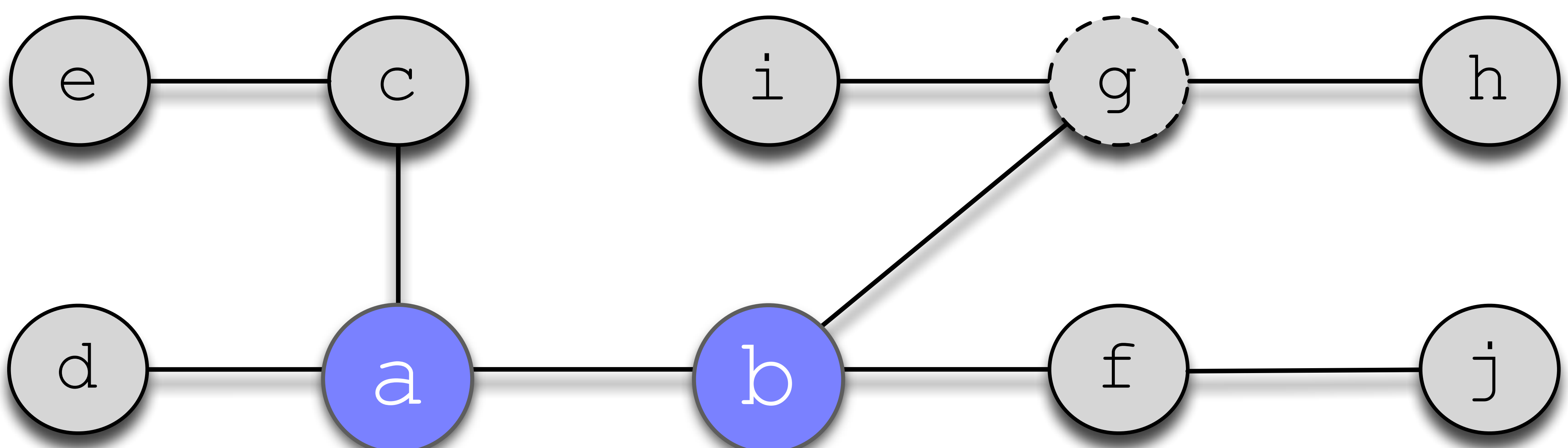
you need to say what it means to be "observable" !!!!!!!

D. Brueni and L. Heath. The PMU Placement Problem. SIAM Journal on Discrete Math, 2005

Grid Measurement + Monitoring

where to place PMUs to maximize observability?

- observability rule 1: if a PMU is placed at v then v and its neighbors are observed
- observability rule 2: if a node is observable and all neighbors except one are observed, then all neighbors are observable

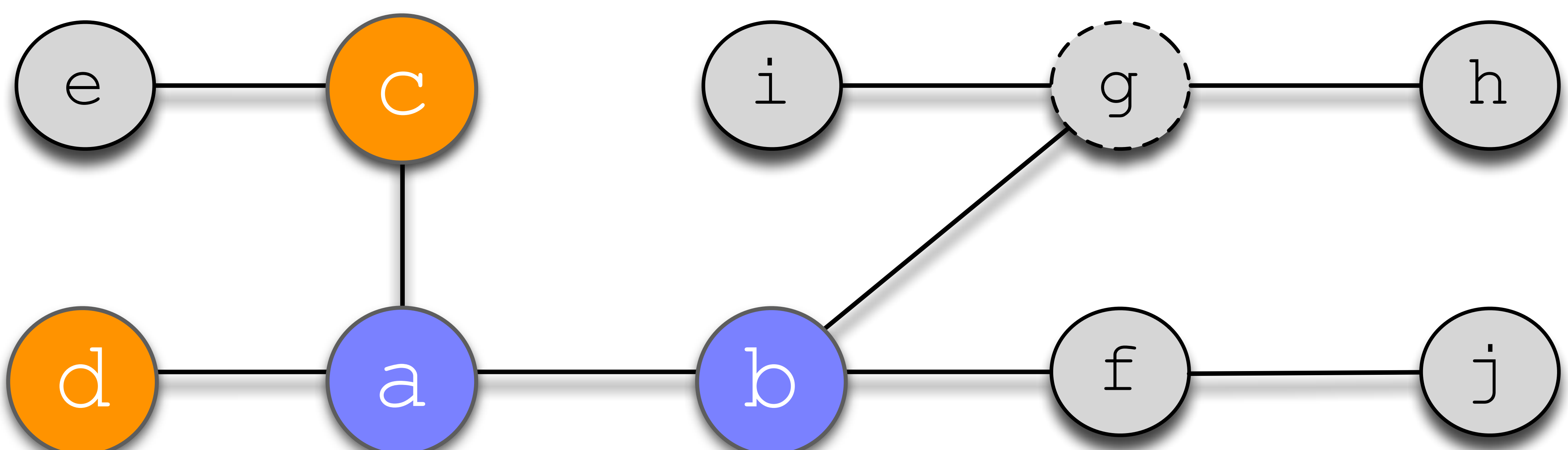


D. Brueni and L. Heath. The PMU Placement Problem. SIAM Journal on Discrete Math, 2005

Grid Measurement + Monitoring

where to place PMUs to maximize observability?

- observability rule 1: if a PMU is placed at v then v and its neighbors are observed
- observability rule 2: if a node is observable and all neighbors except one are observed, then all neighbors are observable

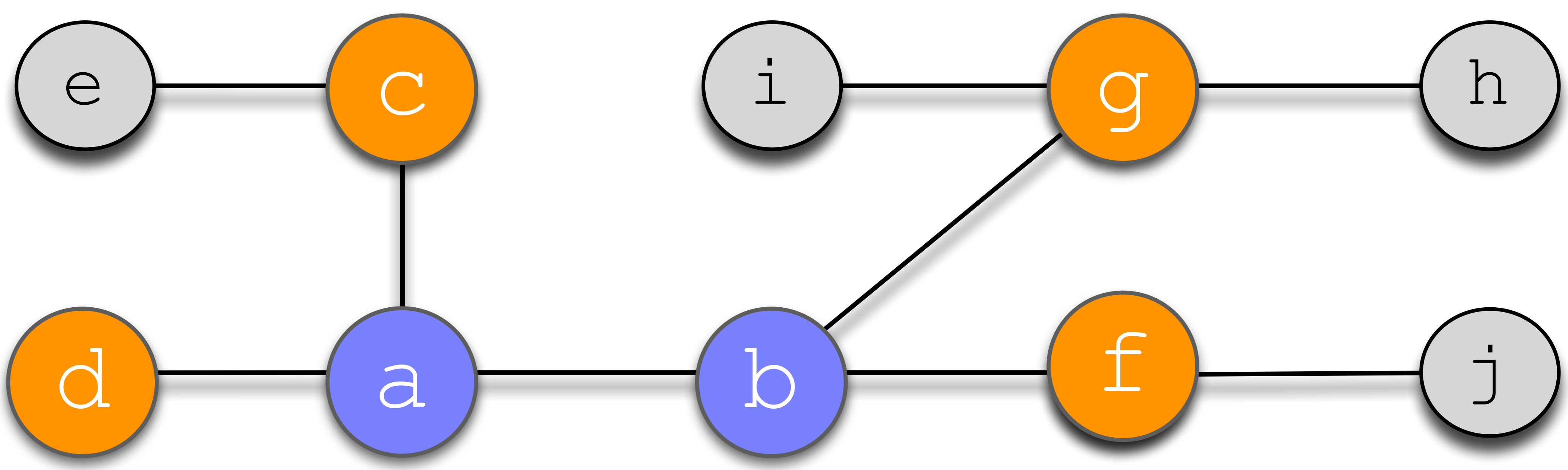


D. Brueni and L. Heath. The PMU Placement Problem. SIAM Journal on Discrete Math, 2005

Grid Measurement + Monitoring

where to place PMUs to maximize observability?

- observability rule 1: if a PMU is placed at v then v and its neighbors are observed
- observability rule 2: if a node is observable and all neighbors except one are observed, then all neighbors are observable



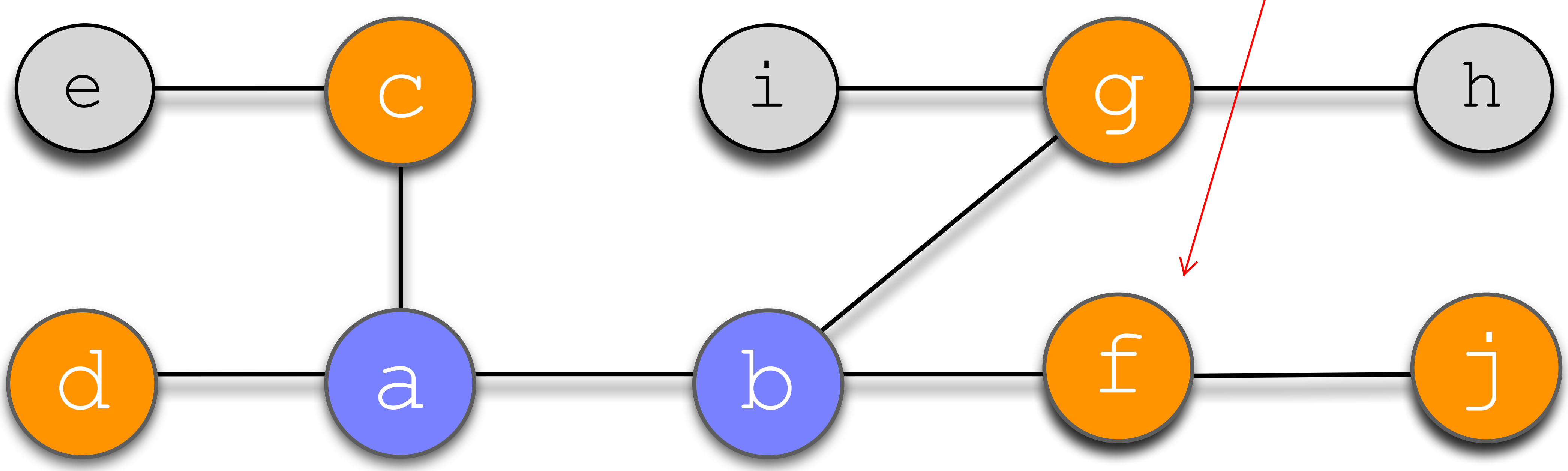
D. Brueni and L. Heath. The PMU Placement Problem. SIAM Journal on Discrete Math, 2005

Grid Measurement + Monitoring

where to place PMUs to maximize observability?

- <u>observability rule 1</u>: if a PMU is placed at v then v and its neighbors are observed
- <u>observability rule 2</u>: if a node is observed and all neighbors except one are observed, then all neighbors are observable

you should point to these as you're going along. (i.e., physicall ypoint). You're a bit "flat" in your talk here - not moving, not pointing. I am guessing you are tired.

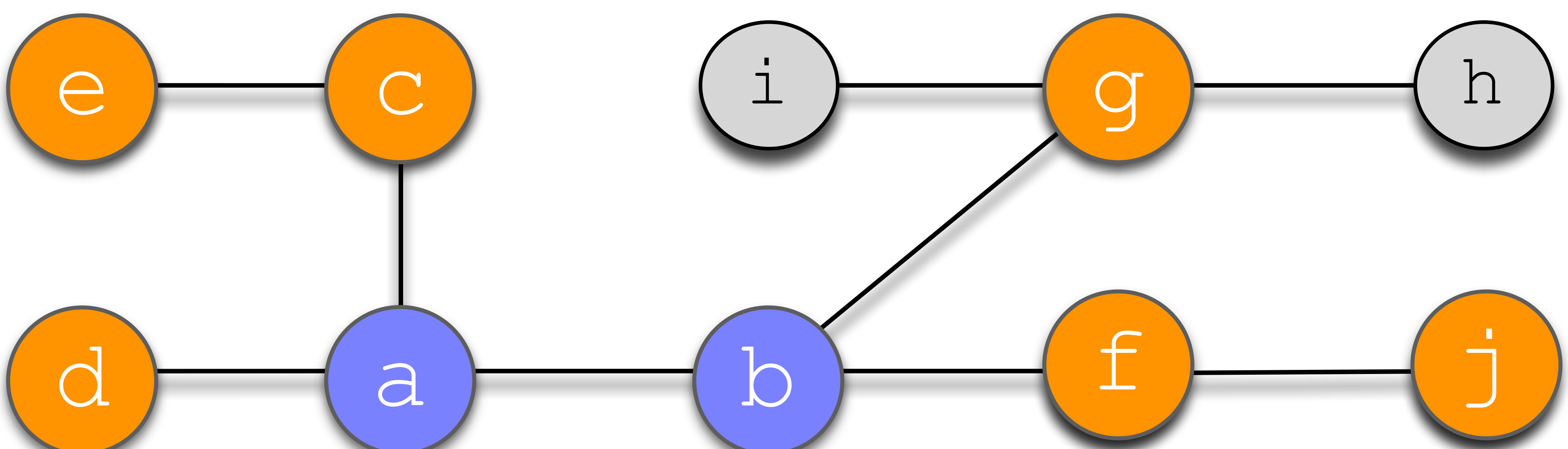


D. Brueni and L. Heath. The PMU Placement Problem. SIAM Journal on Discrete Math, 2005

Grid Measurement + Monitoring

where to place PMUs to maximize observability?

- <u>observability rule 1</u>: if a PMU is placed at v then v and its neighbors are observed
- <u>observability rule 2</u>: if a node is observable and all neighbors except one are observed, then all neighbors are observable



D. Brueni and L. Heath. The PMU Placement Problem. SIAM Journal on Discrete Math, 2005

MAX-OBSERVE Problem

- Input: `G=(V,E)` and `k` PMUs
- Output: placement of `k` PMUs to maximize number of observed nodes

MAX-OBSERVE Problem

- Input: `G=(V,E)` and `k` PMUs
- Output: placement of `k` PMUs to maximize number of observed nodes

MAX-OBSERVE is NP-Complete, reduce from planar 3SAT

PMU Placement w/ Cross-Validation

where to place PMUs to maximize observability with requirement that all PMUs are cross-validated?

Vanfretti et al. A Phasor-data-based State Estimator Incorporating Phase Bias Correction. IEEE Transactions on Power Systems, 2010

PMU Placement w/ Cross-Validation

where to place PMUs to maximize observability with requirement that all PMUs are cross-validated?

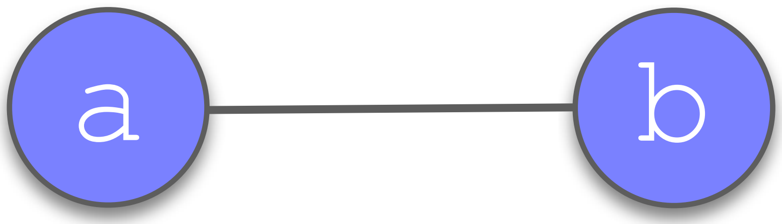
- cross-validation provides measurement error detection

Vanfretti et al. A Phasor-data-based State Estimator Incorporating Phase Bias Correction. IEEE Transactions on Power Systems, 2010

PMU Placement w/ Cross-Validation

where to place PMUs to maximize observability with requirement that all PMUs are cross-validated?

- cross-validation provides measurement error detection
- cross-validation rule 1: if PMUs are placed at adjacent nodes, they cross-validate each other

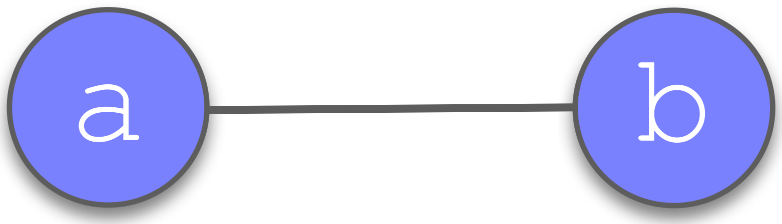


Vanfretti et al. A Phasor-data-based State Estimator Incorporating Phase Bias Correction. IEEE Transactions on Power Systems, 2010

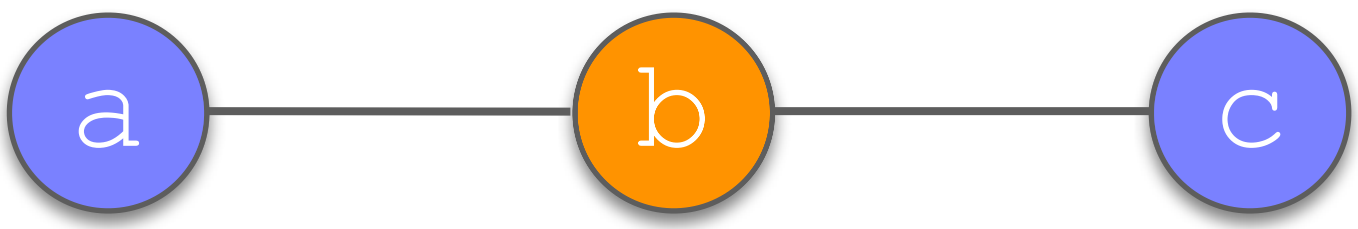
PMU Placement w/ Cross-Validation

where to place PMUs to maximize observability with requirement that all PMUs are cross-validated?

- cross-validation provides measurement error detection
- cross-validation rule 1: if PMUs are placed at adjacent nodes, they cross-validate each other



- cross-validation rule 2: if two PMUs share a common neighbor, they cross-validate each other



Vanfretti et al. A Phasor-data-based State Estimator Incorporating Phase Bias Correction. IEEE Transactions on Power Systems, 2010

MAX-OBSERVE-XV Problem

- Input: `G=(V,E)` and `k` PMUs
- Output: placement of `k` PMUs to maximize number of observed nodes, requiring that all PMUs are cross-validated

MAX-OBSERVE-XV Problem

- Input: `G=(V,E)` and `k` PMUs
- Output: placement of `k` PMUs to maximize number of observed nodes, requiring that all PMUs are cross-validated

MAX-OBSERVE-XV is NP-Complete, reduce from planar 3SAT

Greedy Approximations

Greedy Approximations

GREEDY:

- iteratively add PMU to node that results in the observation of max # of nodes

Greedy Approximations

GREEDY:

- iteratively add PMU to node that results in the observation of max # of nodes

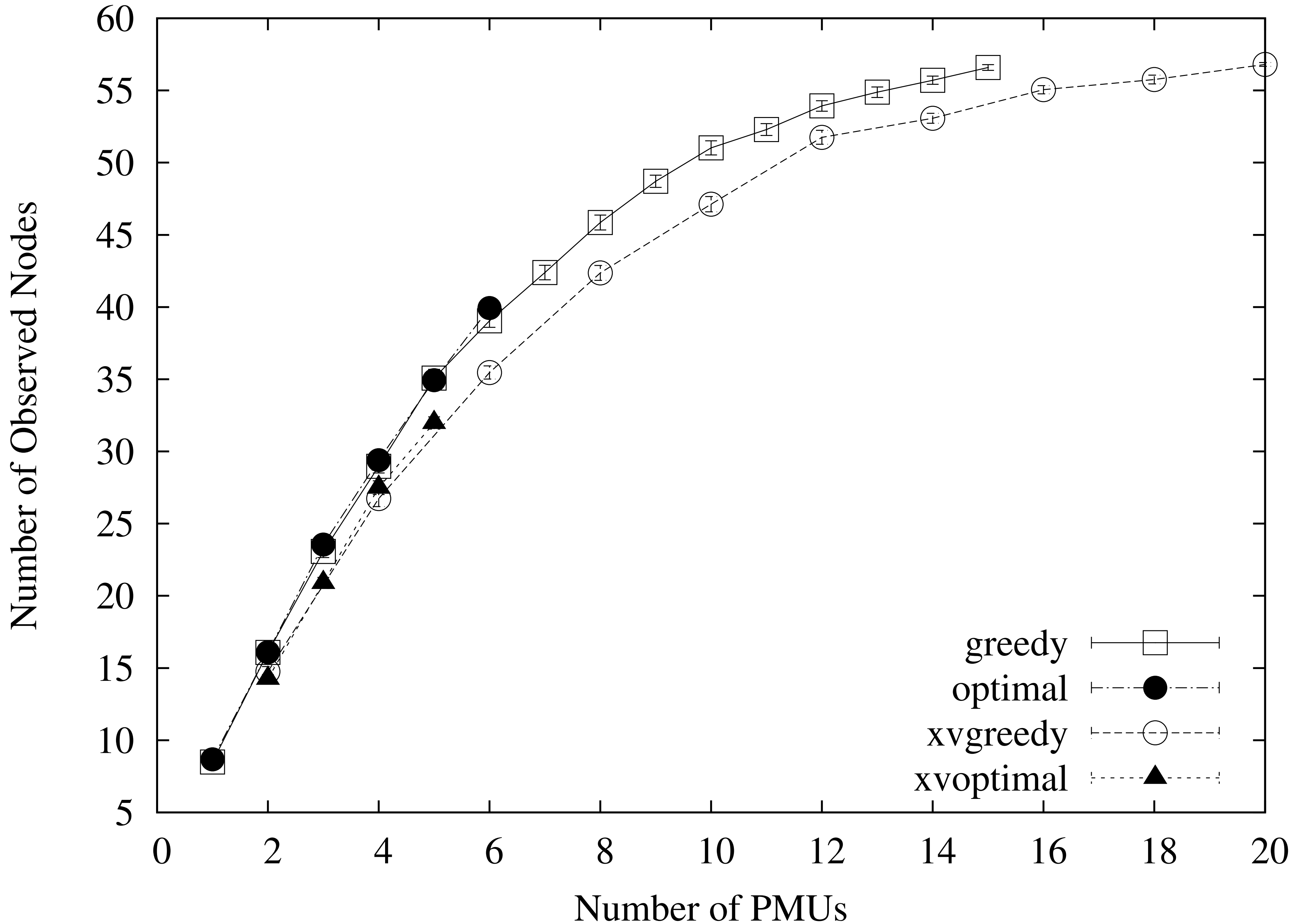
GREEDY-XV:

- iteratively place PMU pairs at nodes `{u,v}` such that `u` and `v` are cross-validated and results in the observation of max # of nodes

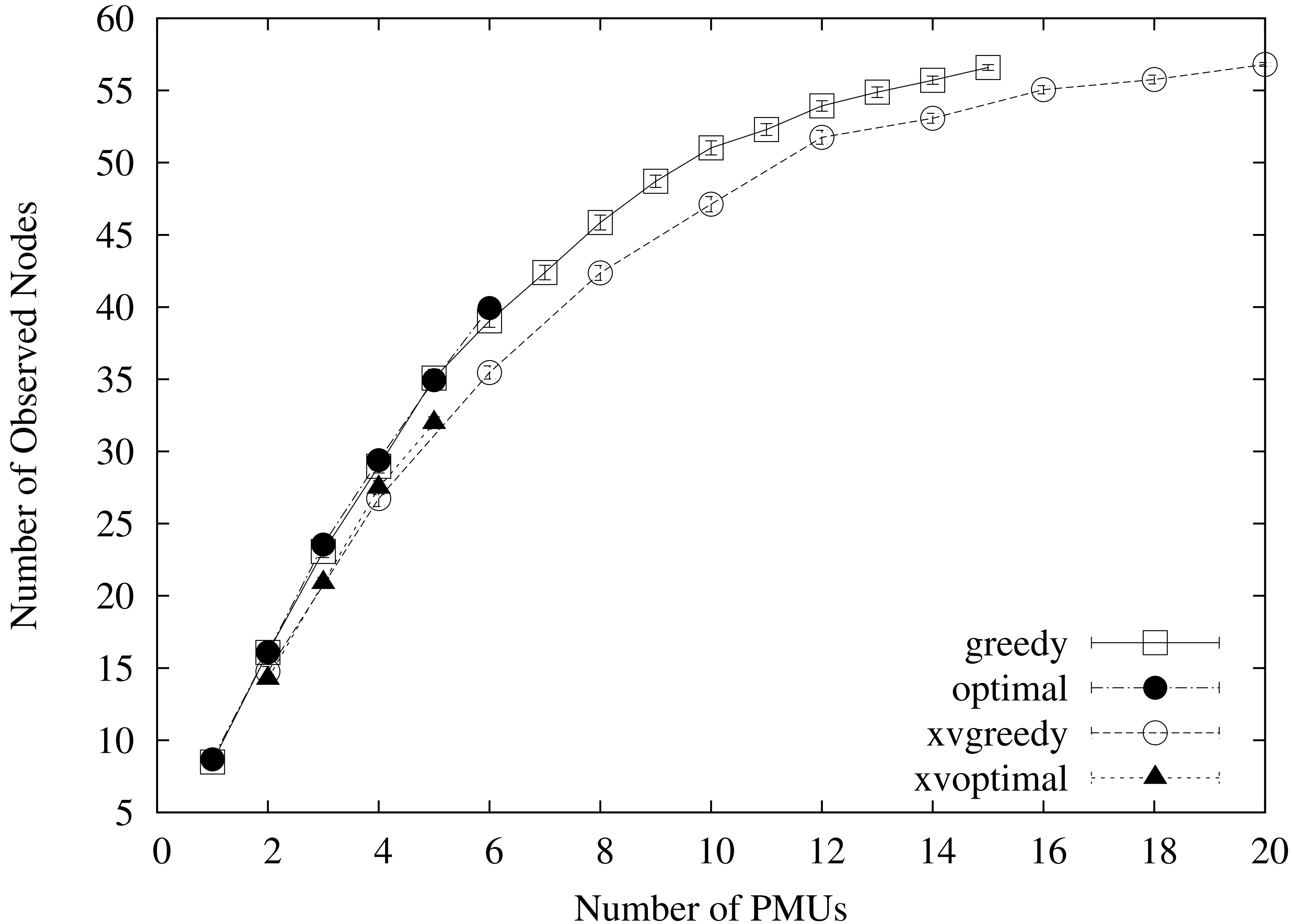
Simulation Setup

- generate grid networks with same degree distribution as IEEE bus systems
  - show results for synthetic topologies based on IEEE Bus 57
- compare with brute-force optimal solution by enumeration with small # of PMUs

Simulation Results

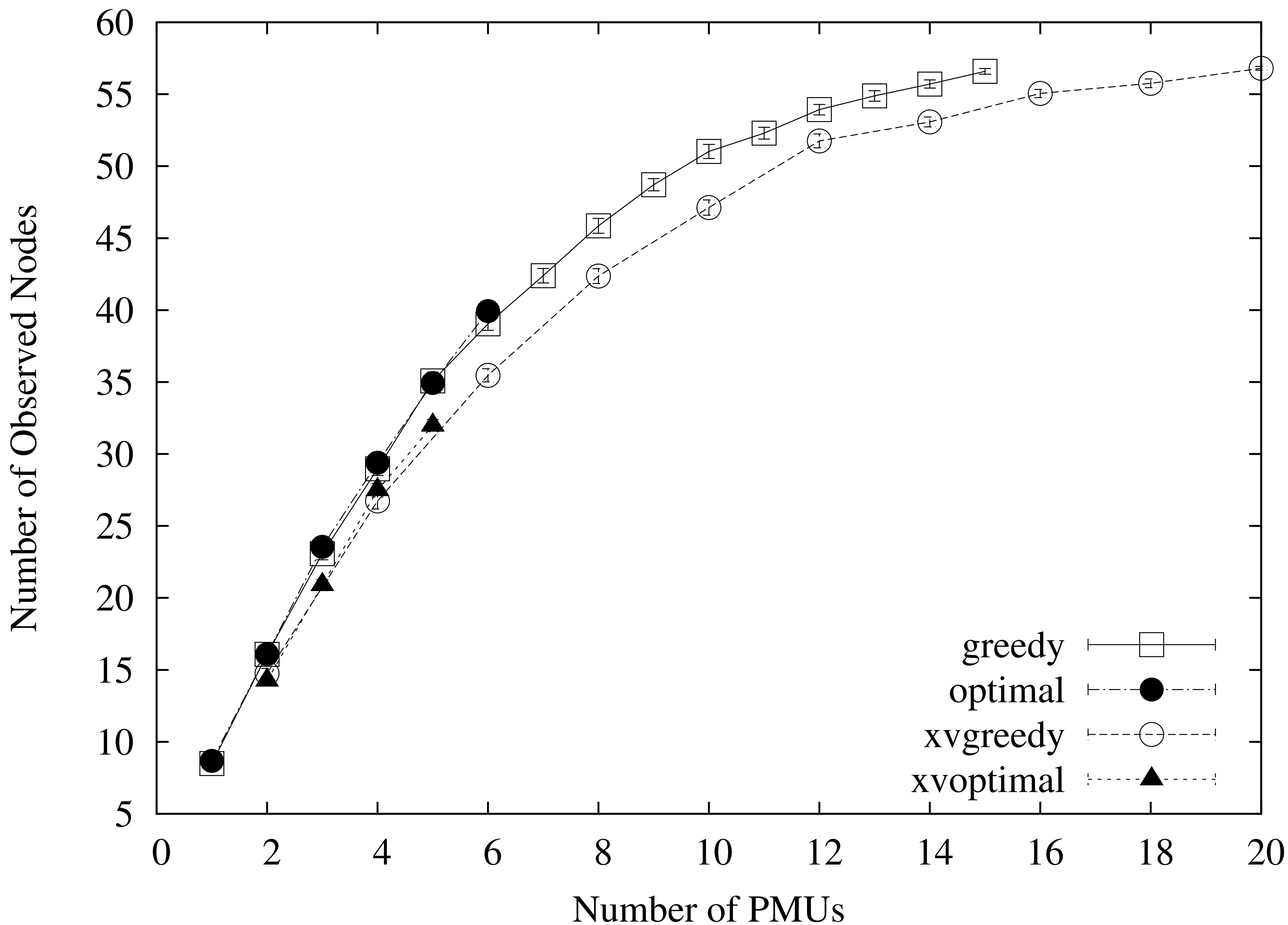


Simulation Results



result 1: greedy solutions within 97% of optimal

Simulation Results



<u>result 1</u>: greedy solutions within 97% of optimal

<u>result 2</u>: cross-validation decreases # observed by ~ 5%

Talk Outline

- thesis introduction
- placement of smart grid sensors to enable measurement error detection
- recovery from failed communication links in a smart grid
- recovery from malicious nodes injecting false routing state
- outline for future work and conclusions

Power Grid Data Dissemination

Power Grid Data Dissemination

- currently use outdated SCADA system (1960s)
  - course-grained measurements
  - data only locally distributed

Power Grid Data Dissemination

- currently use outdated SCADA system (1960s)
  - course-grained measurements
  - data only locally distributed
- smart grid wide-area data dissemination
  - many sensors producing data (PMUs, IEDs,...)
  - many sinks with interests in subset of data (utility companies, balancing authorities,...)

Power Grid Data Dissemination

- currently use outdated SCADA system (1960s)
  - course-grained measurements
  - data only locally distributed
- smart grid wide-area data dissemination
  - many sensors producing data (PMUs, IEDs,...)
  - many sinks with interests in subset of data (utility companies, balancing authorities,...)

wide-area data dissemination will enable more efficient ways to operate and manage the grid

Smart Grid Data Dissemination

Smart Grid Data Dissemination

- richer communication paradigms

Smart Grid Data Dissemination

- richer communication paradigms
  - QoS bandwidth, delay, rate guarantees

Smart Grid Data Dissemination

- richer communication paradigms
  - QoS bandwidth, delay, rate guarantees
  - multicast delivery (1 to many)

Smart Grid Data Dissemination

- richer communication paradigms
  - QoS bandwidth, delay, rate guarantees
  - multicast delivery (1 to many)
    - multiple source-based multicast trees (MTs)

Smart Grid Data Dissemination

- richer communication paradigms
  - QoS bandwidth, delay, rate guarantees
  - multicast delivery (1 to many)
    - multiple source-based multicast trees (MTs)
  - self-healing multicast delivery

Smart Grid Data Dissemination

- richer communication paradigms
    - QoS bandwidth, delay, rate guarantees
    - multicast delivery (1 to many)
        - multiple source-based multicast trees (MTs)
    - self-healing multicast delivery

Smart Grid Data Dissemination

- richer communication paradigms
  - QoS bandwidth, delay, rate guarantees
  - multicast delivery (1 to many)
    - multiple source-based multicast trees (MTs)
  - self-healing multicast delivery

focus: recovery from comm. link failures in smart grid
=> precompute backup MTs offline in case link fails

Smart Grid App QoS Requirements

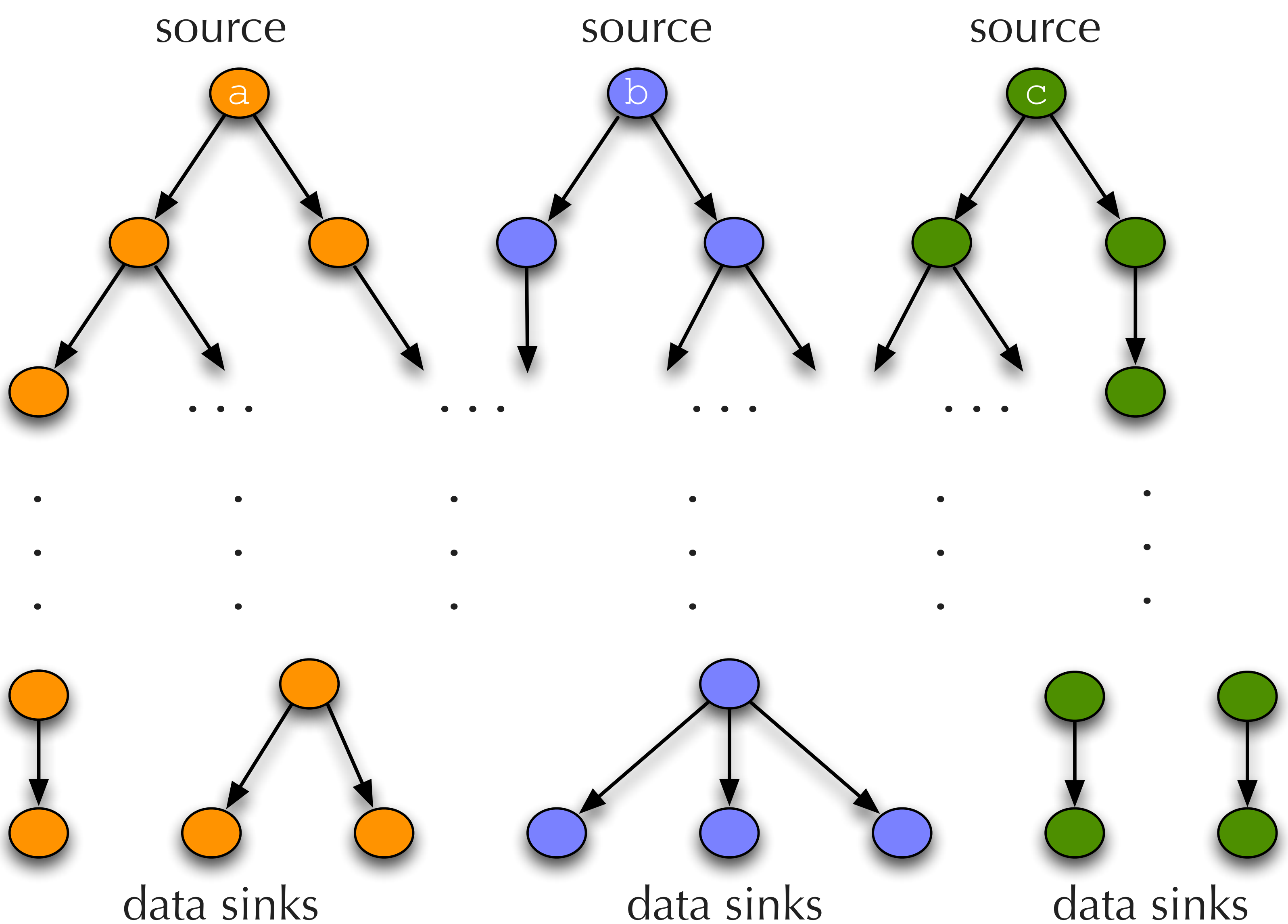
- focus on QoS requirements for critical smart applications (e.g., real-time control)
- stringent and challenge QoS
    - low latency
        - ex. 8-16 ms per packet for real-time control
    - low tolerance for packet loss

Smart Grid App QoS Requirements

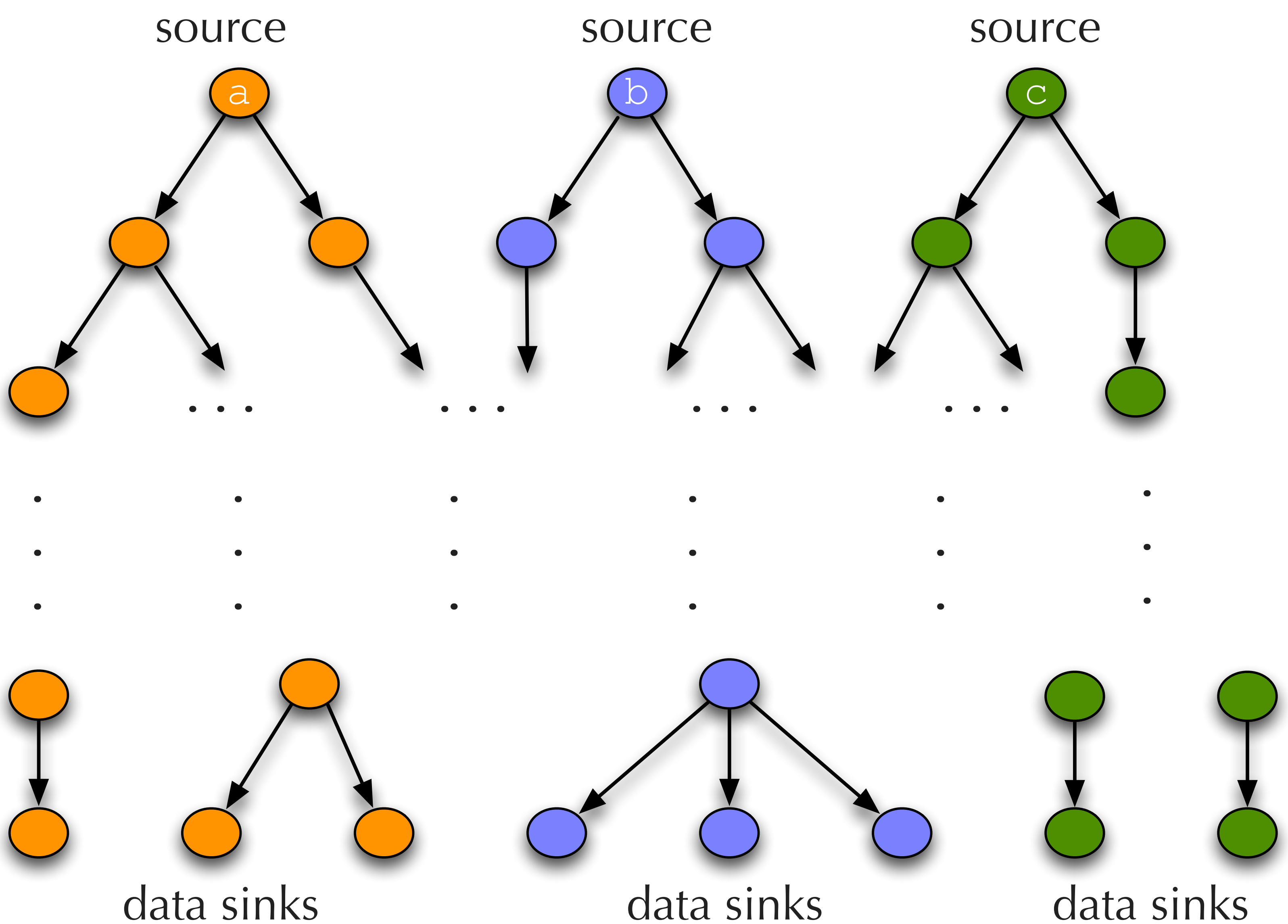
- focus on QoS requirements for critical smart applications (e.g., real-time control)
- stringent and challenge QoS
  - low latency
    - ex. 8-16 ms per packet for real-time control
  - low tolerance for packet loss

hard E2E delivery guarantees are needed!

Problem Description

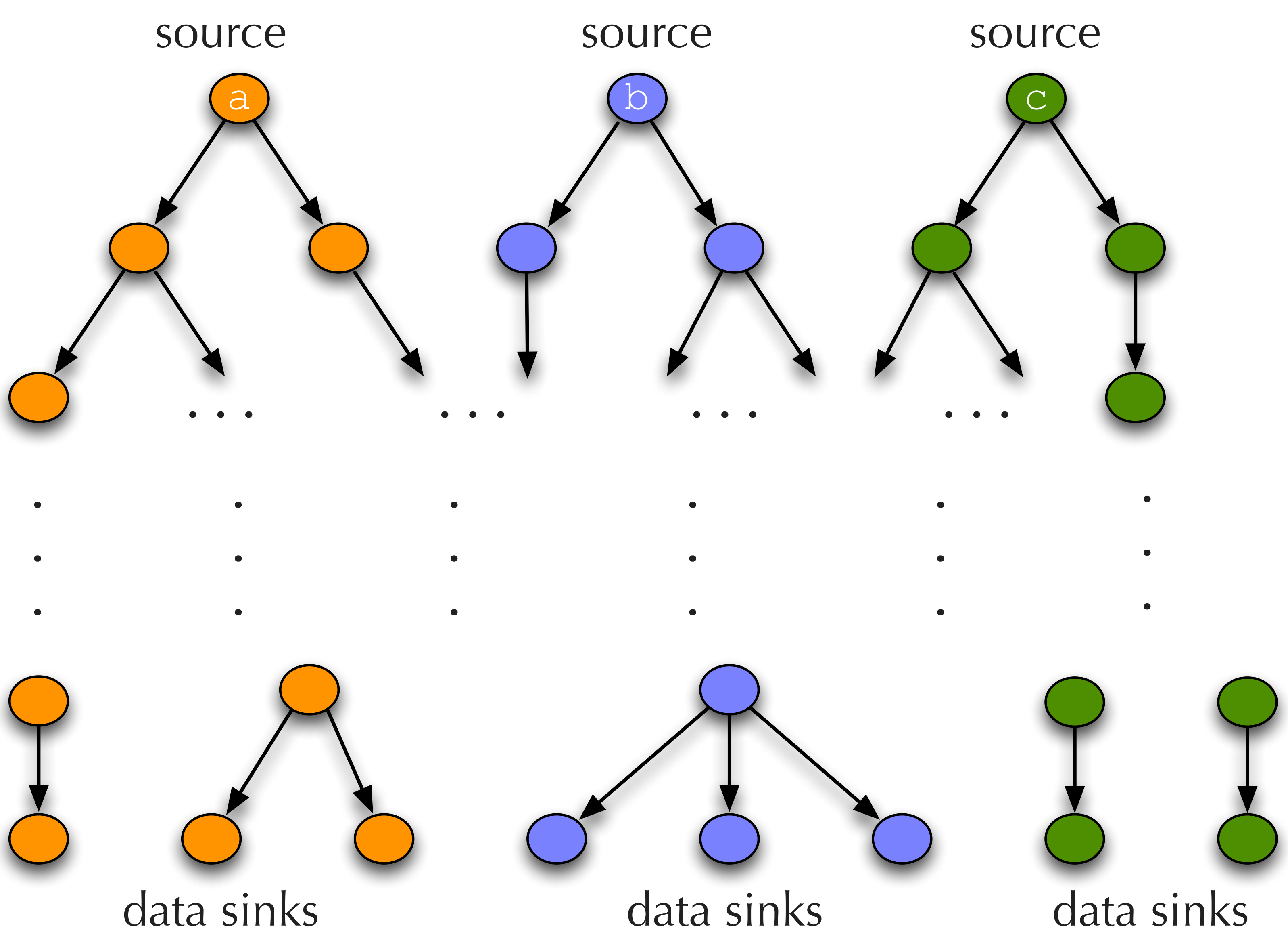


Problem Description



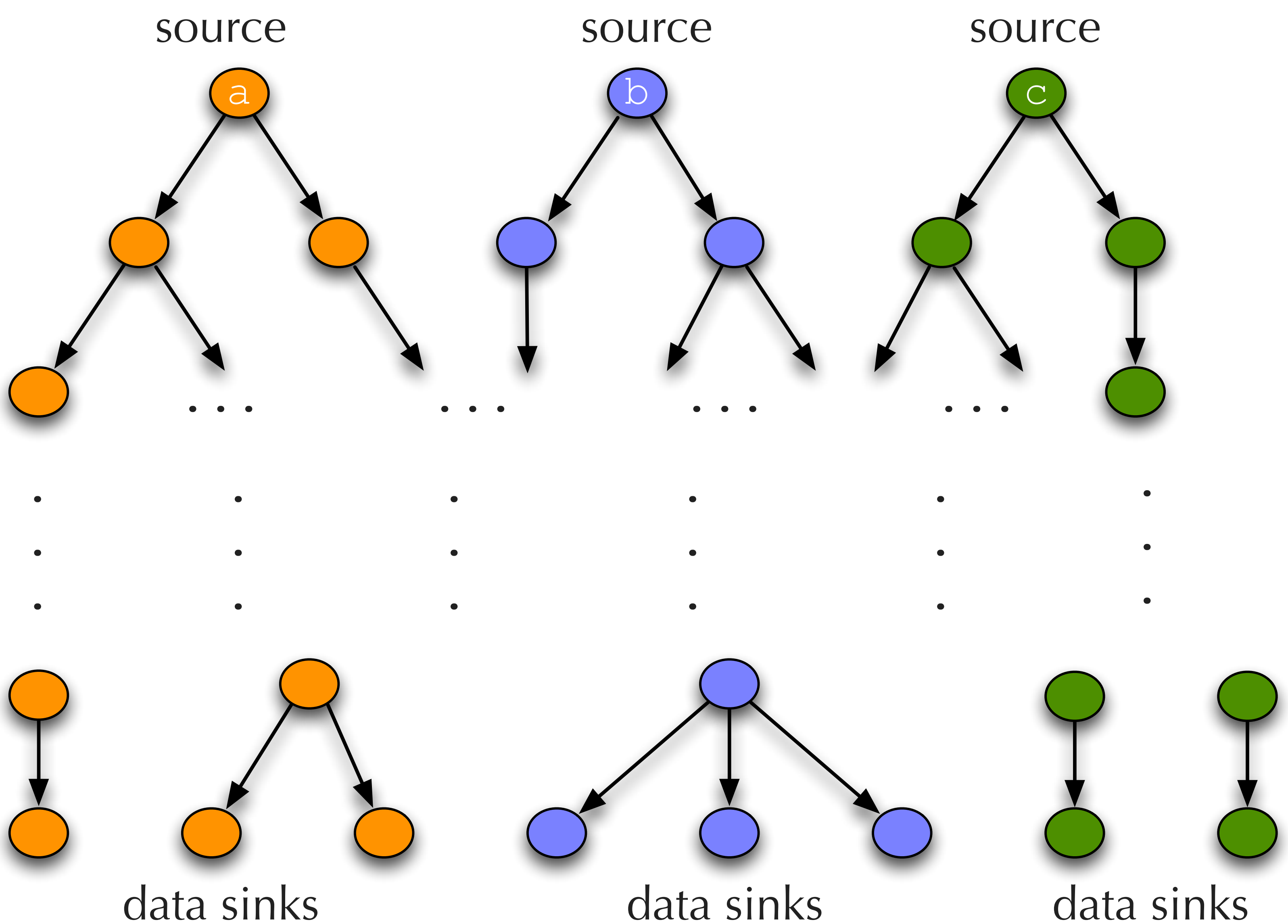
- multiple source-based MTs

Problem Description



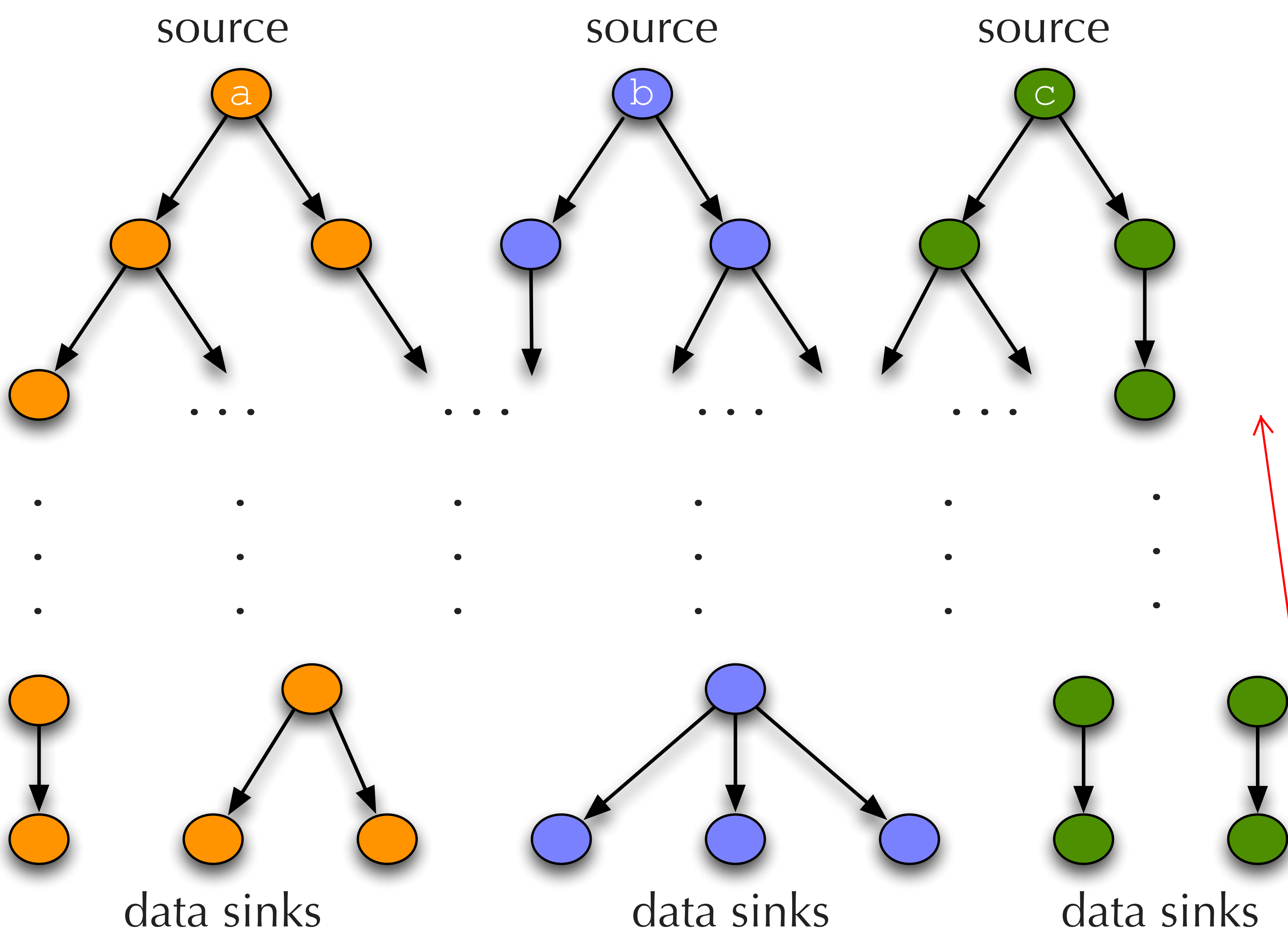
- multiple source-based MTs
  - rate based traffic from sender

Problem Description



- multiple source-based MTs
  - rate based traffic from sender
  - each (source,sink) pair has E2E per packet delay req.

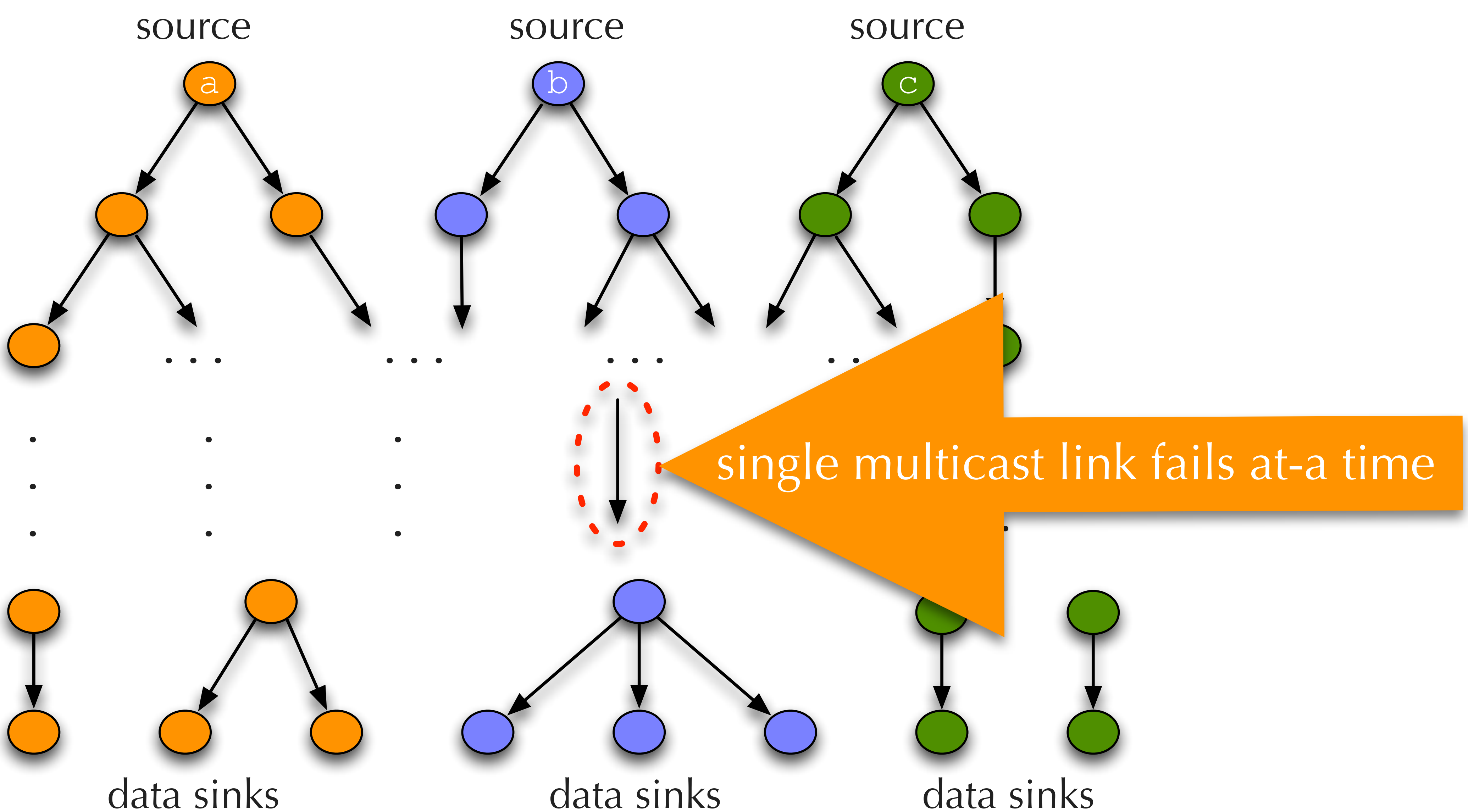
Problem Description



- multiple source-based MTs
  - rate based traffic from sender
  - each (source,sink) pair has E2E per packet delay re
- fixed sender and receiver set => predictable traffic

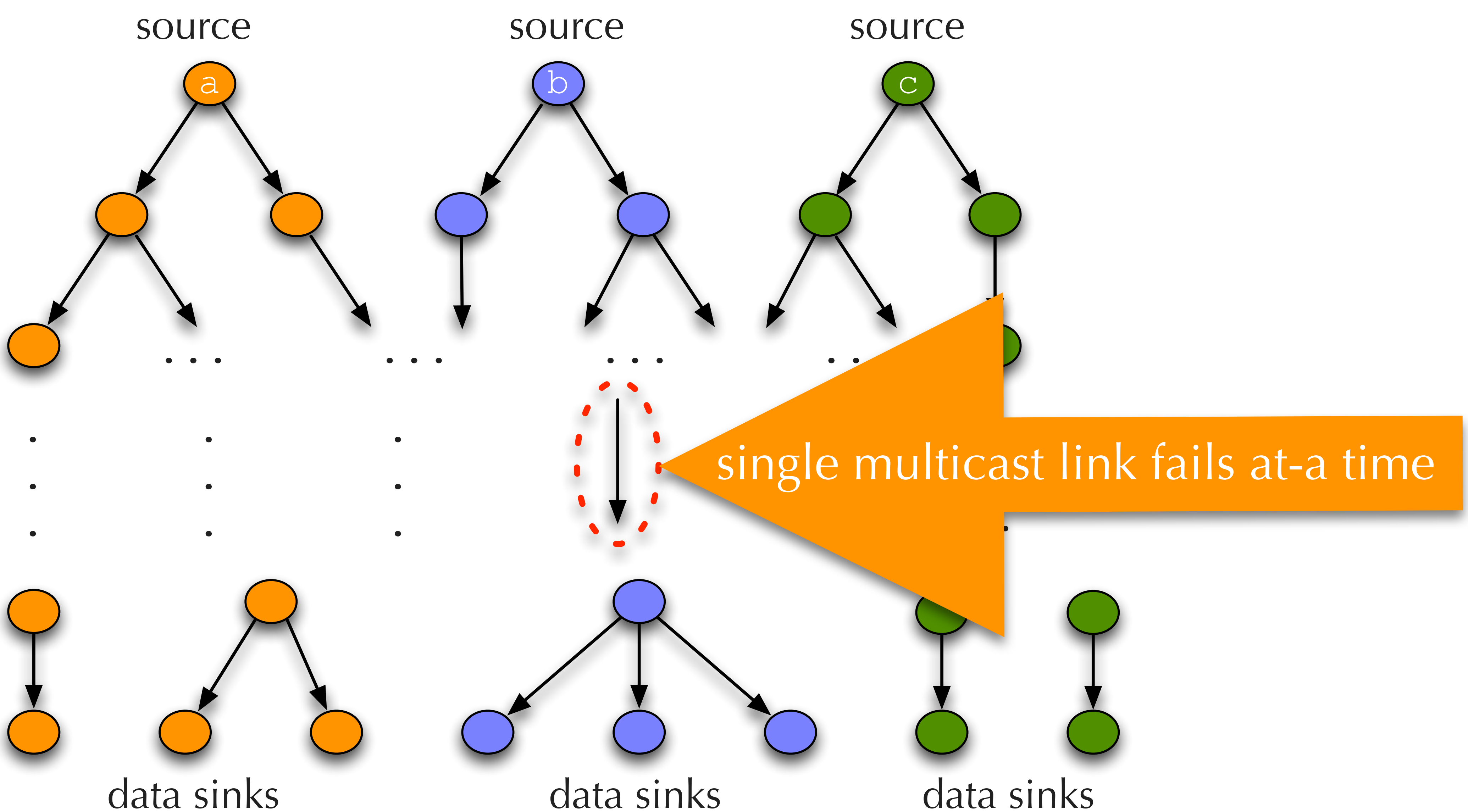
let's discuss this figure.
I think you want to overlay different colored trees on one graph (and in particular showing multiple strees crossing the same link). the three different colors each being saprate here doesn't get the idea across

Problem Description



- multiple source-based MTs
  - rate based traffic from sender
  - each (source,sink) pair has E2E per packet delay req.
- fixed sender and receiver set => predictable traffic

Problem Description



how to detect MT link failure + recover such that E2E packet loss and delay is minimized?

- fixed sender and receiver set => predictable traffic

2 Step Solution

add a hyphen

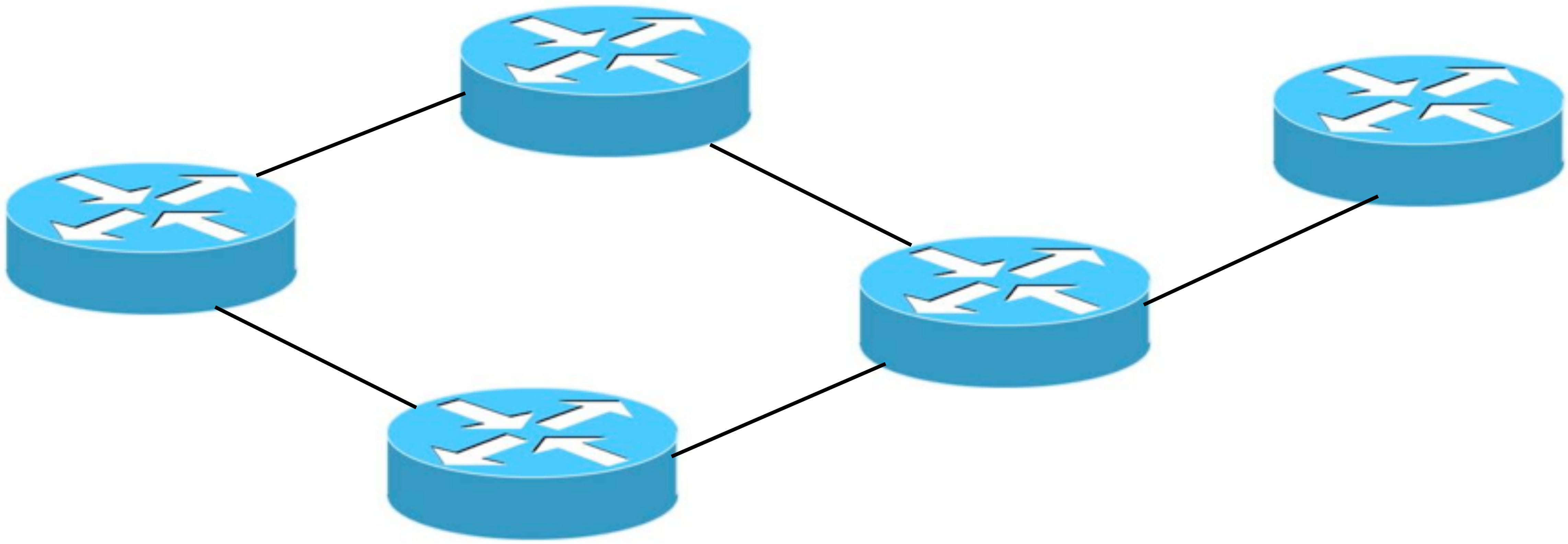
- two steps to provide link failure robustness
  1. fast detection of link failures
     - detect link failure inside network
  2. install precomputed backup multicast trees

2 Step Solution

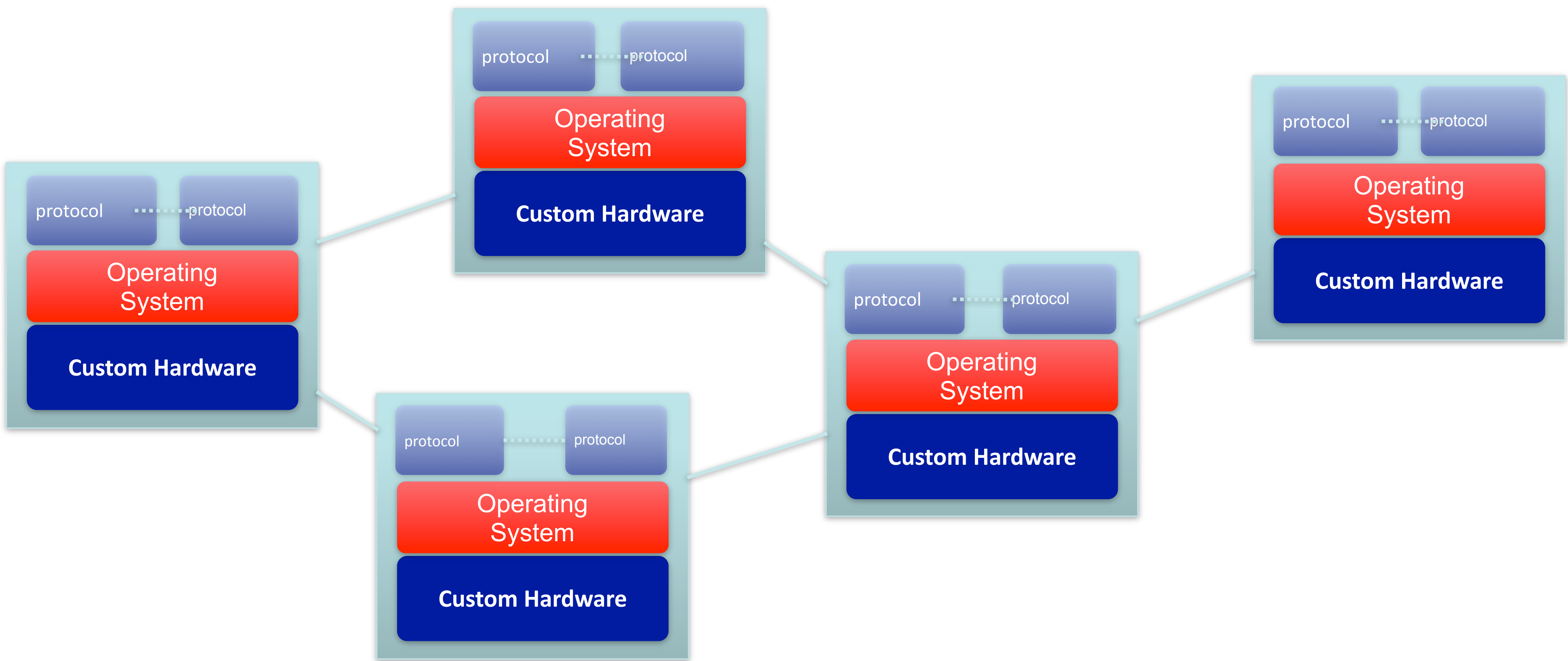
- two steps to provide link failure robustness
  1. fast detection of link failures
     - detect link failure inside network
  2. install precomputed backup multicast trees

requires changes to existing routers ... how?

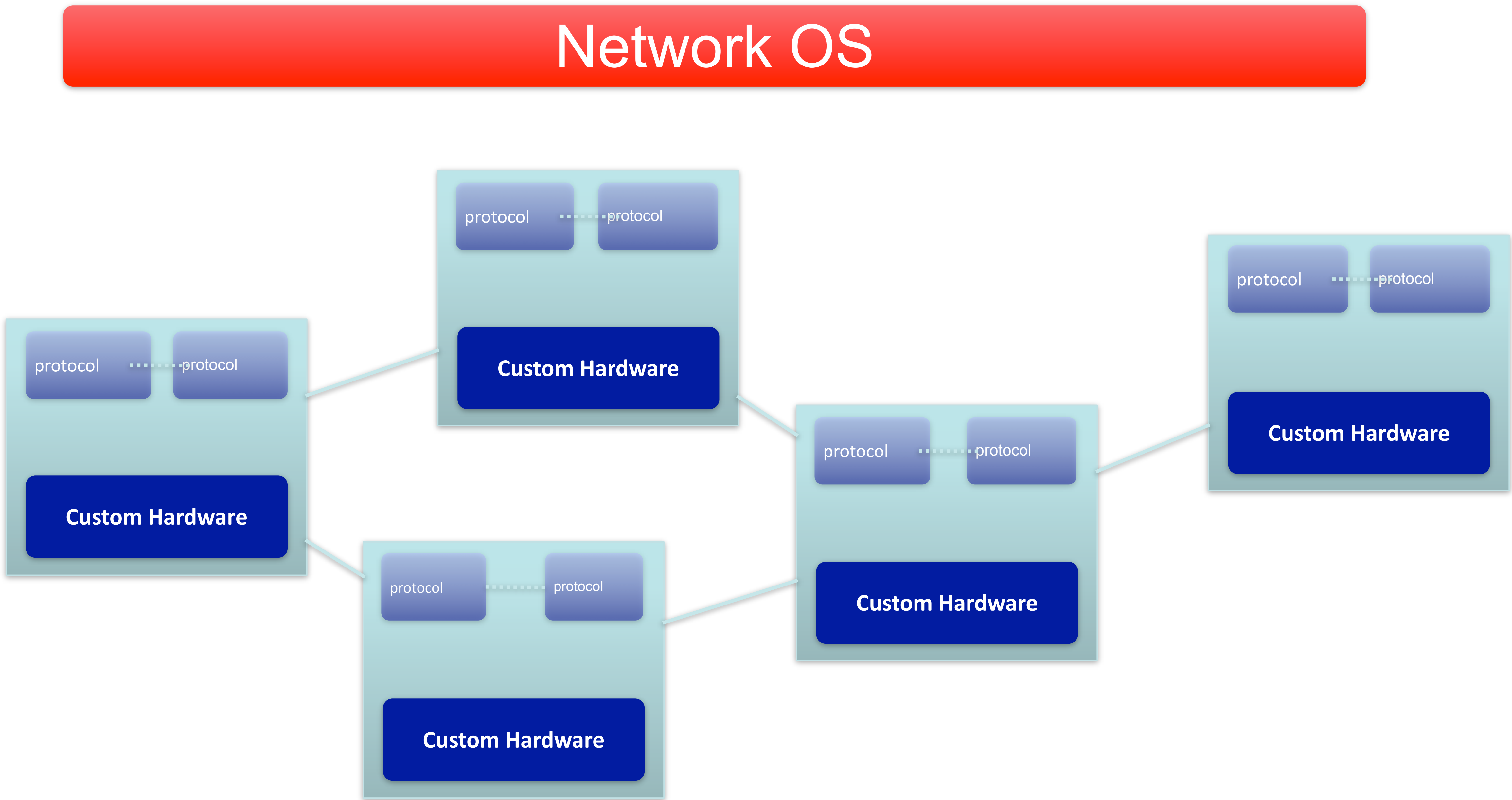
OpenFlow: Open Network Control Plane



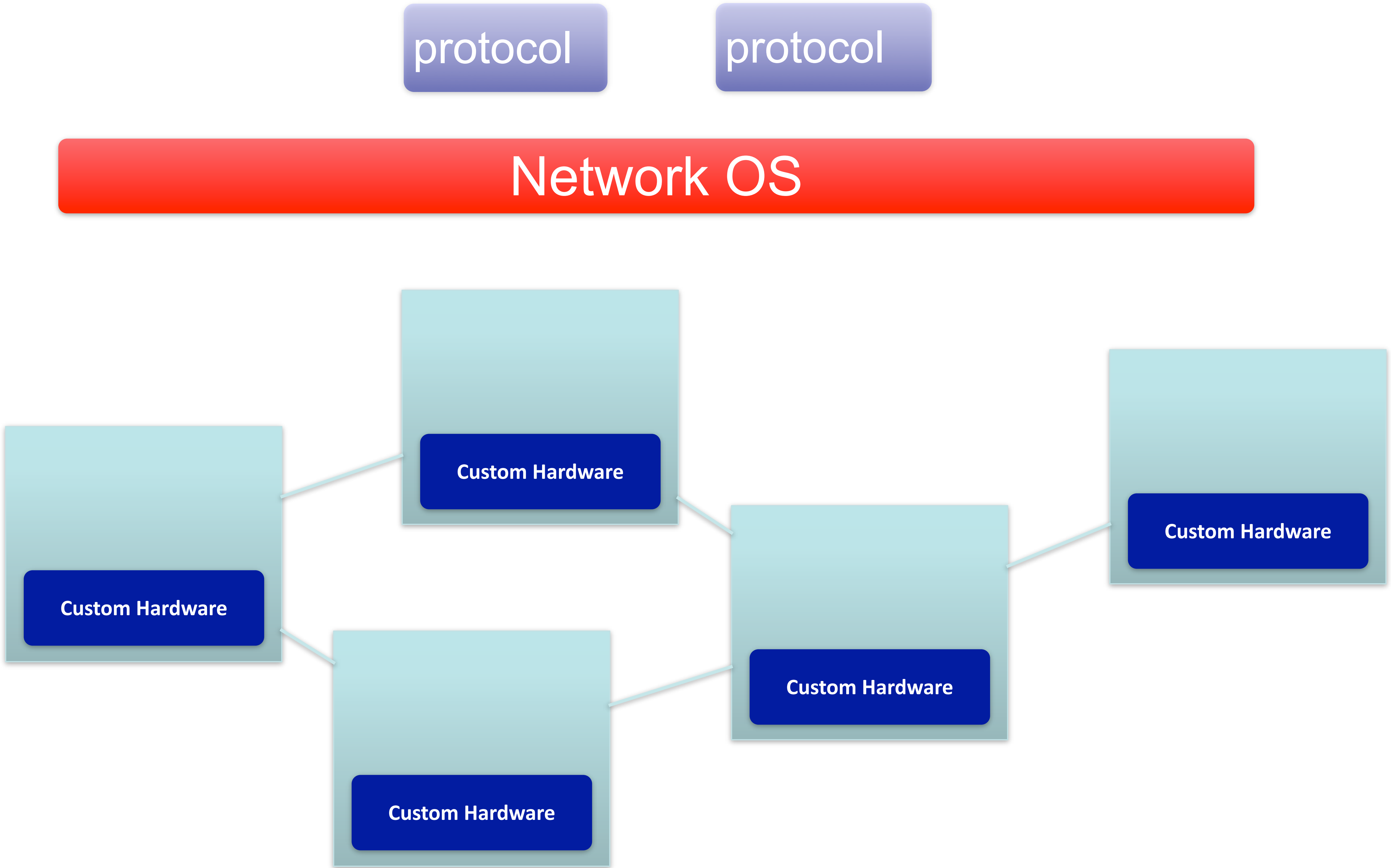
OpenFlow: Open Network Control Plane



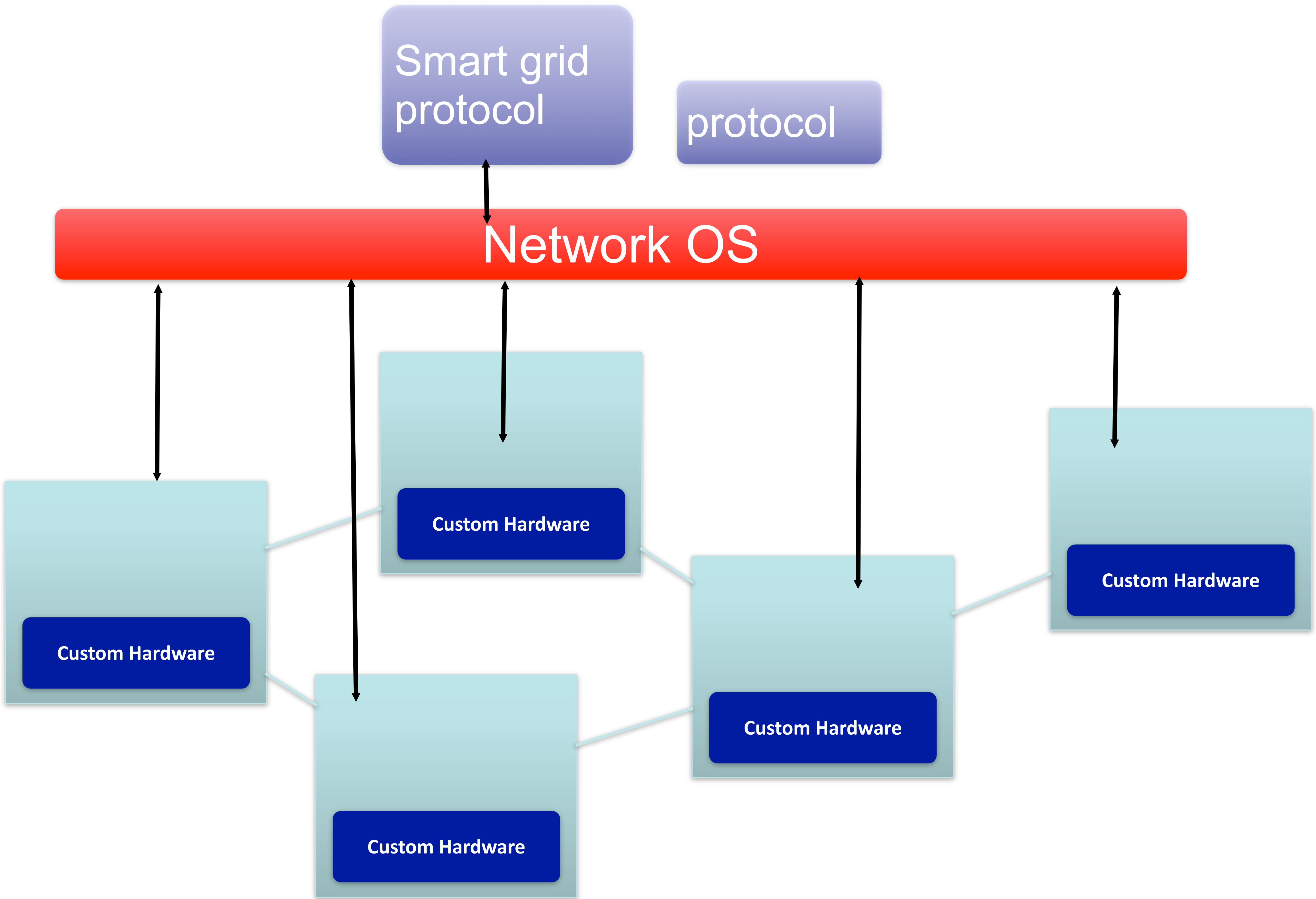
OpenFlow: Open Network Control Plane



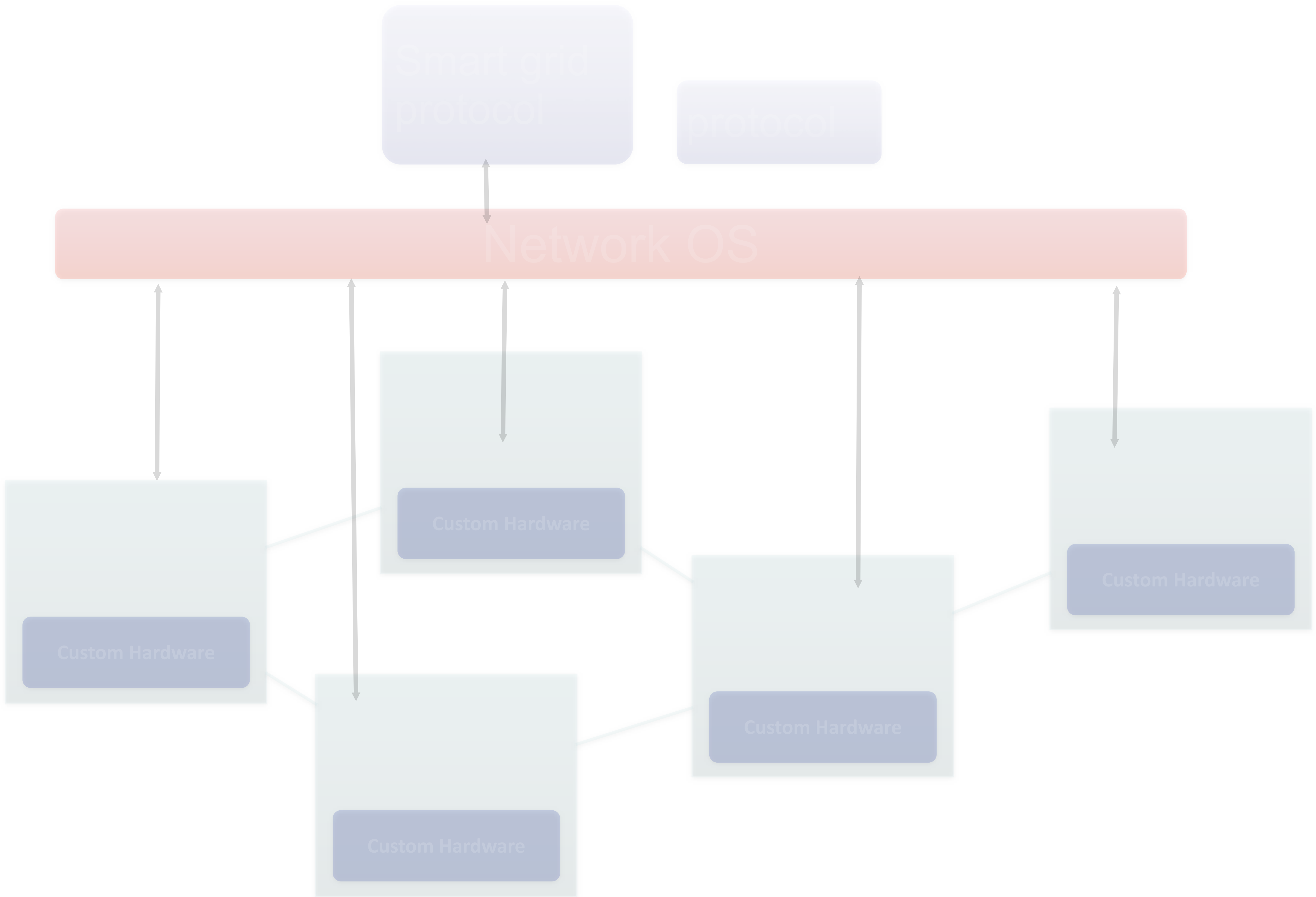
OpenFlow: Open Network Control Plane



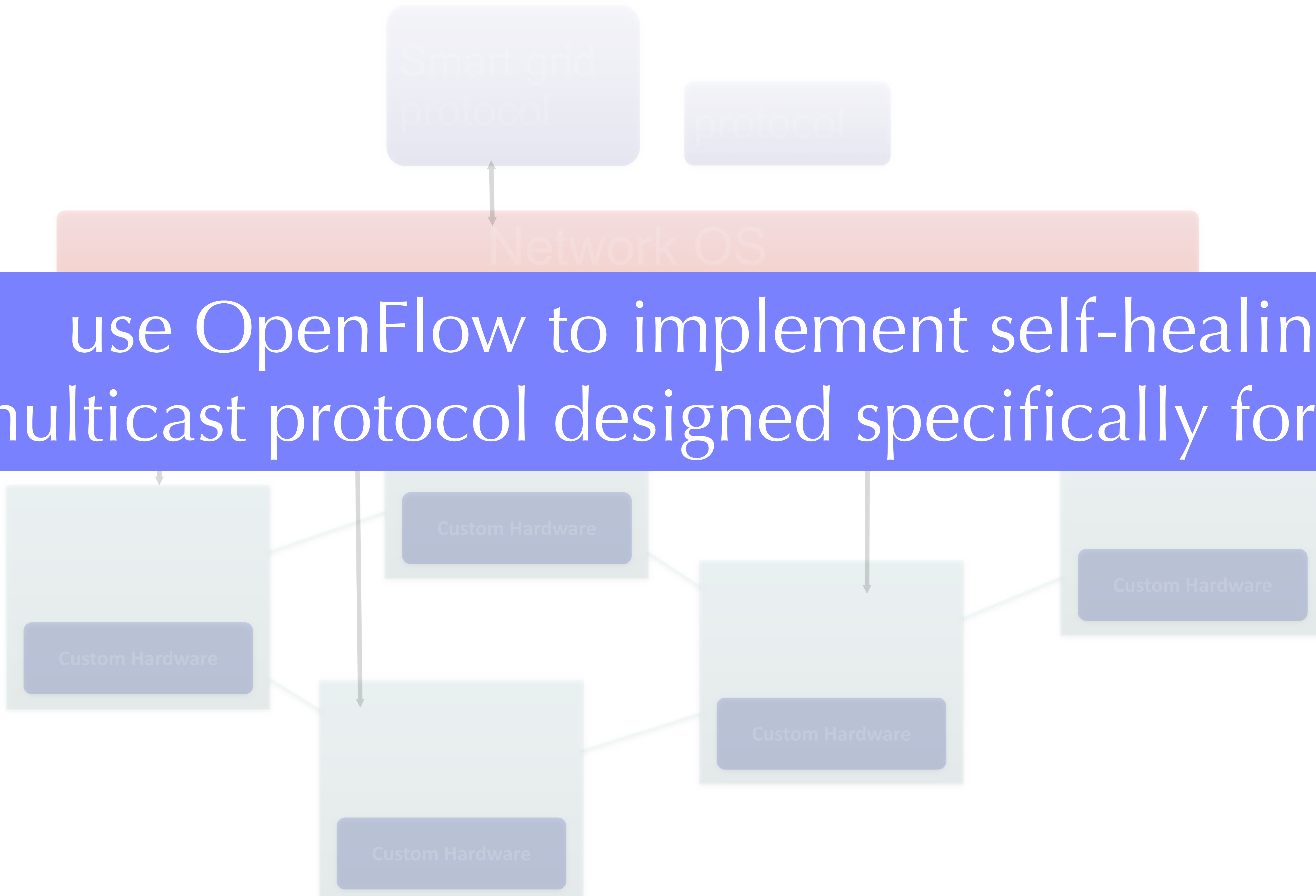
OpenFlow: Open Smart Grid Control Plane



OpenFlow: Open Smart Grid Control Plane



OpenFlow: Open Smart Grid Control Plane



use OpenFlow to implement self-healing multicast protocol designed specifically for grid

Why Develop a New Solution?

Why Develop a New Solution?

- public Internet
  - best-effort model unsuitable to meeting hard E2E delay requirements
  - not optimized for power grid conditions
    - small scale, traffic is known

Why Develop a New Solution?

- public Internet
  - best-effort model unsuitable to meeting hard E2E delay requirements
  - not optimized for power grid conditions
    - small scale, traffic is known
- MPLS fast reroute: similar to our approach but
  - optimization criteria specified over single MT, we consider criteria across multiple MTs

Why Develop a New Solution?

- public Internet
  - best-effort model unsuitable to meeting hard E2E delay requirements
  - not optimized for power grid conditions
    - small scale, traffic is known
- MPLS fast reroute: similar to our approach but
  - optimization criteria specified over single MT, we consider criteria across multiple MTs
- Gridstat: does not address link failures

Link Failure Detection

are packets being lost at a given network link?

Link Failure Detection

are packets being lost at a given network link?

- leverage OpenFlow native packet counters + ability to tag packets

1. count and mark packets at upstream switch
2. count the marked packets at the downstream switch

Link Failure Detection

are packets being lost at a given network link?

- leverage OpenFlow native packet counters + ability to tag packets

1. count and mark packets at upstream switch
2. count the marked packets at the downstream switch

- algorithm ensures that upstream and downstream switches consider the same set of packets

MIN-FLOWS Backup Multicast Trees

MIN-FLOWS Backup Multicast Trees

- INPUT

MIN-FLOWS Backup Multicast Trees

- INPUT
  - undirected graph, set of multicast trees, set of all multicast flows, a link (`l`)

MIN-FLOWS Backup Multicast Trees

- INPUT
  - undirected graph, set of multicast trees, set of all multicast flows, a link (`l`)
- OUTPUT

MIN-FLOWS Backup Multicast Trees

- INPUT
  - undirected graph, set of multicast trees, set of all multicast flows, a link (`l`)
- OUTPUT
  - a backup MT for each MT using `l`

MIN-FLOWS Backup Multicast Trees

- INPUT
  - undirected graph, set of multicast trees, set of all multicast flows, a link (`l`)
- OUTPUT
  - a backup MT for each MT using `l`
    - call this set of MT T_l

MIN-FLOWS Backup Multicast Trees

- INPUT
  - undirected graph, set of multicast trees, set of all multicast flows, a link (`l`)
- OUTPUT
  - a backup MT for each MT using `l`
    - call this set of MT T_l
  - T_l computed s.t. *across all network links*, the same # of flows traverse each link

MIN-FLOWS Backup Multicast Trees

- INPUT
  - undirected graph, set of multicast trees, set of all multicast flows, a link (`l`)
- OUTPUT
  - a backup MT for each MT using `l`
    - call this set of MT T_l
  - T_l computed s.t. *across all network links*, the same # of flows traverse each link
  - all under the condition that E2E per-packet delay requirements are met using T_l

MIN-FLOWS Backup Multicast Trees

- INPUT
  - undirected graph, set of multicast trees, set of all multicast flows, a link (`l`)
- OUTPUT
  - a backup MT for each MT using `l`
    - call this set of MT T_l
  - T_l computed s.t. *across all network links*, the

INTUITION: min # of network flows impacted by future link failures (occurring after link `l` fails)

delay requirements are met using T_l

MIN-SINKS Backup Multicast Trees

MIN-SINKS Backup Multicast Trees

- INPUT

MIN-SINKS Backup Multicast Trees

- INPUT
  - undirected graph, set of multicast trees, set of all multicast flows, a link (`l`)

MIN-SINKS Backup Multicast Trees

- INPUT
  - undirected graph, set of multicast trees, set of all multicast flows, a link (`l`)
- OUTPUT

MIN-SINKS Backup Multicast Trees

- INPUT
  - undirected graph, set of multicast trees, set of all multicast flows, a link (`l`)
- OUTPUT
  - a backup MT for each MT using `l`

MIN-SINKS Backup Multicast Trees

- INPUT
  - undirected graph, set of multicast trees, set of all multicast flows, a link (`l`)
- OUTPUT
  - a backup MT for each MT using `l`
    - call this set of MT $\mathtt{T}_{\mathtt{l}}$

MIN-SINKS Backup Multicast Trees

- INPUT
  - undirected graph, set of multicast trees, set of all multicast flows, a link (`l`)
- OUTPUT
  - a backup MT for each MT using `l`
    - call this set of MT T_l
  - T_l computed s.t. *across all network links*, each link has the # of downstream sink nodes

MIN-SINKS Backup Multicast Trees

- INPUT
  - undirected graph, set of multicast trees, set of all multicast flows, a link (`l`)
- OUTPUT
  - a backup MT for each MT using `l`
    - call this set of MT T_l
  - T_l computed s.t. *across all network links*, each link has the # of downstream sink nodes
  - all under the condition that E2E per-packet delay requirements are met using T_l

MIN-SINKS Backup Multicast Trees

- INPUT
  - undirected graph, set of multicast trees, set of all multicast flows, a link (`l`)
- OUTPUT
  - a backup MT for each MT using `l`
    - call this set of MT T_l
  - T_l computed s.t. *across all network links*, each

INTUITION: min # of data sinks impacted by future link failures (occurring after link `l` fails)

delay requirements are met using T_l

MIN-CONTROL Backup MTs

MIN-CONTROL Backup MTs

- INPUT

MIN-CONTROL Backup MTs

- INPUT
  - undirected graph, set of multicast trees, set of all multicast flows, a link (`l`)

MIN-CONTROL Backup MTs

- INPUT
  - undirected graph, set of multicast trees, set of all multicast flows, a link (`l`)
- OUTPUT

MIN-CONTROL Backup MTs

- INPUT
  - undirected graph, set of multicast trees, set of all multicast flows, a link (`l`)
- OUTPUT
  - a backup MT for each MT using `l`

MIN-CONTROL Backup MTs

- INPUT
  - undirected graph, set of multicast trees, set of all multicast flows, a link (`l`)
- OUTPUT
  - a backup MT for each MT using `l`
    - call this set of MT T_l

MIN-CONTROL Backup MTs

- INPUT
  - undirected graph, set of multicast trees, set of all multicast flows, a link (`l`)
- OUTPUT
  - a backup MT for each MT using `l`
    - call this set of MT T_l
  - T_l computed s.t. the minimum # of routers need to be signaled to change state

MIN-CONTROL Backup MTs

- INPUT
  - undirected graph, set of multicast trees, set of all multicast flows, a link (`l`)
- OUTPUT
  - a backup MT for each MT using `l`
    - call this set of MT T_l
  - T_l computed s.t. the minimum # of routers need to be signaled to change state
  - all under the condition that E2E per-packet delay requirements are met using T_l

MIN-CONTROL Backup MTs

- INPUT
  - undirected graph, set of multicast trees, set of all multicast flows, a link (`l`)
- OUTPUT
  - a backup MT for each MT using `l`
    - call this set of MT T_l
  - T_l computed s.t. the minimum # of routers

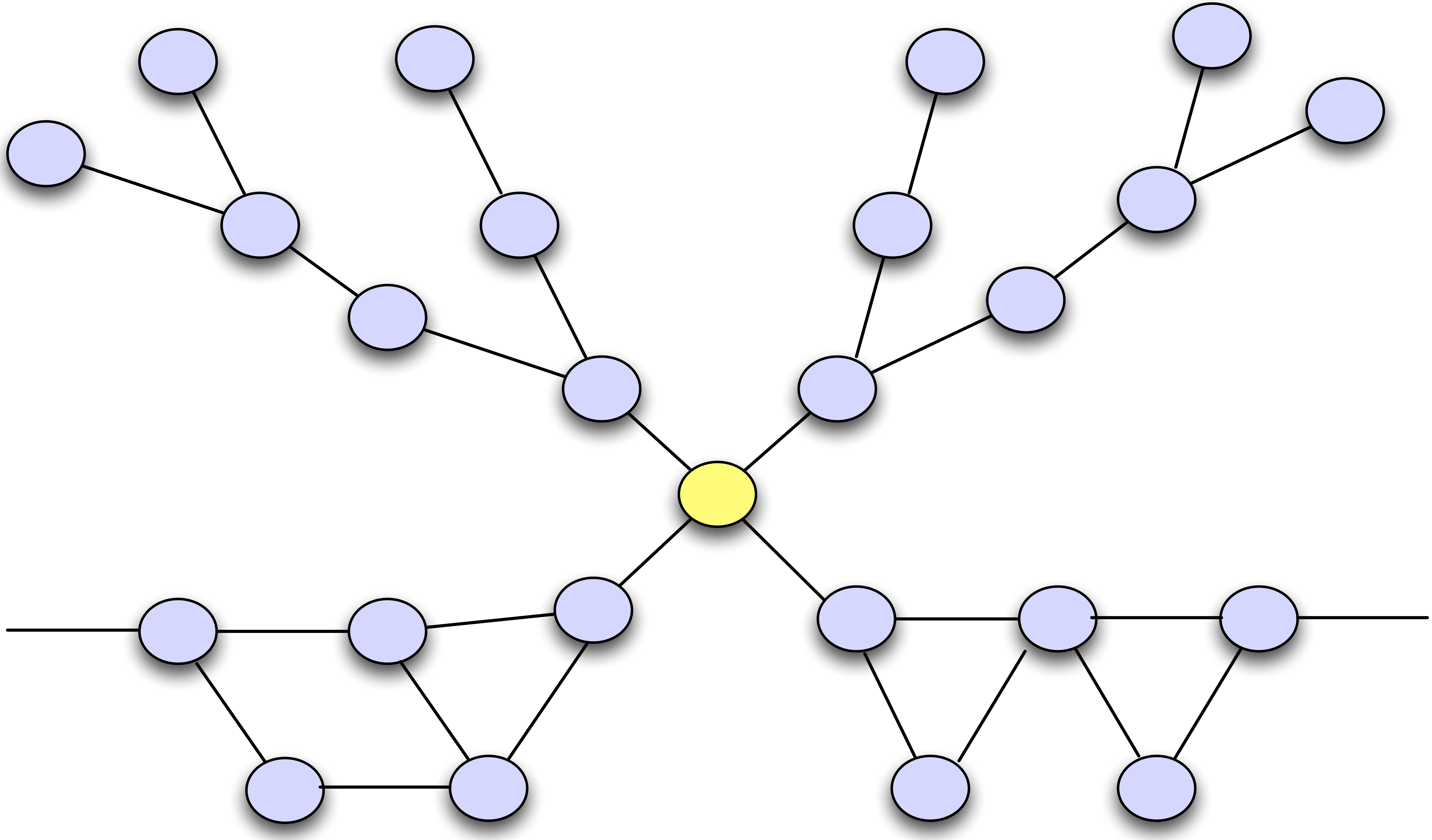
INTUITION: quick installation of backup multicast trees because fewer switches to signal

delay requirements are met using T_l

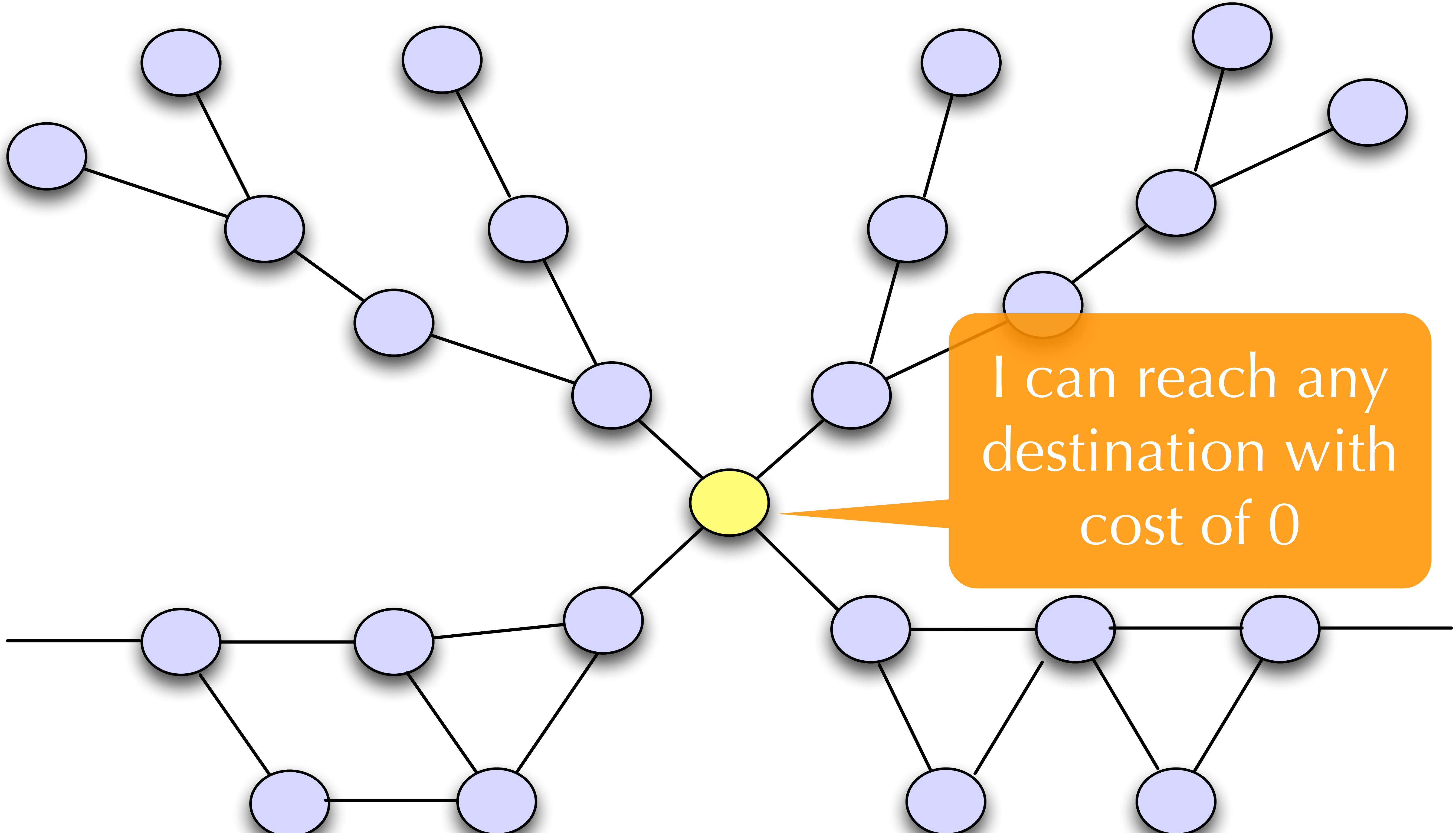
Talk Outline

- thesis introduction
- placement of smart grid sensors to enable measurement error detection
- recovery from failed communication links in a smart grid
- recovery from malicious nodes injecting false routing state
- outline for future work and conclusions

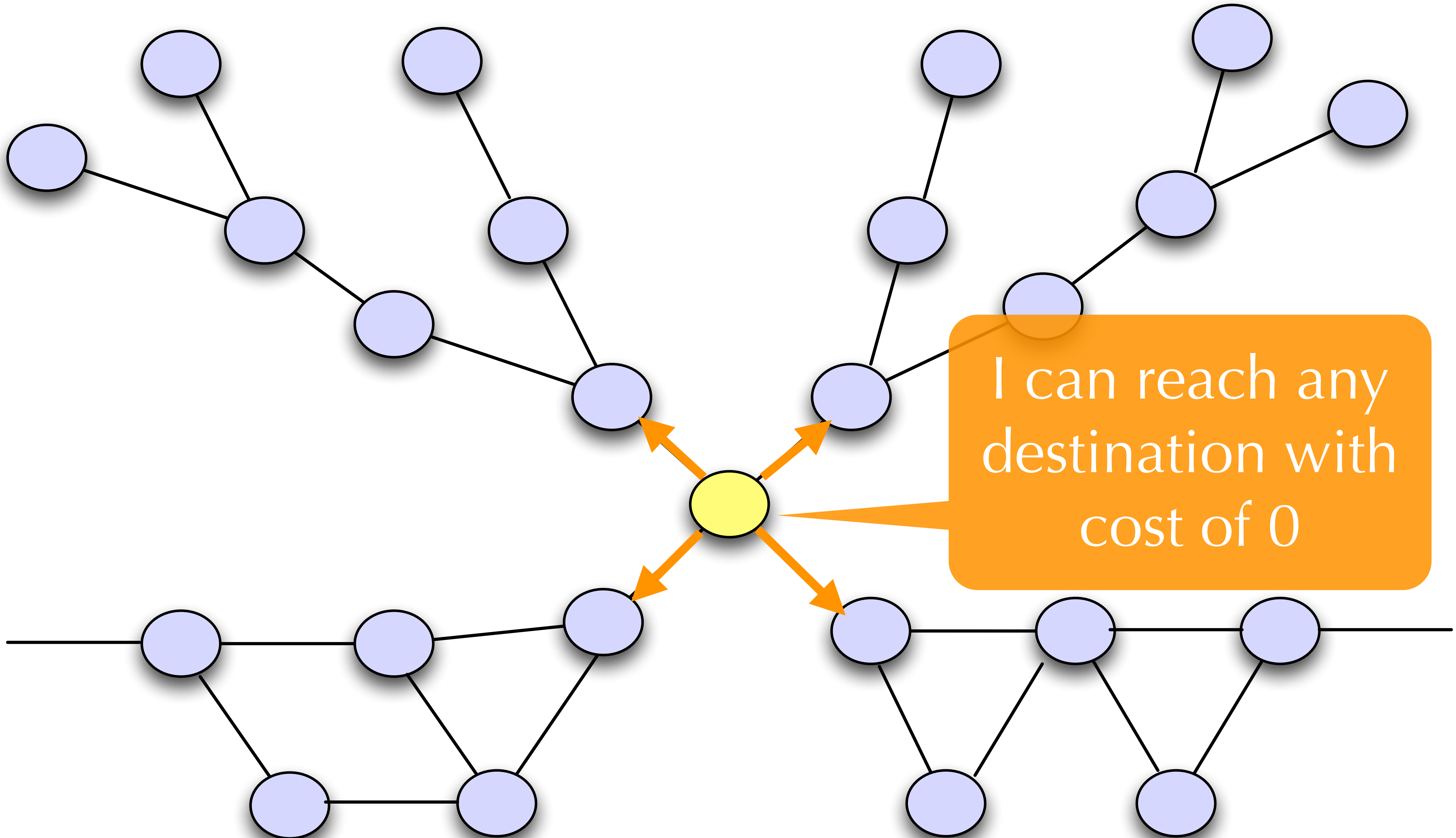
Ch 3: Network Router Failure



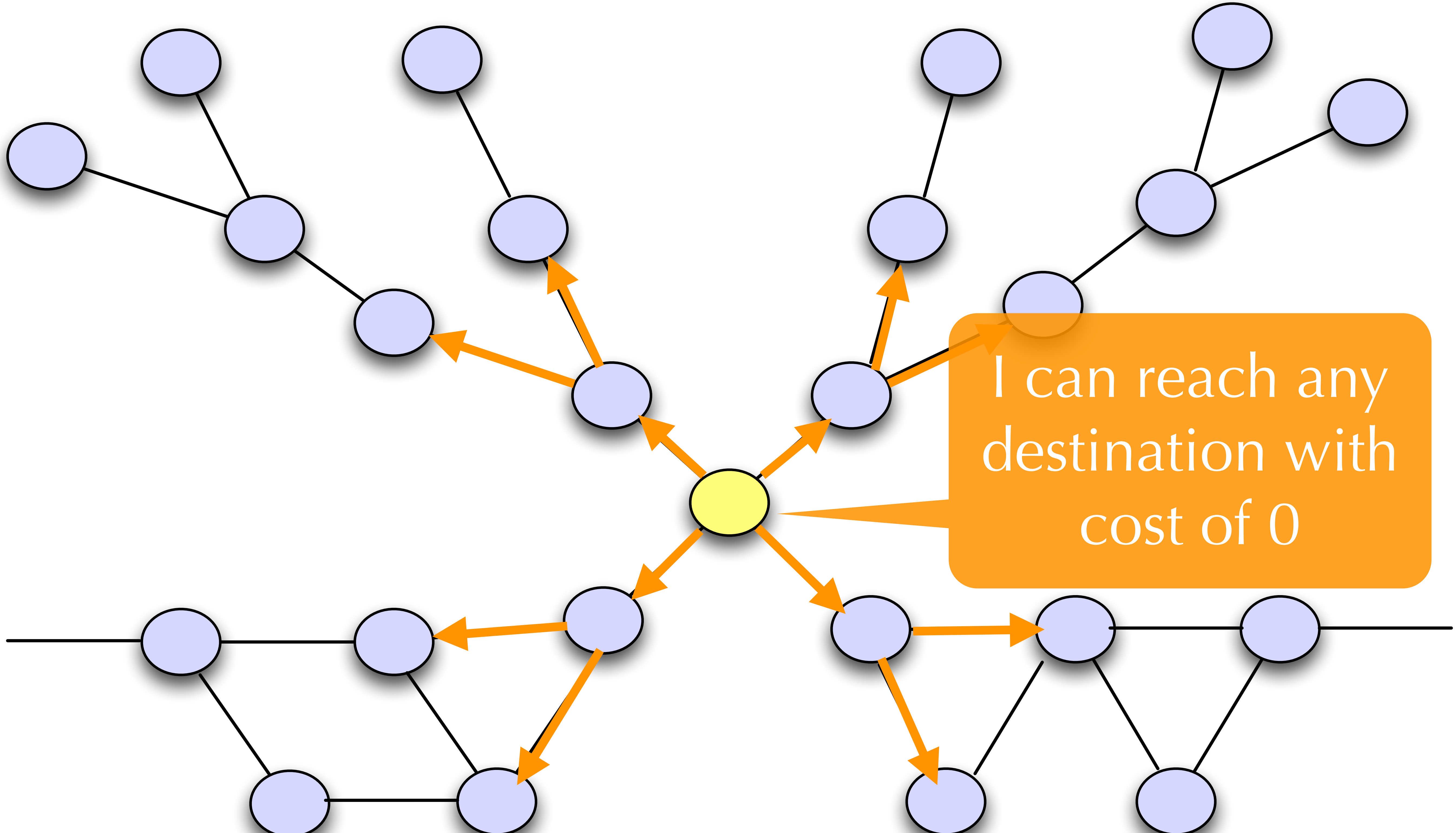
Ch 3: Network Router Failure



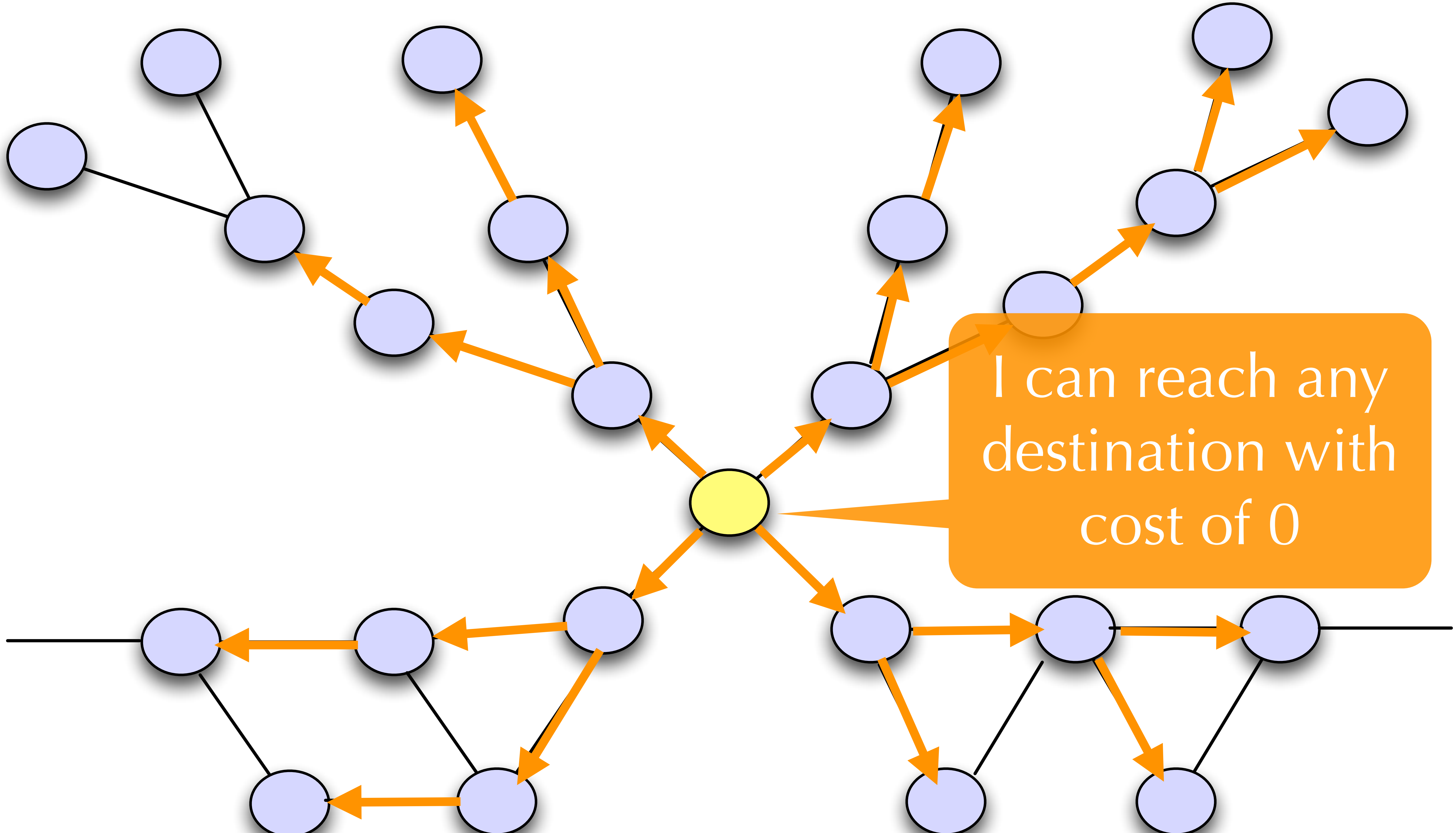
Ch 3: Network Router Failure



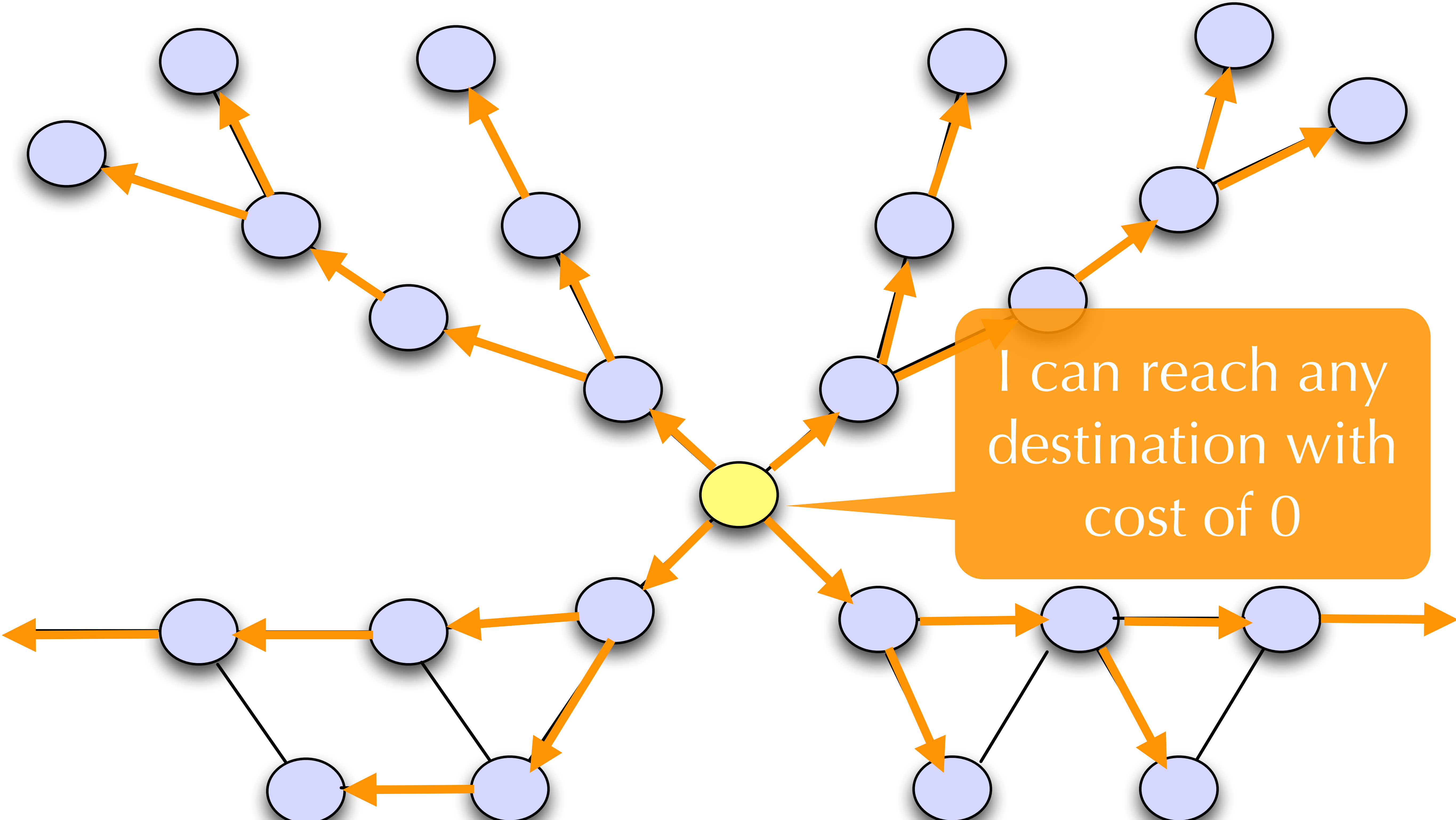
Ch 3: Network Router Failure



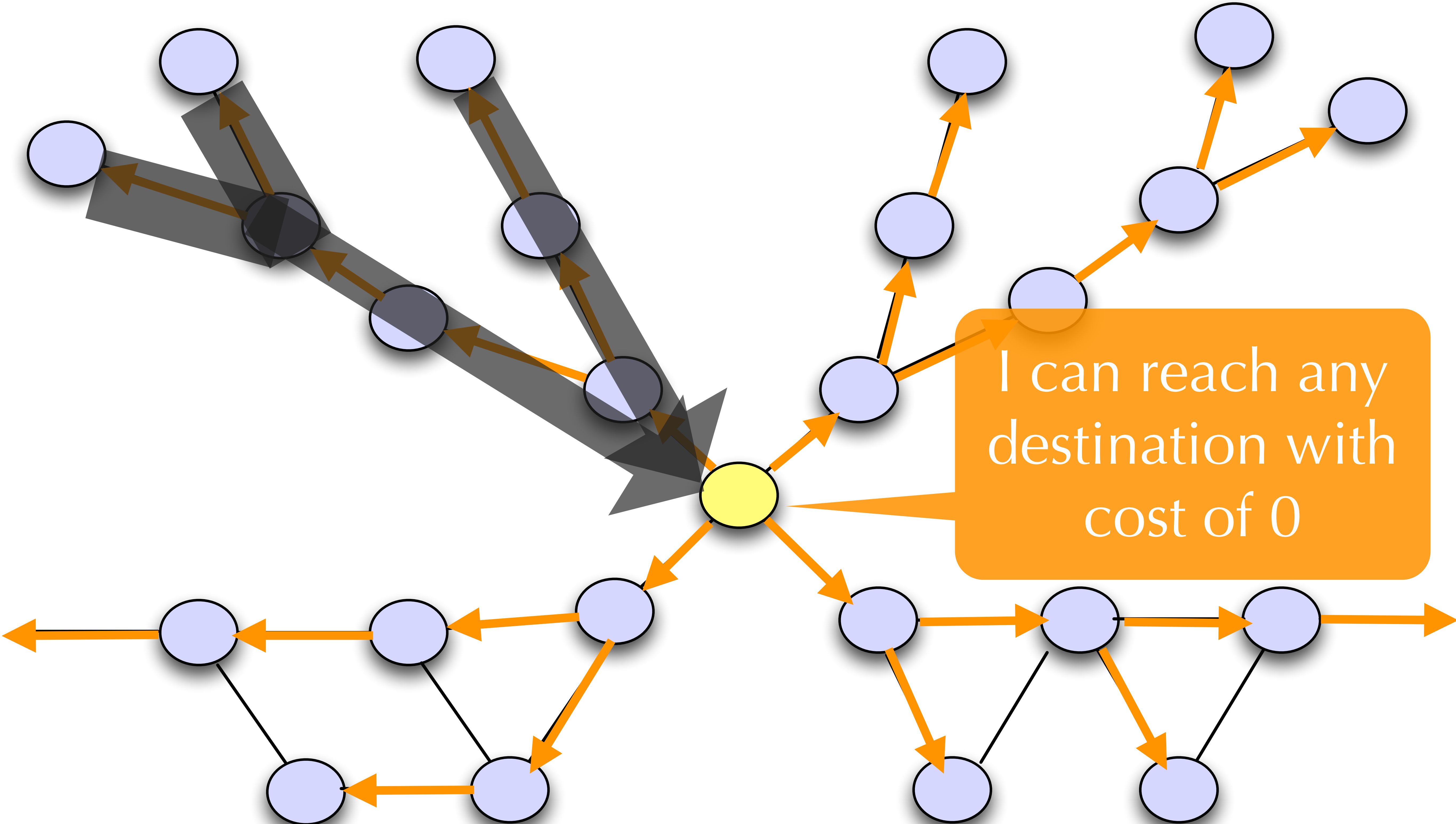
Ch 3: Network Router Failure



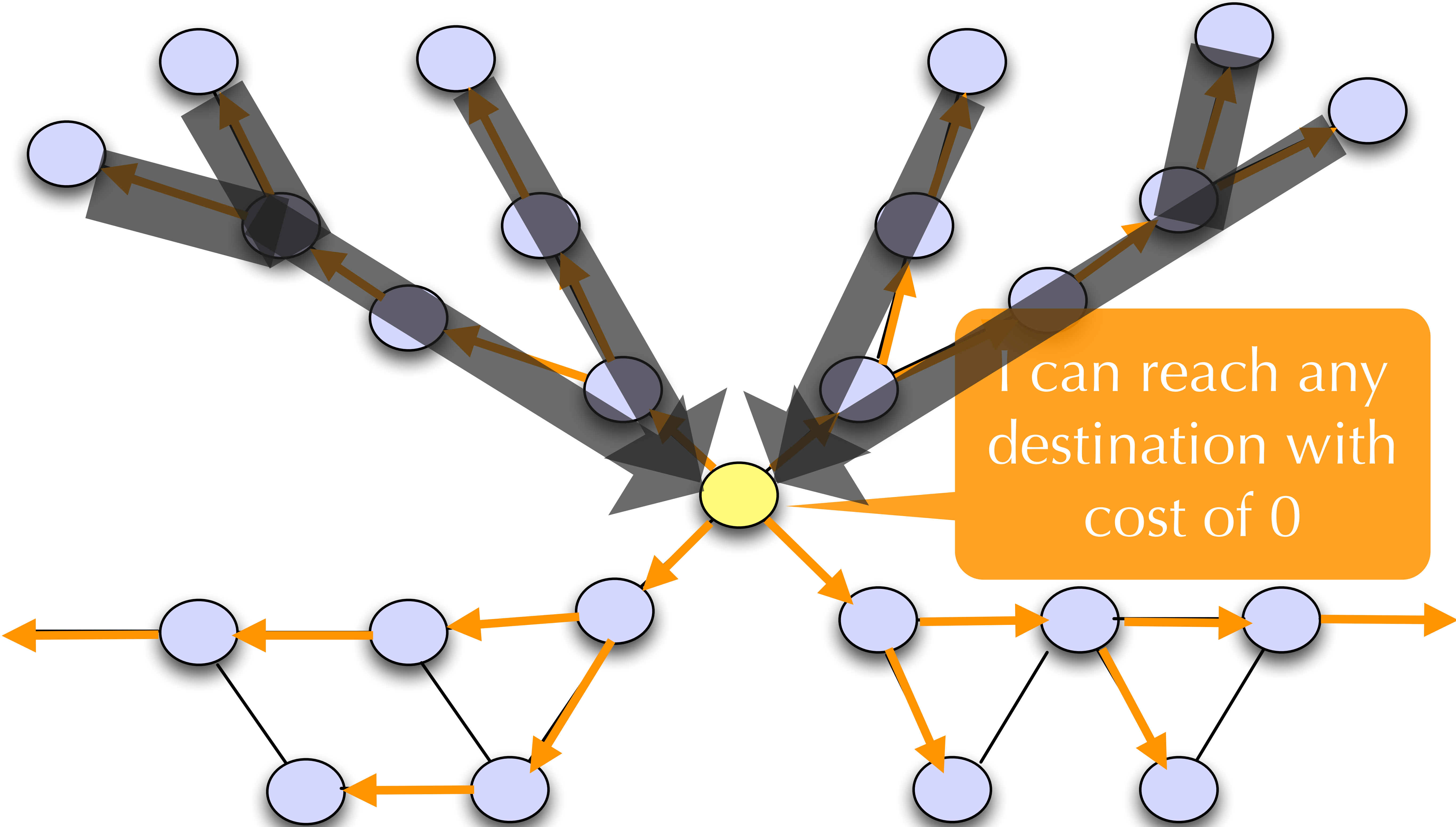
Ch 3: Network Router Failure



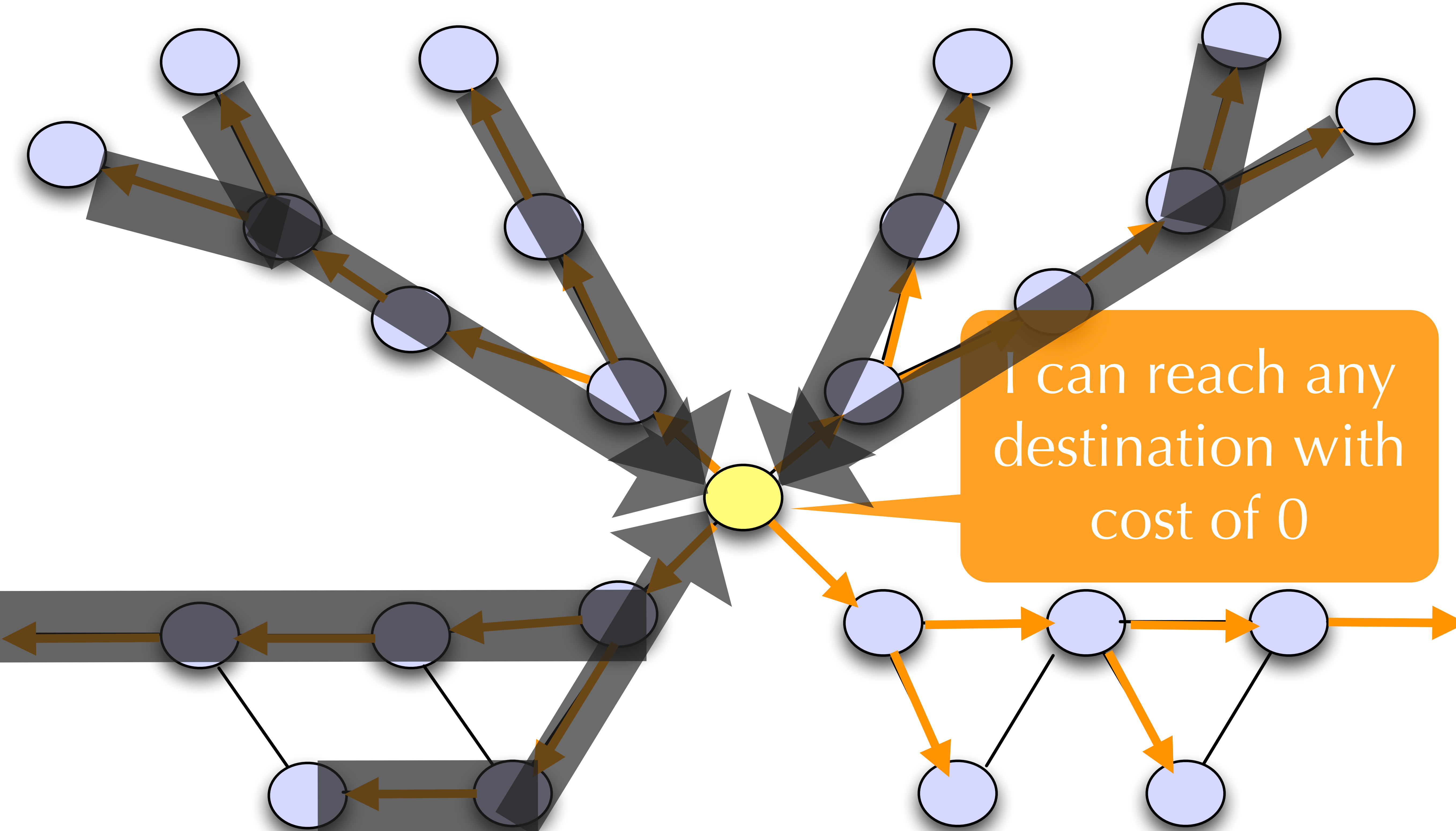
Ch 3: Network Router Failure



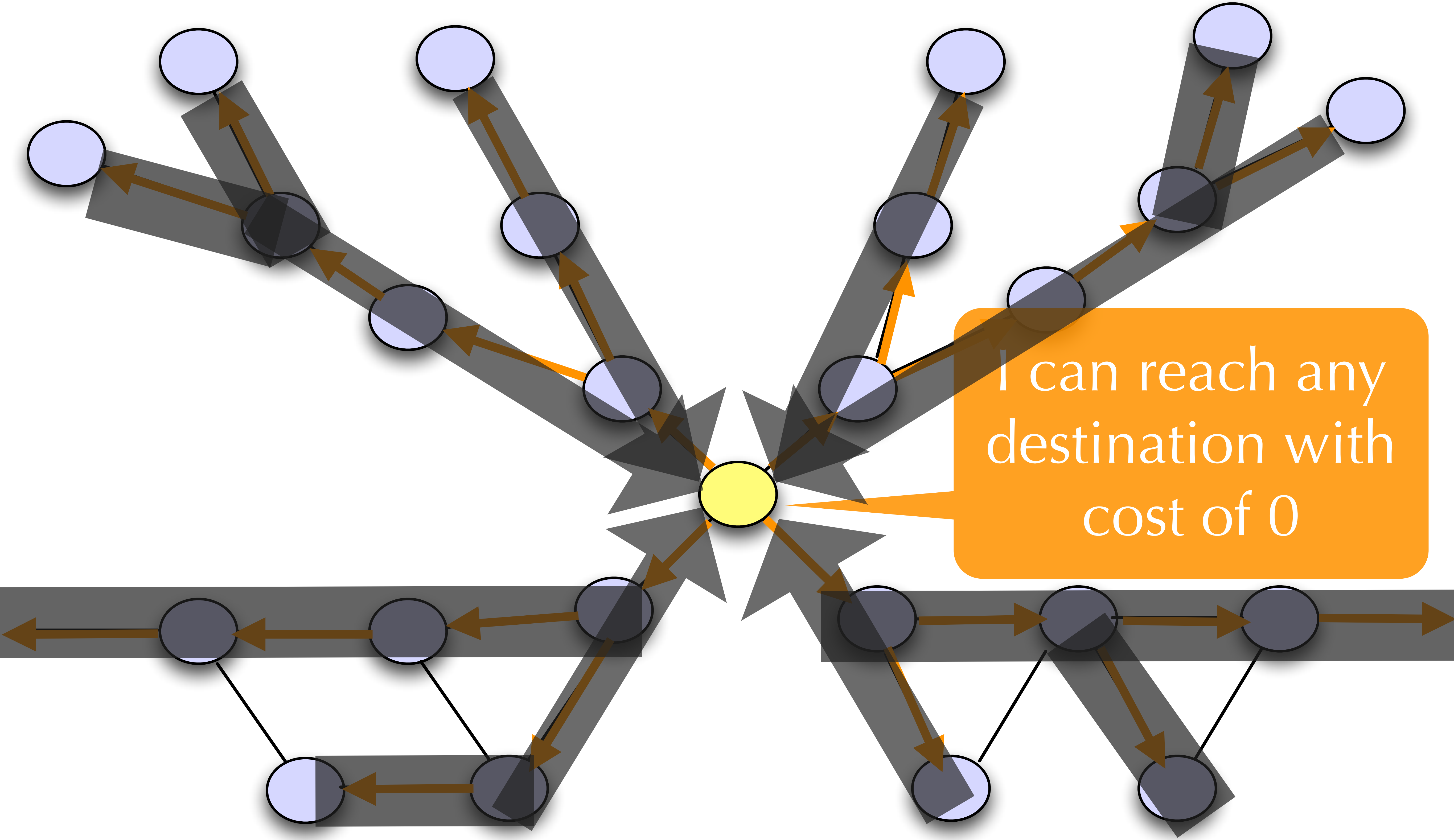
Ch 3: Network Router Failure



Ch 3: Network Router Failure



Ch 3: Network Router Failure



Problem Description

Problem Description

1. nodes share false routing state

Problem Description

1. nodes share false routing state
2. outside algorithm identifies compromised node

Problem Description

1. nodes share false routing state
2. outside algorithm identifies compromised node
3. recover

Problem Description

1. nodes share false routing state
2. outside algorithm identifies compromised node
3. recover
    a. remove compromised nodes from graph

Problem Description

1. nodes share false routing state
2. outside algorithm identifies compromised node
3. recover
    a. remove compromised nodes from graph
    b. compute least cost paths that route around compromised nodes

2nd BEST Recovery Algorithm

2nd BEST Recovery Algorithm

2nd BEST

2nd BEST Recovery Algorithm

2nd BEST

1. locally remove false routing state

2nd BEST Recovery Algorithm

2nd BEST

1. locally remove false routing state
    - neighbors of compromised node set cost to compromised to ∞

2nd BEST Recovery Algorithm

2nd BEST

1. locally remove false routing state
   - neighbors of compromised node set cost to compromised to ∞
2. initiate new distance vector computation to recompute new least cost paths that route around compromised node

2nd BEST Recovery Algorithm

2nd BEST

1. locally remove false routing state
   - neighbors of compromised node set cost to compromised to ∞
2. initiate new distance vector computation to recompute new least cost paths that route around compromised node

(+) simple, (−) risk of routing loops

PURGE Recovery Algorithm

PURGE Recovery Algorithm

PURGE

PURGE Recovery Algorithm

PURGE

1. globally remove false routing state

PURGE Recovery Algorithm

PURGE

1. globally remove false routing state
   - use diffusing computations

PURGE Recovery Algorithm

PURGE

1. globally remove false routing state
   - use diffusing computations
2. initiate new distance vector computation to recompute new least cost paths that route around compromised node

PURGE Recovery Algorithm

PURGE

1. globally remove false routing state
   - use diffusing computations
2. initiate new distance vector computation to recompute new least cost paths that route around compromised node

(+) removes all routing loops

CPR Recovery Algorithm

CPR Recovery Algorithm

CPR (**C**heck**P**oint and **R**ollback)

CPR Recovery Algorithm

CPR (**C**heck**P**oint and **R**ollback)

- each node takes snapshots of routing table

CPR Recovery Algorithm

CPR (**C**heck**P**oint and **R**ollback)

- each node takes snapshots of routing table

1. Each node rolls back to checkpoint taken before node is compromised

CPR Recovery Algorithm

CPR (**C**heck**P**oint and **R**ollback)

- each node takes snapshots of routing table

1. Each node rolls back to checkpoint taken before node is compromised
2. initiate new distance vector computation w/ each node using routing tables from step #1

CPR Recovery Algorithm

CPR (**C**heck**P**oint and **R**ollback)

- each node takes snapshots of routing table

1. Each node rolls back to checkpoint taken before node is compromised
2. initiate new distance vector computation w/ each node using routing tables from step #1
    - computes new least cost paths that route around compromised node + updates any stale state

CPR Recovery Algorithm

CPR (**C**heck**P**oint and **R**ollback)

- each node takes snapshots of routing table

1. Each node rolls back to checkpoint taken before node is compromised
2. initiate new distance vector computation w/ each node using routing tables from step #1
   - computes new least cost paths that route around compromised node + updates any

(+) remove stale state w/ rollback, (–) requires synchronized clocks, (–) storage overhead

Related Work

- PURGE is similar to Garcia-Luna-Aceves DUAL algorithm for loop-free routing
  - both use diffusing computations
- CPR borrows ideas from
  - database crash recovery
  - recovery from malicious but committed database transactions

Simulation Scenario

Simulation Scenario

1. all nodes correctly compute least cost paths

Simulation Scenario

1. all nodes correctly compute least cost paths
2. a node is compromised and claims cost of "1" to every node

Simulation Scenario

1. all nodes correctly compute least cost paths
2. a node is compromised and claims cost of "1" to every node
3. nodes notified of compromised

Simulation Scenario

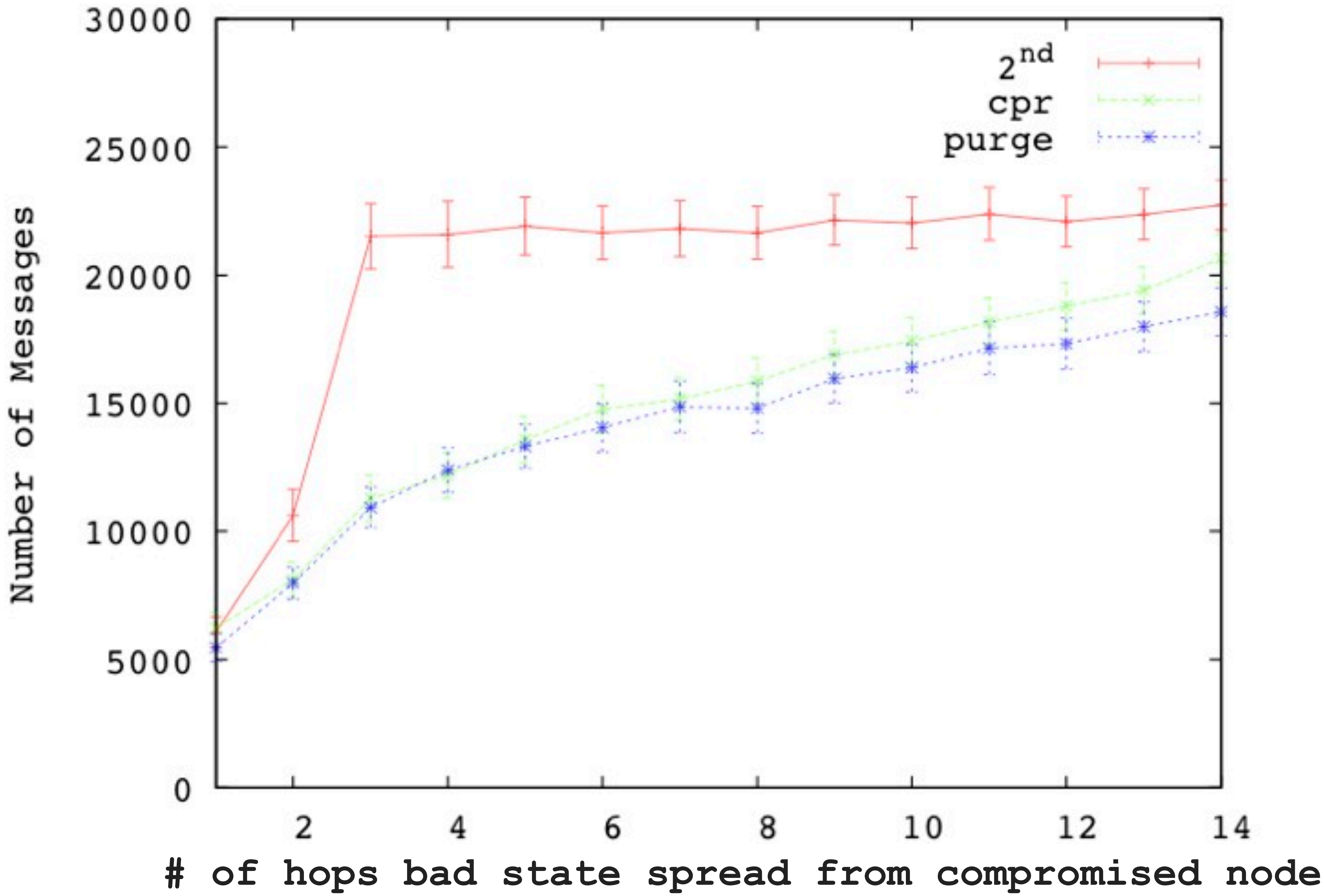
1. all nodes correctly compute least cost paths
2. a node is compromised and claims cost of "1" to every node
3. nodes notified of compromised
4. run recovery algorithm (messages counted here)

Simulation Scenario

1. all nodes correctly compute least cost paths
2. a node is compromised and claims cost of "1" to every node
3. nodes notified of compromised
4. run recovery algorithm (messages counted here)

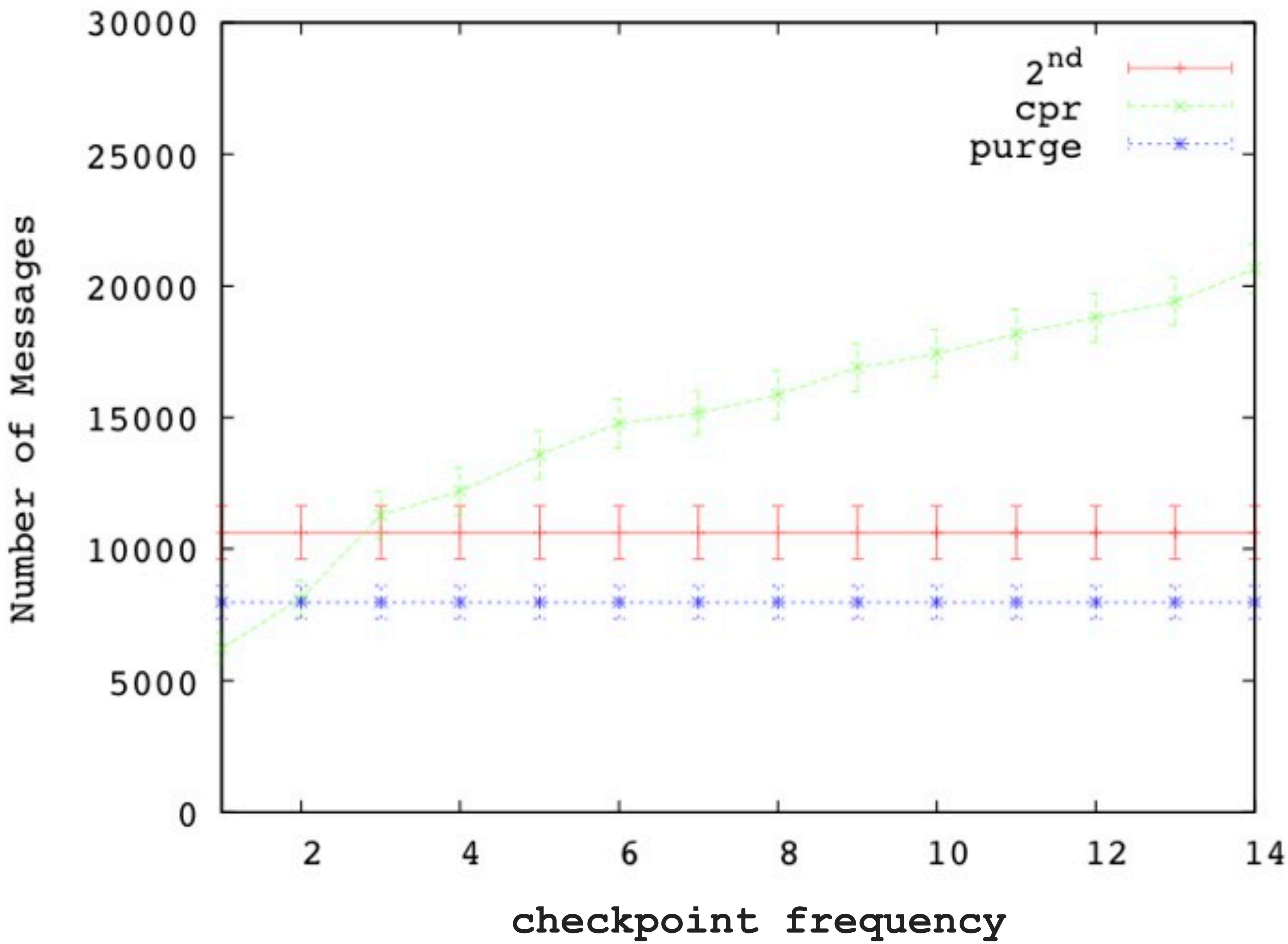
- show results for Erdos-Renyi graphs
  - `n=100`, link weights between `[1,100]`

msgs vs # hops bad state spreads



- 2ND BEST: many routing loops
- CPR: has stale state after rollback + assumes synch clocks
- PURGE: no routing loops + no stale state during recovery

How to Set Checkpoint Frequency?



- CPR: less frequent checkpoints => more overhead
- 2ND BEST and PURGE: constant overhead b/c neither algorithm checkpoints

Talk Outline

- thesis introduction
- placement of smart grid sensors to enable measurement error detection
- recovery from failed communication links in a smart grid
- recovery from malicious nodes injecting false routing state
- outline for future work and conclusions

Thesis Timeline: Completed Work

Thesis Timeline: Completed Work

- Ch 1: “Recovery from False Routing State in Distributed Routing Algorithms”
  - published in *IFIP Networking 2010 Conference*
  - done (unless committee has suggestions)

Thesis Timeline: Completed Work

- Ch 1: “Recovery from False Routing State in Distributed Routing Algorithms”
    - published in *IFIP Networking 2010 Conference*
    - done (unless committee has suggestions)
- Ch 2: “PMU Sensor Placement for Measurement Error Detection in the Smart Grid”
    - published in *e-Energy 2012*
    - done (unless committee has suggestions)

Thesis Timeline: Work in Progress

- Ch. 3: "Recovery from Link Failures in Smart Grid Communication Network"
  - problem well-defined
  - algorithms, implementation, analysis, and evaluation yet to be completed

Thesis Timeline: Work In Progress

Thesis Timeline: Work In Progress

- Ch. 3 milestones

Thesis Timeline: Work In Progress

- Ch. 3 milestones
  - implement FAILED-LINK using POX Openflow controller

Thesis Timeline: Work In Progress

- Ch. 3 milestones
  - implement FAILED-LINK using POX Openflow controller
  - test + profile FAILED-LINK

Thesis Timeline: Work In Progress

- Ch. 3 milestones
  - implement FAILED-LINK using POX Openflow controller
  - test + profile FAILED-LINK
  - implement MIN-FLOWS, MIN-SINKS, + MIN-CONTROL using POX Openflow controller

Thesis Timeline: Work In Progress

- Ch. 3 milestones
  - implement FAILED-LINK using POX Openflow controller
  - test + profile FAILED-LINK
  - implement MIN-FLOWS, MIN-SINKS, + MIN-CONTROL using POX Openflow controller
  - complexity analysis

Thesis Timeline: Work In Progress

- Ch. 3 milestones
  - implement FAILED-LINK using POX Openflow controller
  - test + profile FAILED-LINK
  - implement MIN-FLOWS, MIN-SINKS, + MIN-CONTROL using POX Openflow controller
  - complexity analysis
  - simulation-based study of FAILED-LINK, MIN-FLOWS, MIN-SINKS, MIN-CONTROL

Thesis Timeline: Work In Progress

- Ch. 3 milestones
  - implement FAILED-LINK using POX Openflow controller **3 weeks - Feb**
  - test + profile FAILED-LINK
  - implement MIN-FLOWS, MIN-SINKS, + MIN-CONTROL using POX Openflow controller
  - complexity analysis
  - simulation-based study of FAILED-LINK, MIN-FLOWS, MIN-SINKS, MIN-CONTROL

Thesis Timeline: Work In Progress

- Ch. 3 milestones
  - implement FAILED-LINK using POX Openflow controller **3 weeks - Feb**
  - test + profile FAILED-LINK **2 weeks - March**
  - implement MIN-FLOWS, MIN-SINKS, + MIN-CONTROL using POX Openflow controller
  - complexity analysis
  - simulation-based study of FAILED-LINK, MIN-FLOWS, MIN-SINKS, MIN-CONTROL

Thesis Timeline: Work In Progress

- Ch. 3 milestones
  - implement FAILED-LINK using POX Openflow controller — 3 weeks - Feb
  - test + profile FAILED-LINK — 2 weeks - March
  - implement MIN-FLOWS, MIN-SINKS, + MIN-CONTROL using POX Openflow — 4 weeks - May
  - complexity analysis
  - simulation-based study of FAILED-LINK, MIN-FLOWS, MIN-SINKS, MIN-CONTROL

Thesis Timeline: Work In Progress

- Ch. 3 milestones
  - implement FAILED-LINK using POX Openflow controller 3 weeks - Feb
  - test + profile FAILED-LINK 2 weeks - March
  - implement MIN-FLOWS, MIN-SINKS, + MIN-CONTROL using POX Openflow 4 weeks - May
  - complexity analysis 2 weeks - May
  - simulation-based study of FAILED-LINK, MIN-FLOWS, MIN-SINKS, MIN-CONTROL

Thesis Timeline: Work In Progress

- Ch. 3 milestones
  - implement FAILED-LINK using POX Openflow controller
    3 weeks - Feb
  - test + profile FAILED-LINK
    2 weeks - March
  - implement MIN-FLOWS, MIN-SINKS, + MIN-CONTROL using POX Openflow
    4 weeks - May
  - complexity analysis
    2 weeks - May
  - simulation-based study of FAILED-LINK, MIN-FLOWS, MIN-SINKS, MIN-CONTROL
    3 weeks - June

Thesis Summary

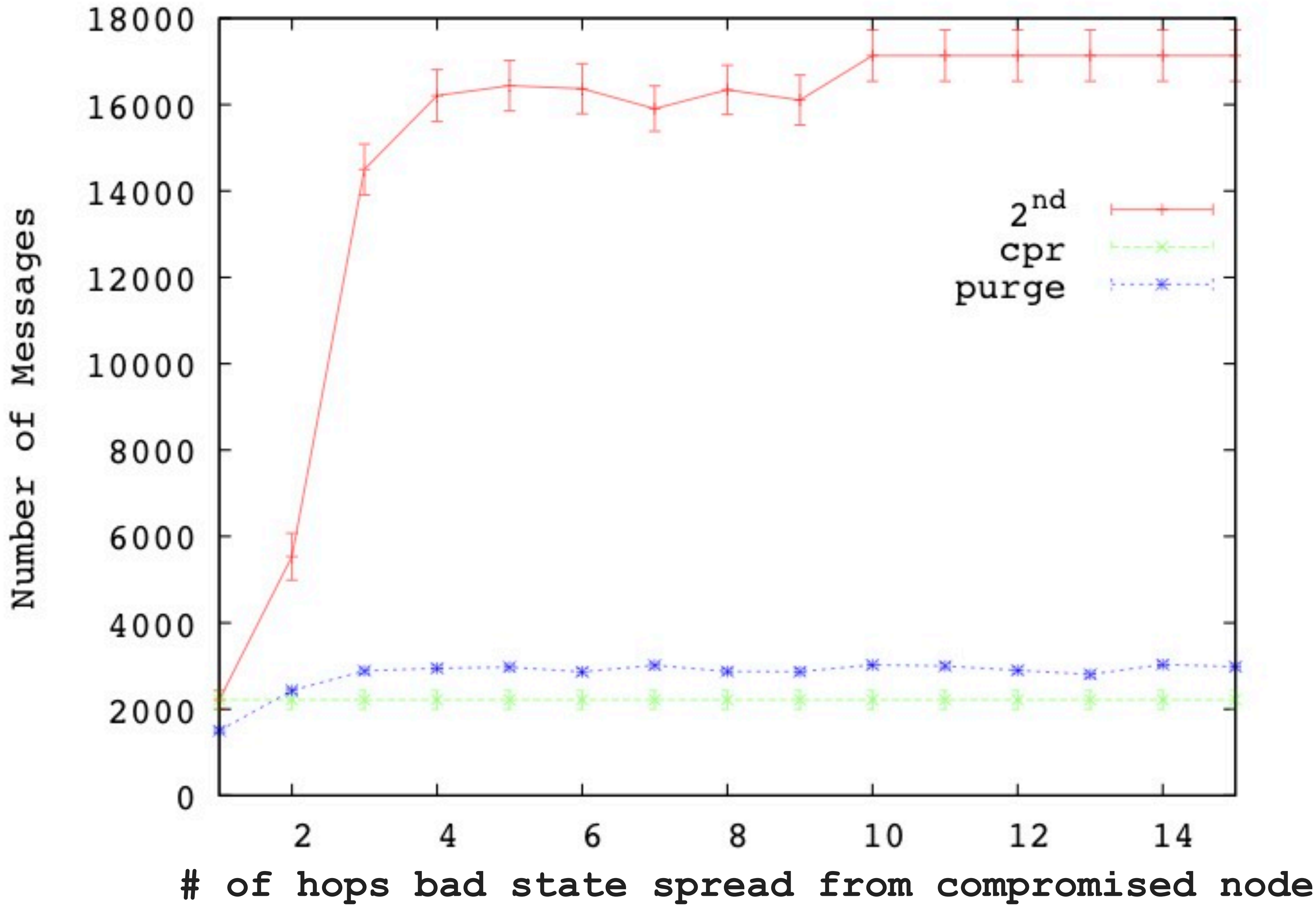
- consider failure of network components
  - router spreading false routing state
  - smart grid sensor measurement error
  - link failures in smart grid communication network
- proposed algorithms for automated recovery

Thank you.

Questions/Comments?

Backup Slides

Sim 1: Fixed Link Cost Graphs

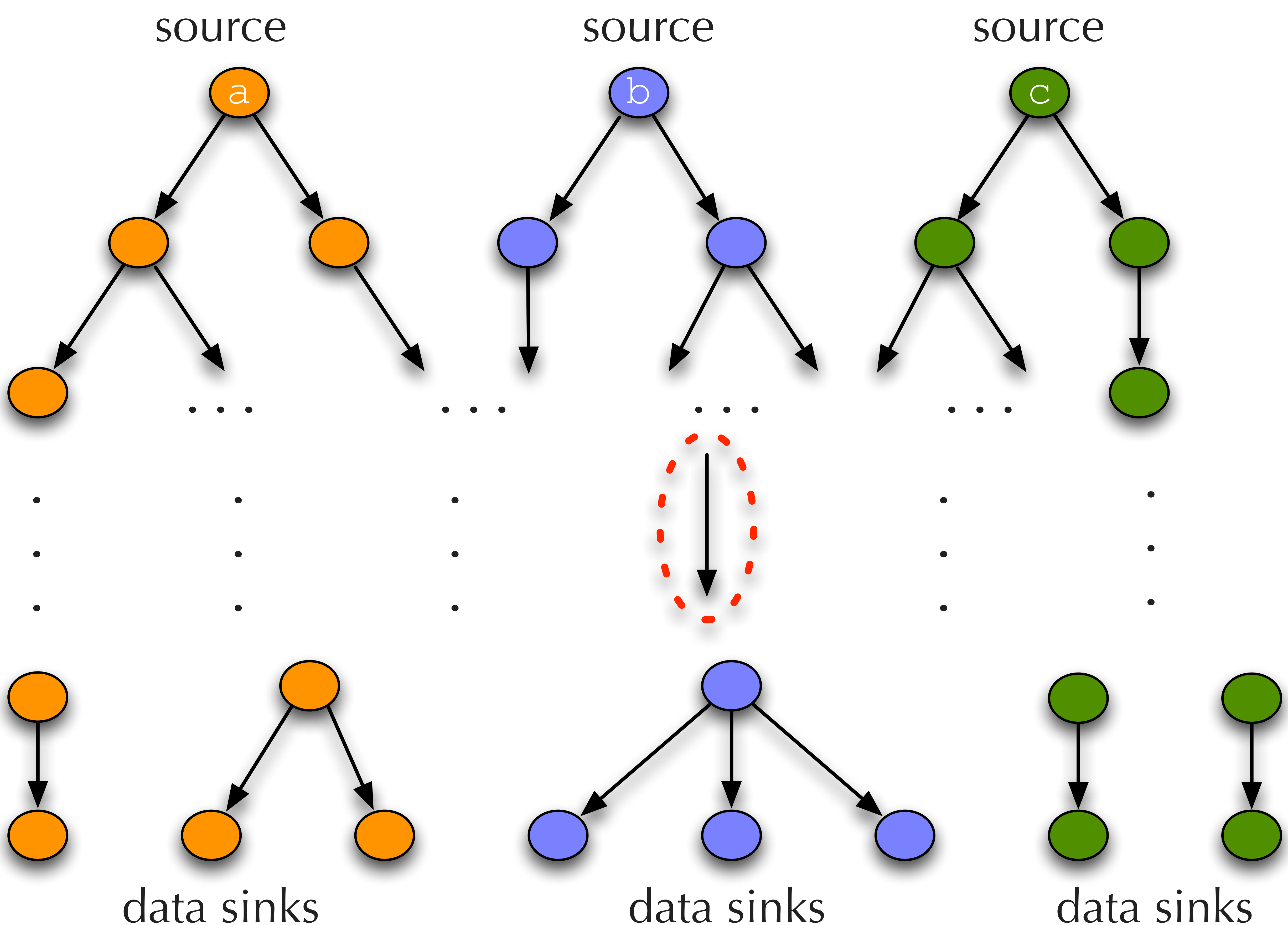


- 2ND BEST: many routing loops
- CPR: removes state with checkpoint and rollback
- 2ND BEST and PURGE: use iterative distance vector

Summary of Simulation Results

- 2ND BEST suffers from routing loops
- CPR is effective because rolling back quickly removes false routing state
- CPR assumes synchronized clocks
- PURGE removes routing loops and has no stale state

Communication Link Failures



Power Grid Data Dissemination

Power Grid Data Dissemination

- SCADA currently used
  - centralized polling
  - no wide-area dissemination

Power Grid Data Dissemination

- SCADA currently used
  - centralized polling
  - no wide-area dissemination
- many data sources
  - PMUs, IEDs, fault recording devices, ...

Power Grid Data Dissemination

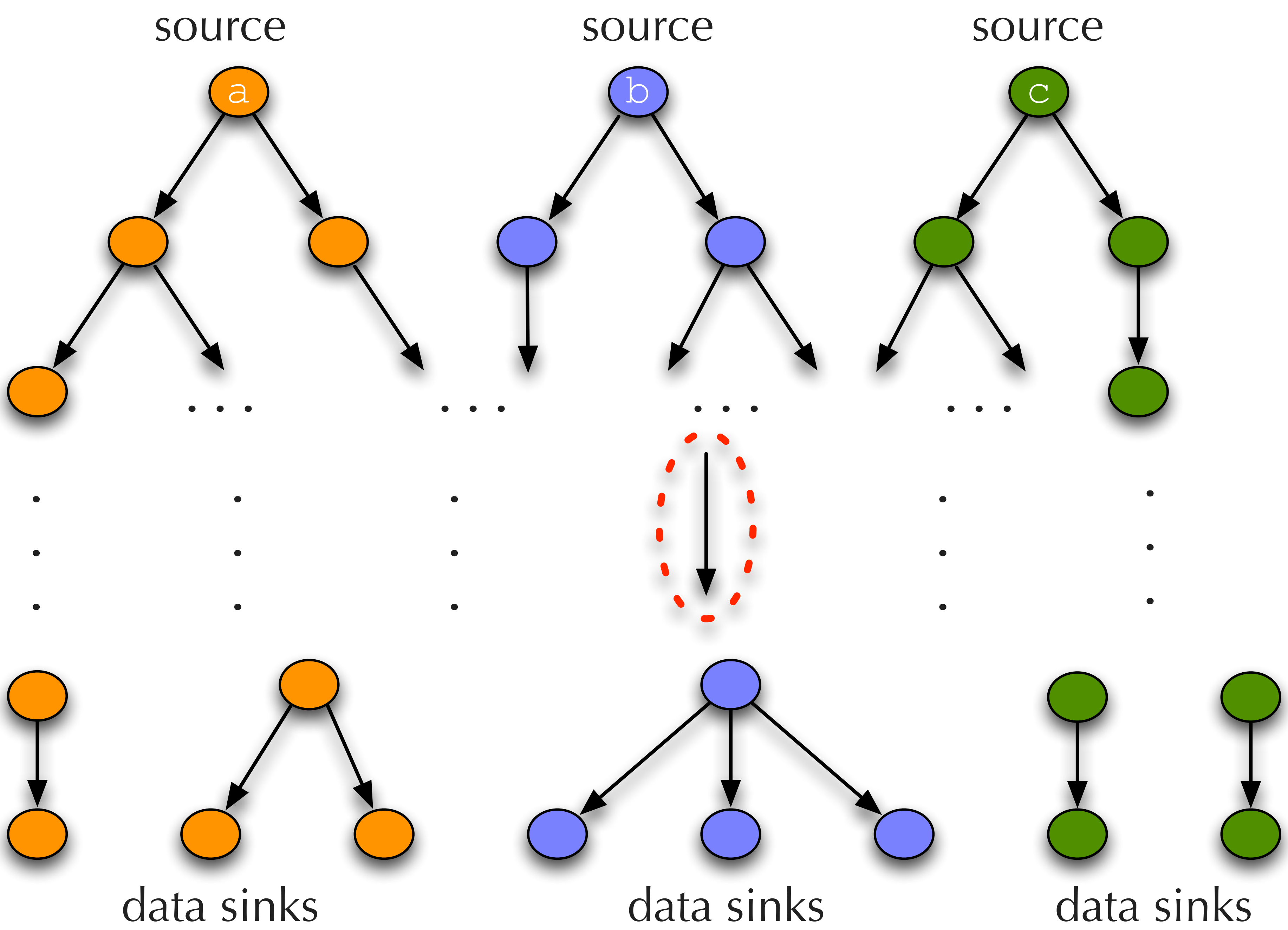
- SCADA currently used
  - centralized polling
  - no wide-area dissemination
- many data sources
  - PMUs, IEDs, fault recording devices, ...
- many sinks with interests in subset of data
  - utility companies, balancing authorites

Power Grid Data Dissemination

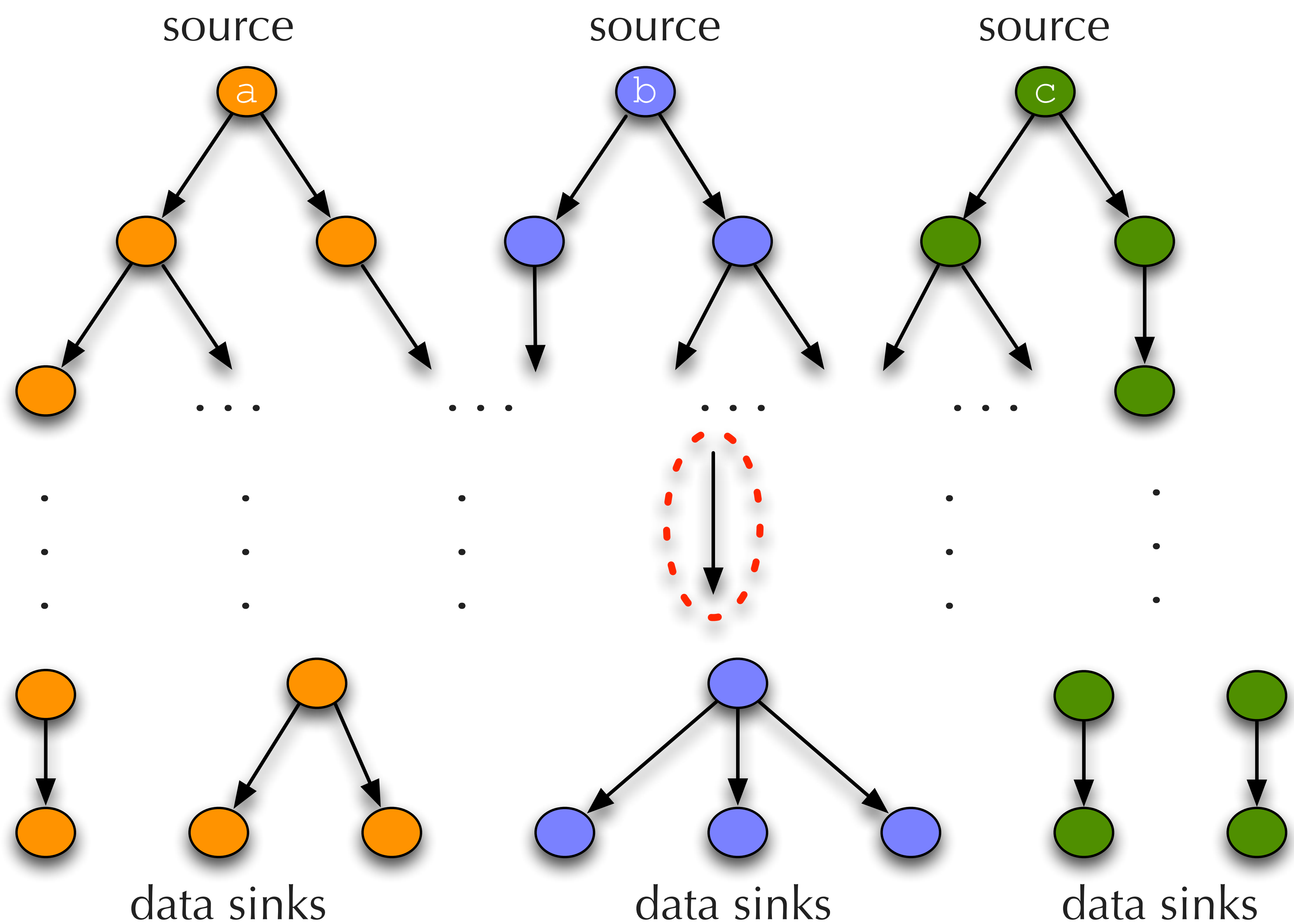
- SCADA currently used
  - centralized polling
  - no wide-area dissemination
- many data sources
  - PMUs, IEDs, fault recording devices, ...
- many sinks with interests in subset of data

great opportunity for richer wide-area data dissemination (big reason for the smart grid)

Problem Description

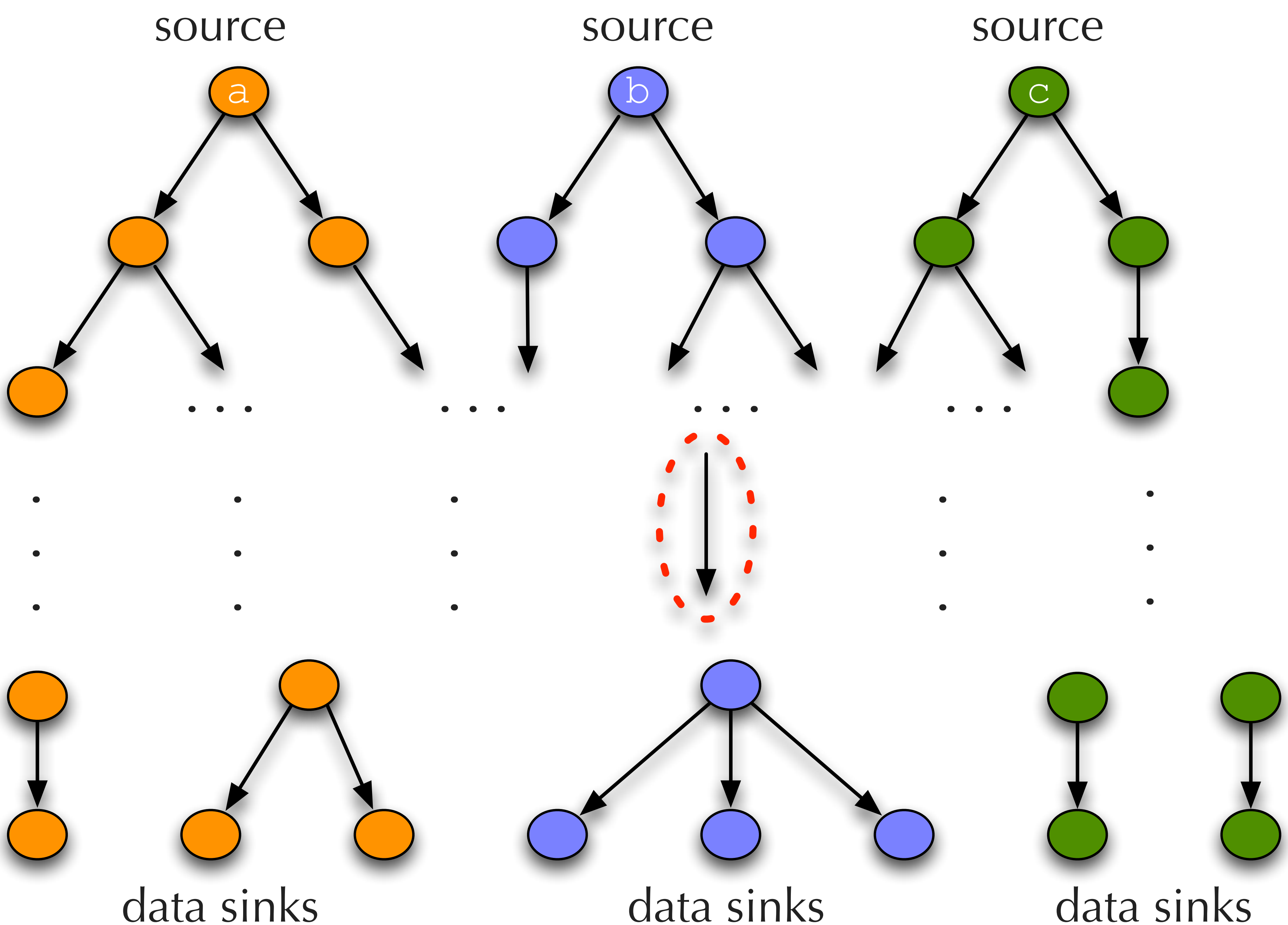


Problem Description



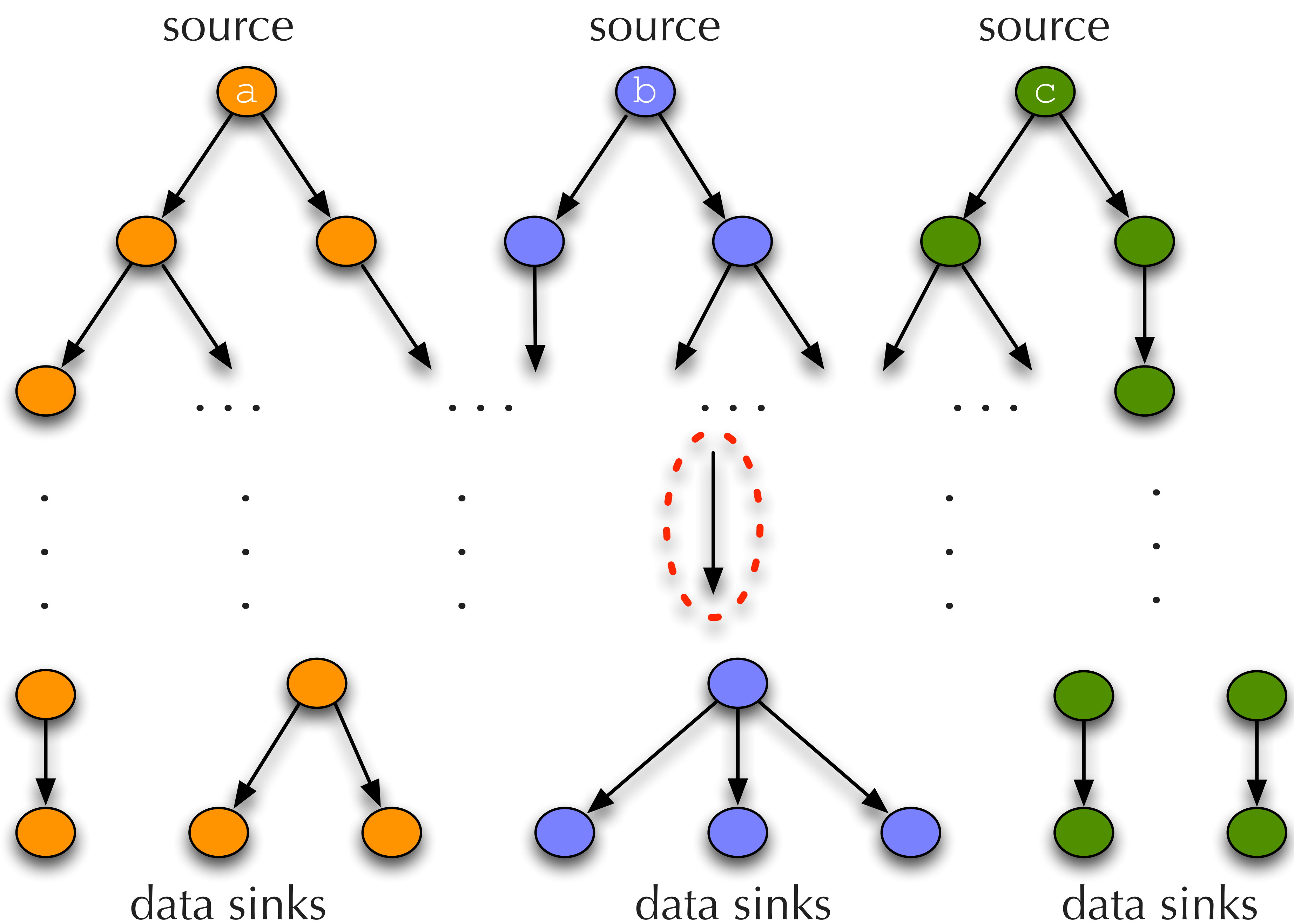
- multiple source-based MTs

Problem Description



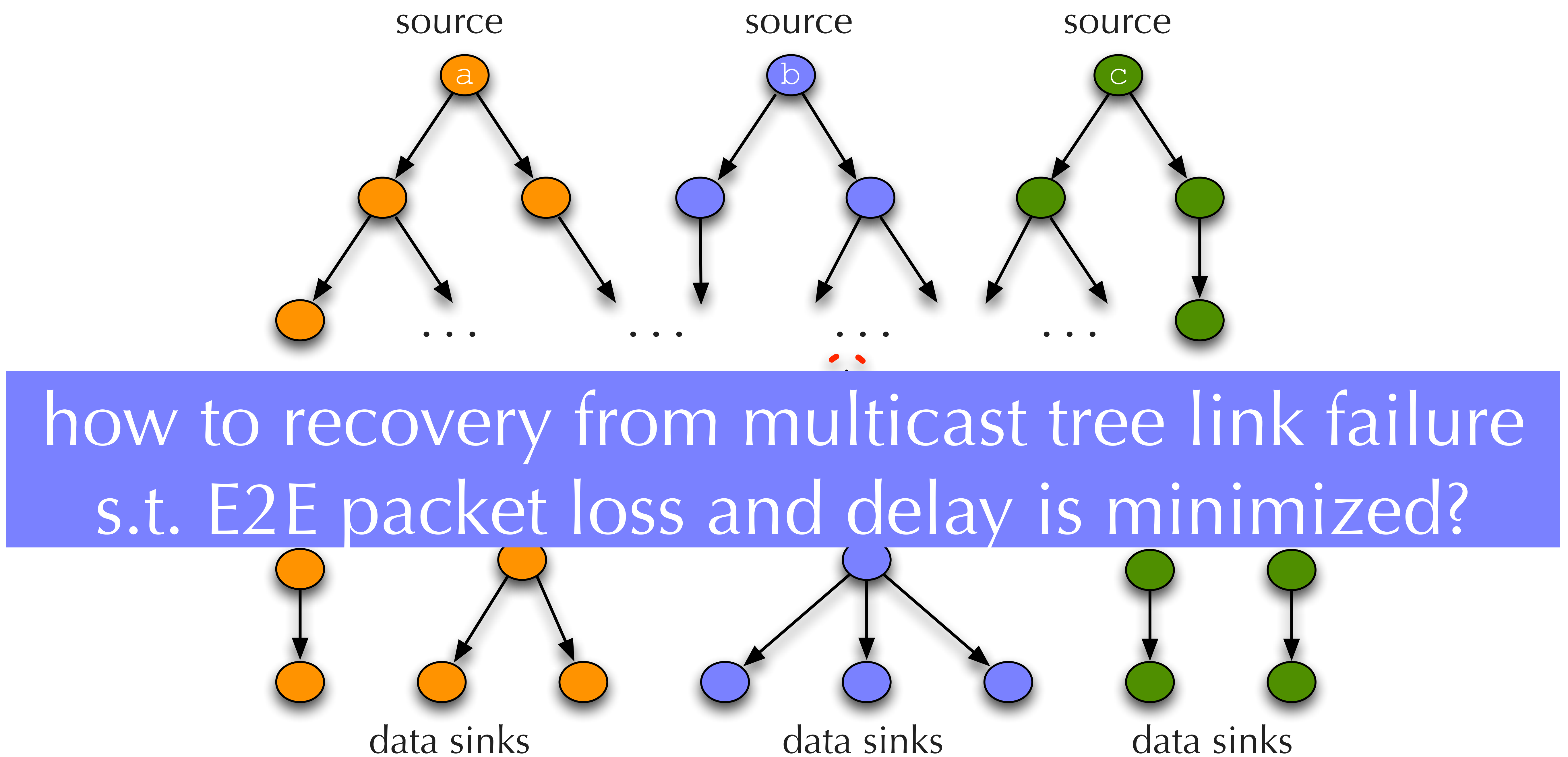
- multiple source-based MTs
- each (source,sink) pair has E2E per packet delay requirement

Problem Description



- multiple source-based MTs
- each (source,sink) pair has E2E per packet delay requirement
- single link fails at-a-time

Problem Description



- multiple source-based MTs
- each (source,sink) pair has E2E per packet delay requirement
- single link fails at-a-time

Smart Grid Data Dissemination

Smart Grid Data Dissemination

- problem characteristics

Smart Grid Data Dissemination

- problem characteristics
  - many data sources (PMUs, IEDs,...)

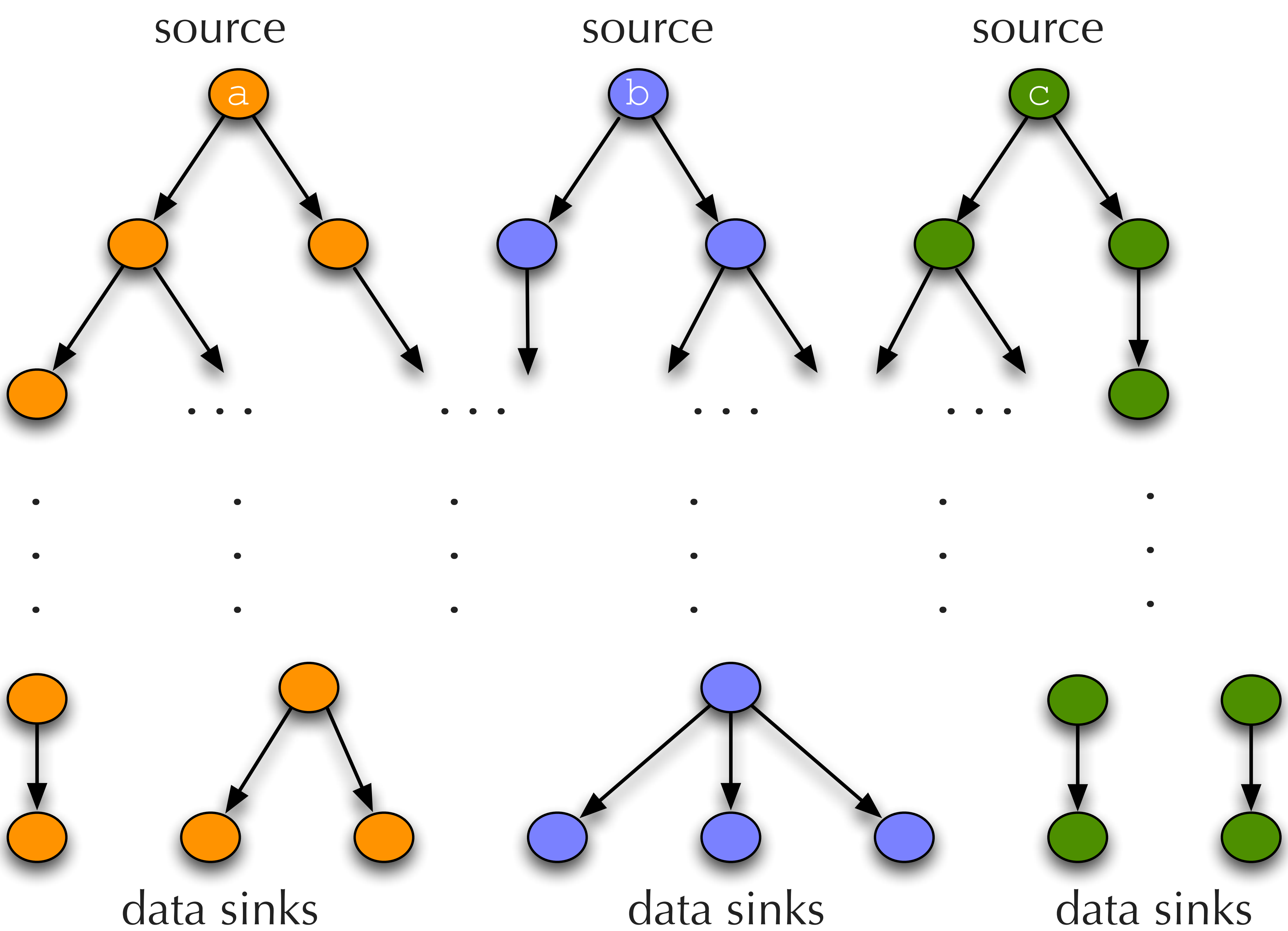
Smart Grid Data Dissemination

- problem characteristics
  - many data sources (PMUs, IEDs,...)
  - many sinks with interests in subset of data (utility companies, balancing authorities,...)

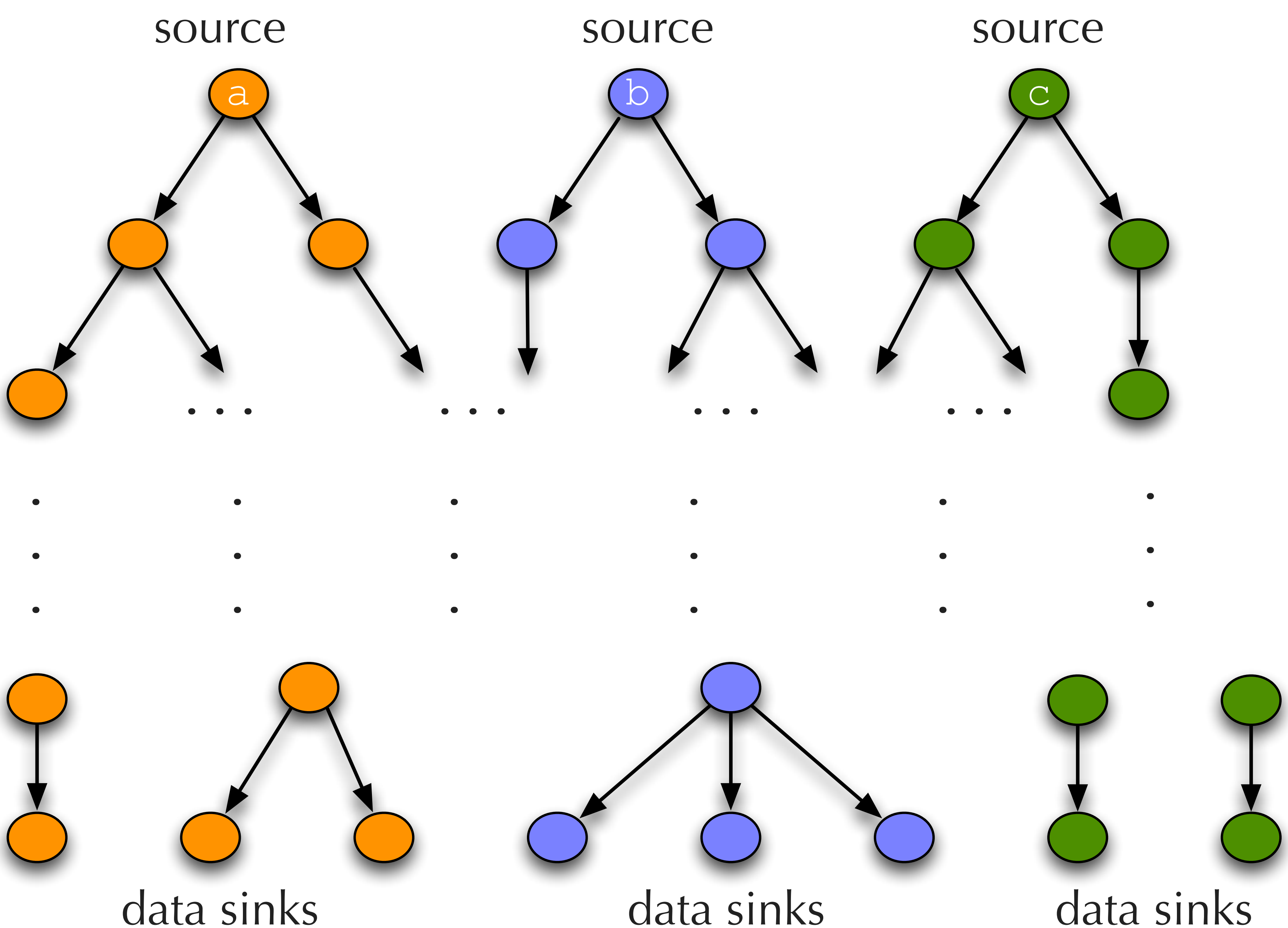
Smart Grid Data Dissemination

- problem characteristics
  - many data sources (PMUs, IEDs,...)
  - many sinks with interests in subset of data (utility companies, balancing authorities,...)
  - predictable fixed rate traffic w/ static sender and receiver sets

Problem Description

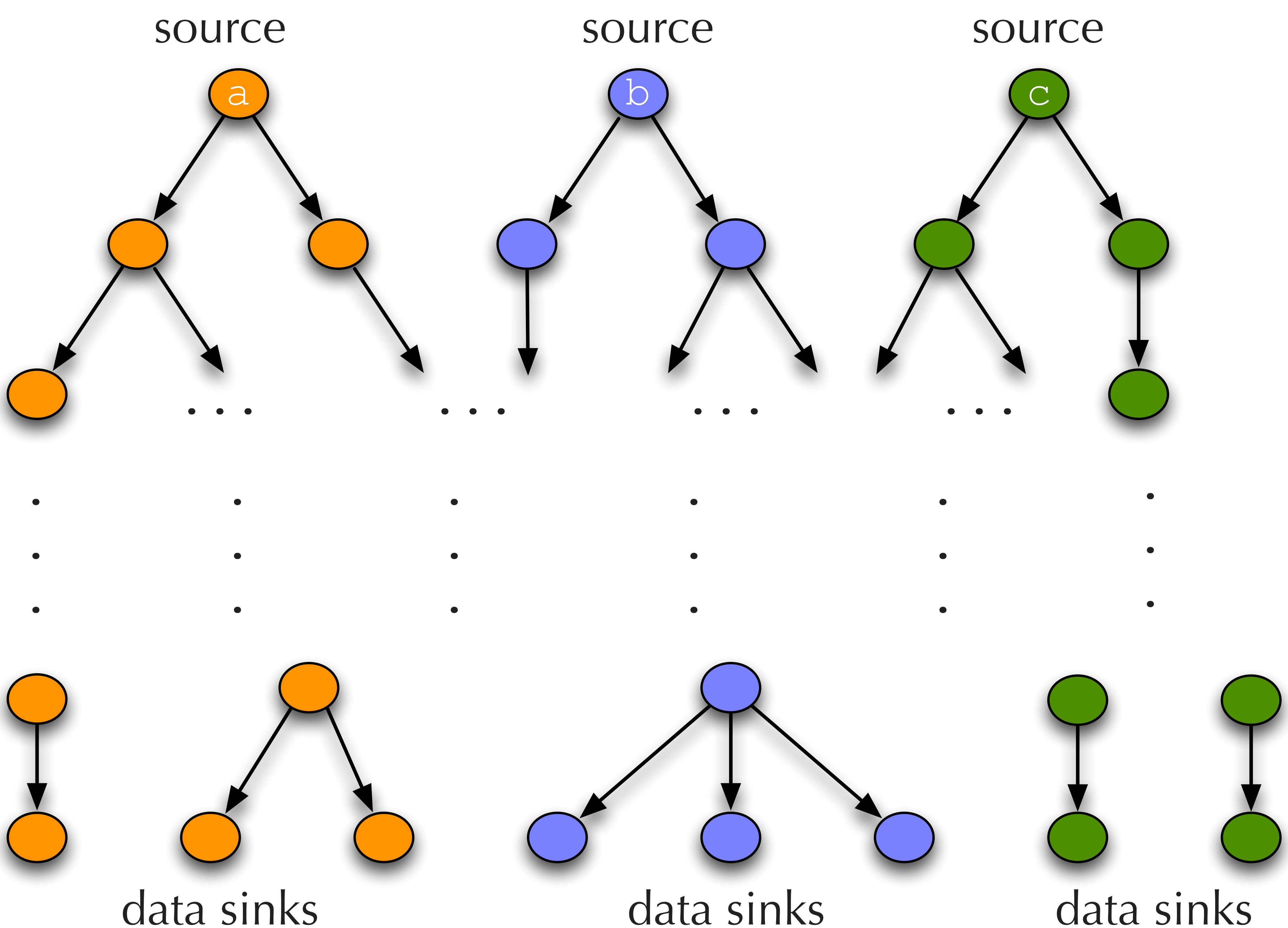


Problem Description



- multiple source-based MTs

Problem Description



- multiple source-based MTs
- each (source,sink) pair has E2E per packet delay requirement