Sympathy for the Sensor Network Debugger

Nithya Ramanathan
Kevin Chang
Eddie Kohler
Deborah Estrin

Sports

Posted 5/12/2005 9:53 PM

Sports Television

Michael Hiestand

Stern sees technology, global reach as NBA's challenge of TV future

Just about all pro sports realize they help themselves when they help TV.

But NBA Commissioner David Stern is unusually open-minded. Asked if TV has gone as far as it can in getting inside NBA games, Stern says, "We're far from seeing the end of it."

After all, he says, "what if they used sensors, dust-like particles, reporting (on-court) activity? I'm not sure where we'd sprinkle them. But imagine if a sensor, the weight of a penny and sewn into uniforms, could transmit broadcast quality images wirelessly."

Title run 2005

More from the NBA playoffs

- Complete playoff coverage
- 2005 NBA Playoffs photos
- Complete playoff schedule
- Playoff statistical leaders
- Game matchups

Today's Top Sports Stories

- Greene's bat powers red-hot Padres - 9:11 PM
- Leagues talk tough as sports commissioners go to Hill - 7:15 PM
- Browns can't say how badly Winslow hurt - 9:35 PM
- Fourth-choice Giacomo draws No. 13 post for Preakness - 8:36 PM
- NBA says players have backtracked; no new talks scheduled - 7:09 PM

• Add USATODAY.com RSS feeds XML

Some Debugging Challenges

- Minimal resource sob story
  - Cannot remotely log on to nodes
- Bugs are hard to track down
- Application behavior changes after deployment
- Extracting debugging information
- Existing fault-tolerance techniques (i.e. rebooting) don’t necessarily apply ; and
- Ensuring system health

After Deploying a Sensor Network…

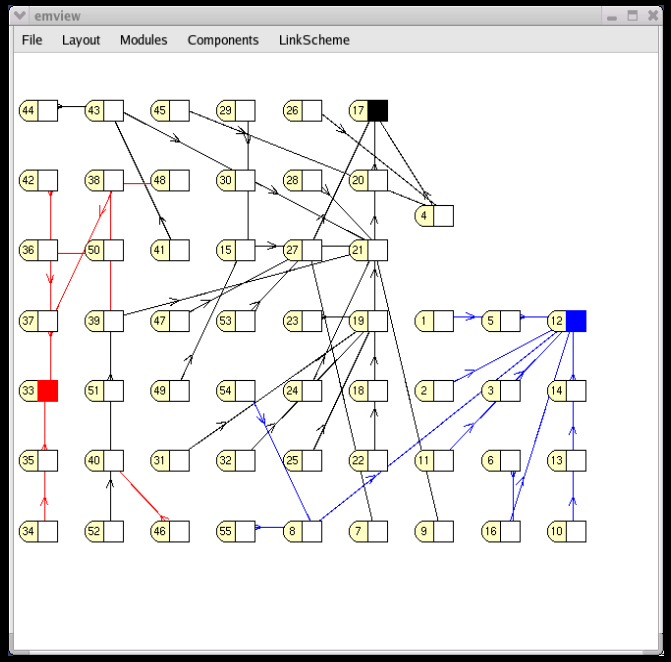
- No data arrives at the sink, could be….

  anything!

- The sink is receiving fluctuating averages from a region – could be caused by
  - Environmental fluctuations
  - Bad sensors
  - Channel drops the data
  - Calculation / algorithmic errors; and
  - Bad nodes

Related Work

- Simulators / Visualizers
  - E.g. EmTOS, EmView, and Tossim
    - Minimal historical context/ event detection
    - Not designed to discern “why” something is happening
- SNMS
  - Interactive health monitoring
- Model-based calibration
- Modeling For System Monitoring



Our Contributions

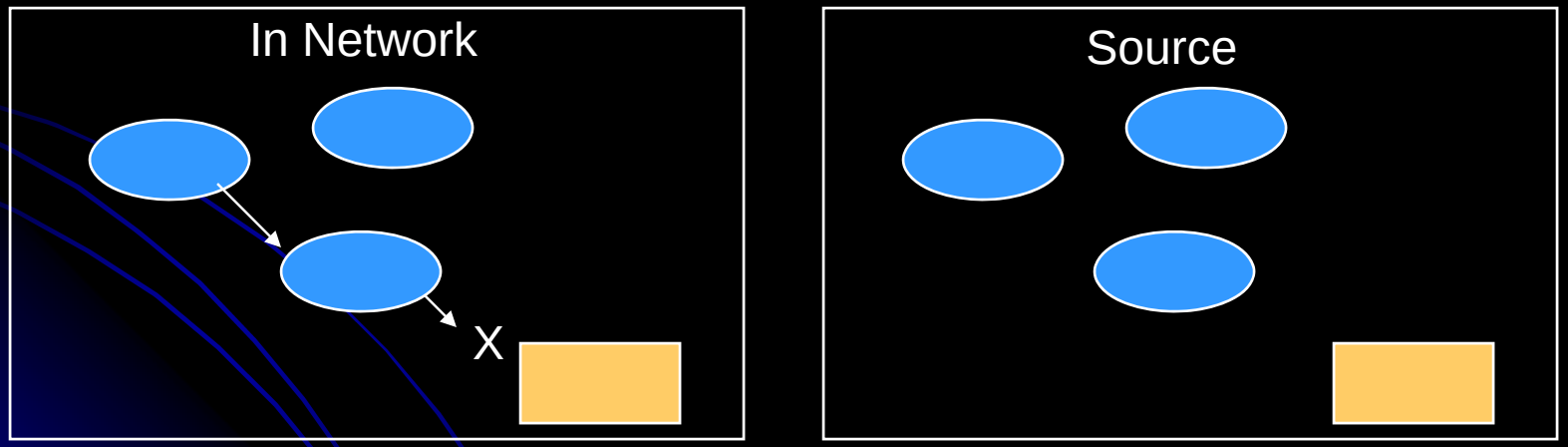
- Working, deployed system that aids in **debugging** by **identifying** and **localizing** failures
  - Debugging – an **iterative** process of detecting and discovering the root-cause of failures
- Low overhead system that runs in pre- or post-deployment environments

Failure Identification

- Application Model
  - Applications that collect data from distributed nodes at a sink
  - “Regular” data exchange required, and interruptions are unexpected
- **Insufficient data => Existence of a problem**
  - “Insufficient data” – defined by components
- Does NOT identify all failures or debug failures to line of code

Failure Localization

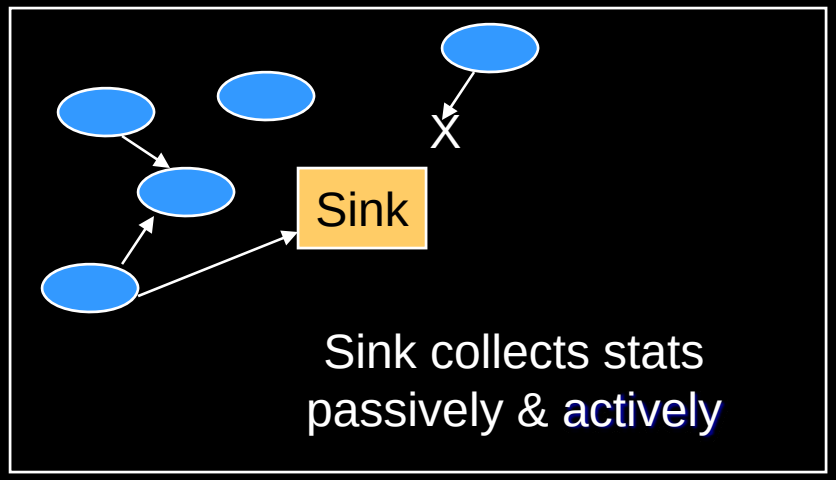
- Determining *why* data is missing
- Physically narrow down cause
  - E.g. Where is the data lost

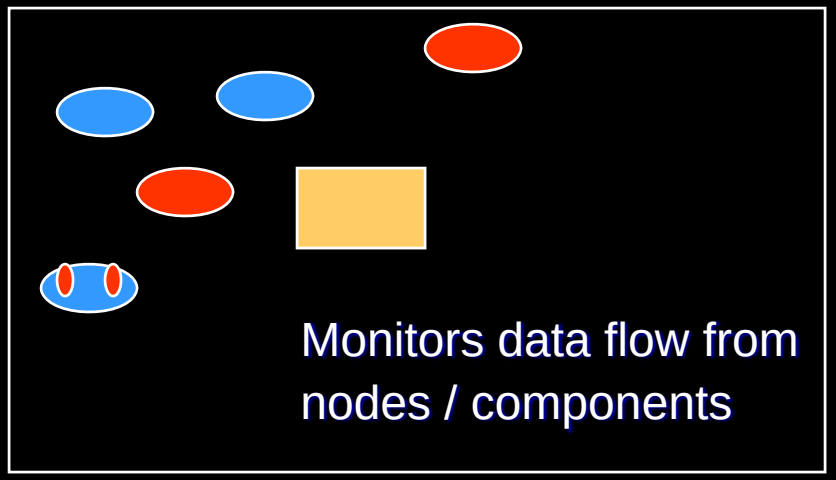


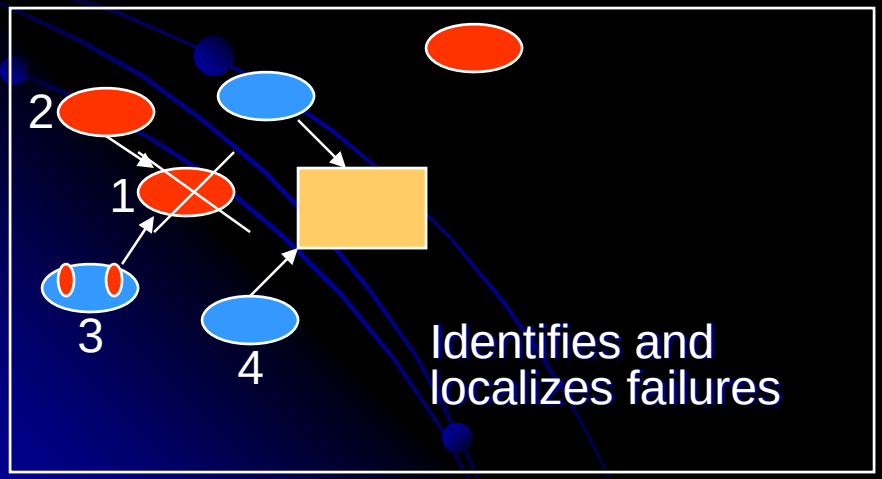
Outline

- **Sympathy's Approach**
- Architecture
- Results

Sympathy Approach



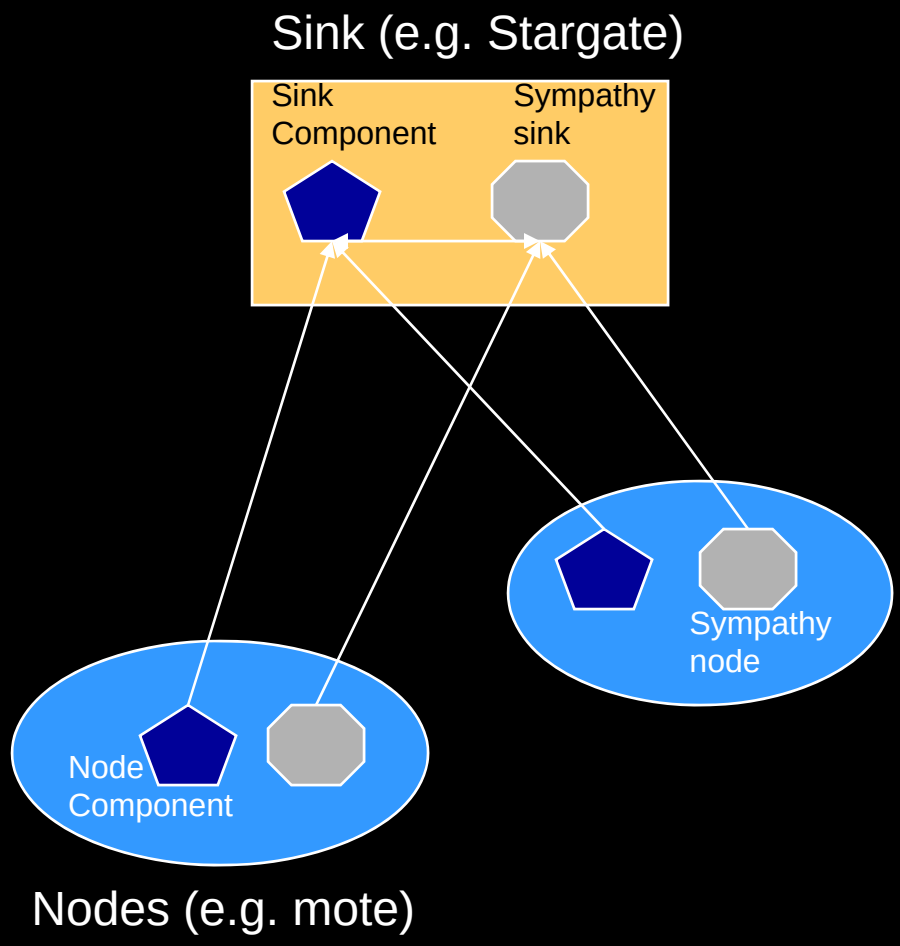




- Highlights failure dependencies and event correlations

Architecture Definitions

- Network: a sink and distributed nodes
- Component
  - Node components
  - Sink components
- Sympathy-sink
  - Communicates with sink components
  - Understands all packet formats sent to the sink
  - Non resource constrained node
- Sympathy-node
- Statistics period
- Epoch



Node Statistics

- Passive (in sink's broadcast domain) and actively transmitted by nodes

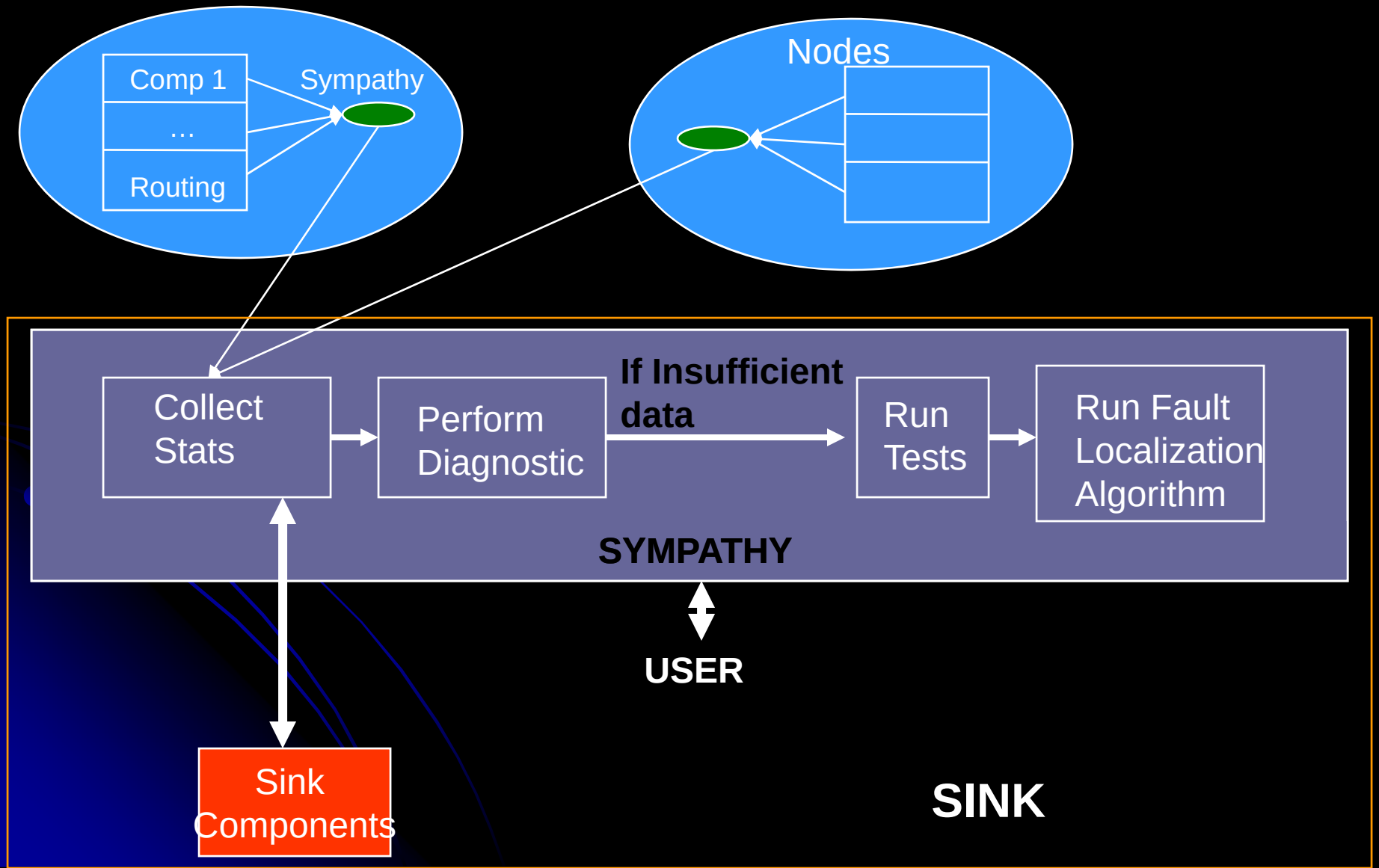
Statistic Name	Description
Routing Table	(Sink, next hop, quality) tuples.
Neighbor Lists	Neighbors and associated ingress/ egress
Time awake	Time node is awake
#Statistics tx	Number of statistics packets transmitted to the sink
#pkts routed	Number of packets routed by the node

Component Statistics

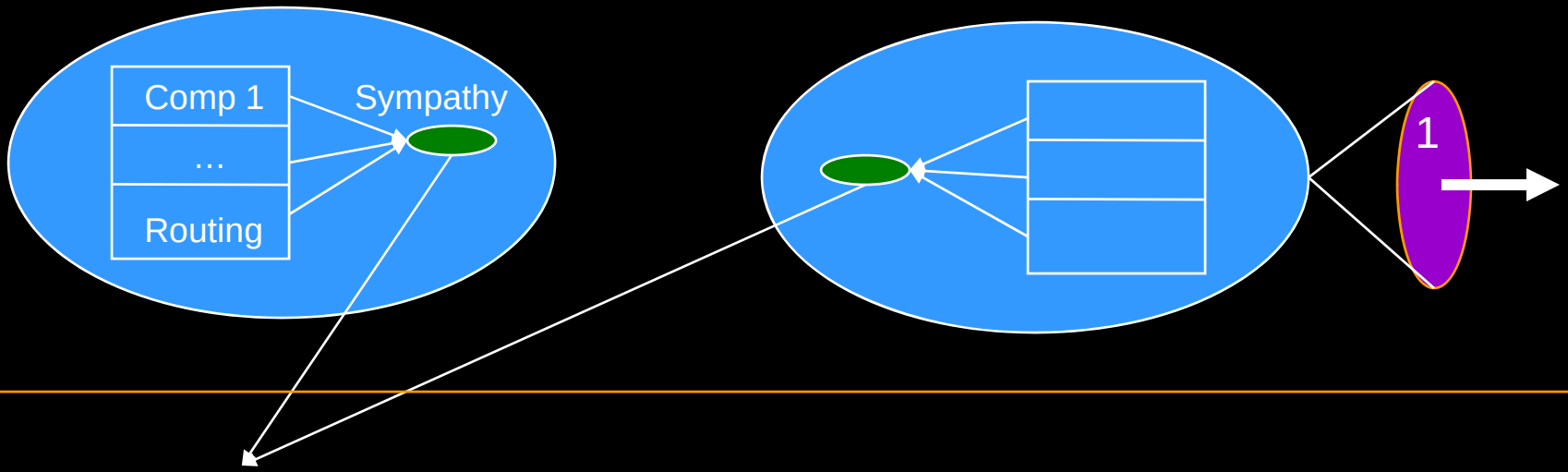
- Actively transmitted by a node to the sink, for each instrumented component

Statistic Name	Description
#Reqs comp rx	Number of packets component received from sink
#Pkts tx	Number of packets component transmitted to sink
Last timestamp	Timestamp of last data stored by component

Sympathy System



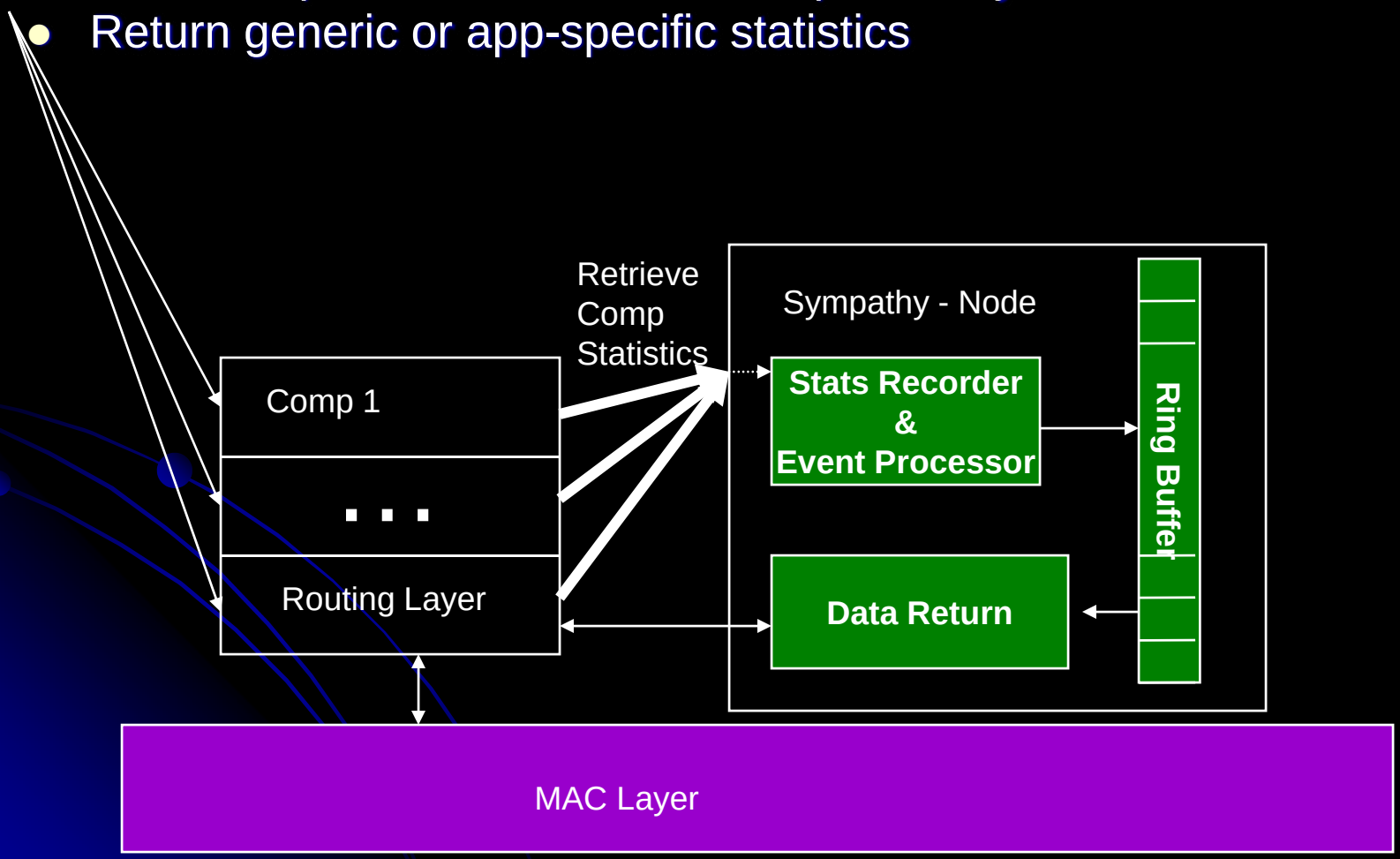
Sympathy System



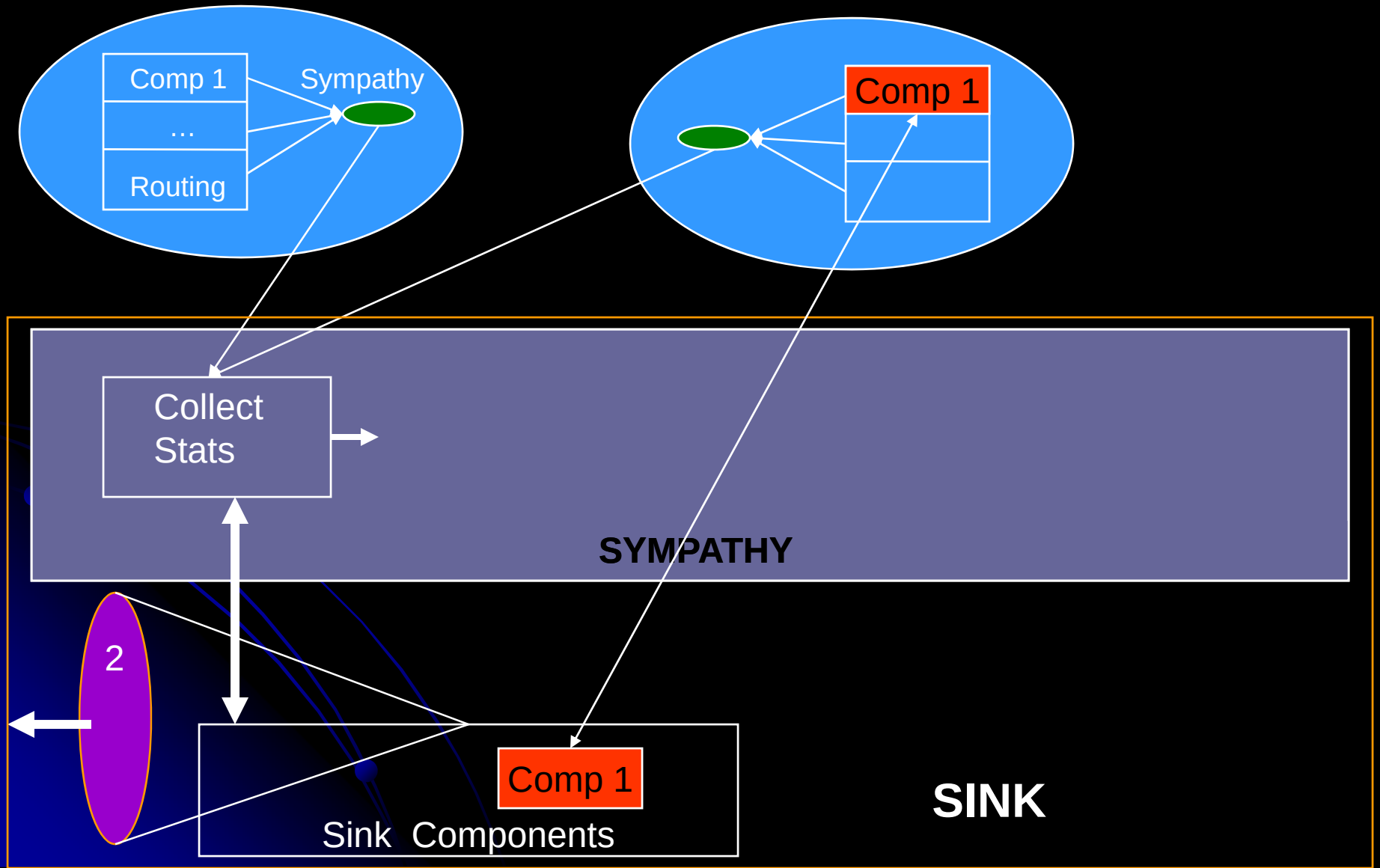
SINK

Network Node

- Each component is monitored independently
- Return generic or app-specific statistics

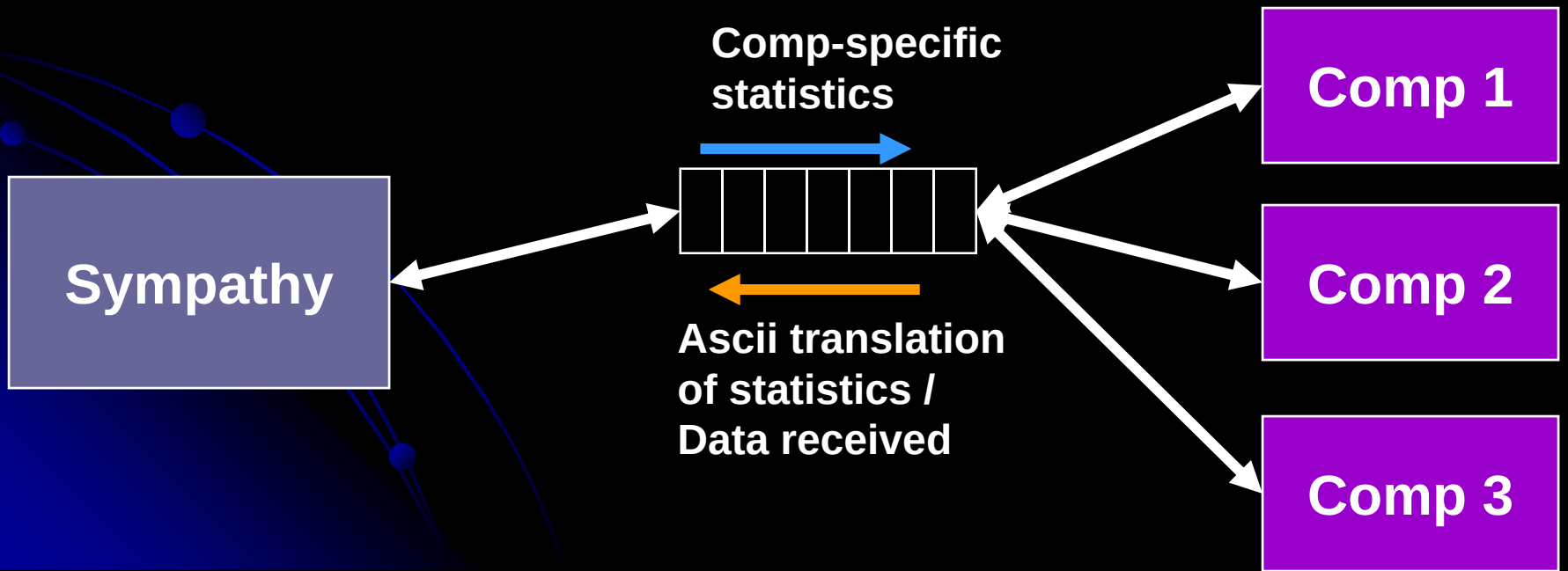


Sympathy System

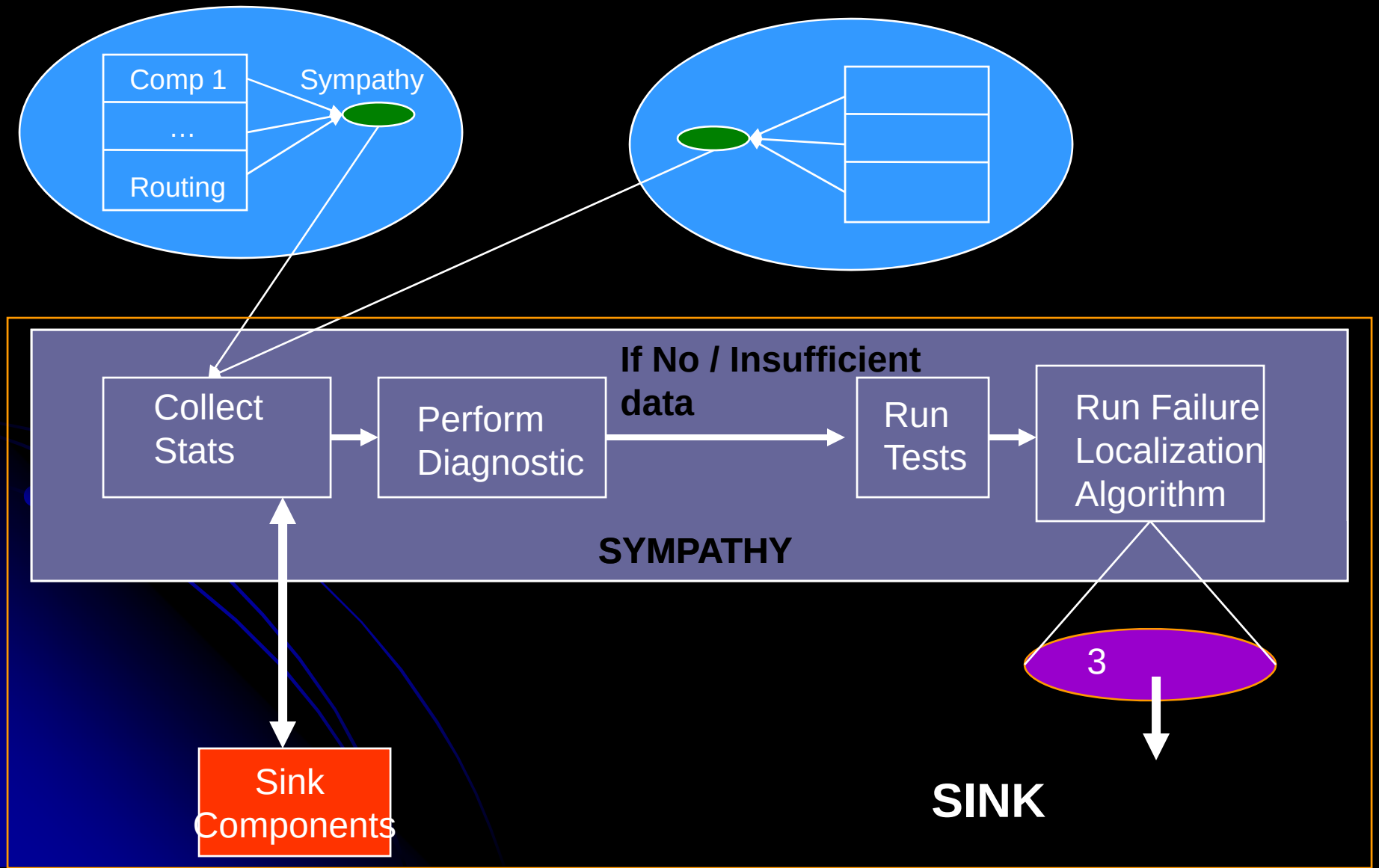


Sink Interface

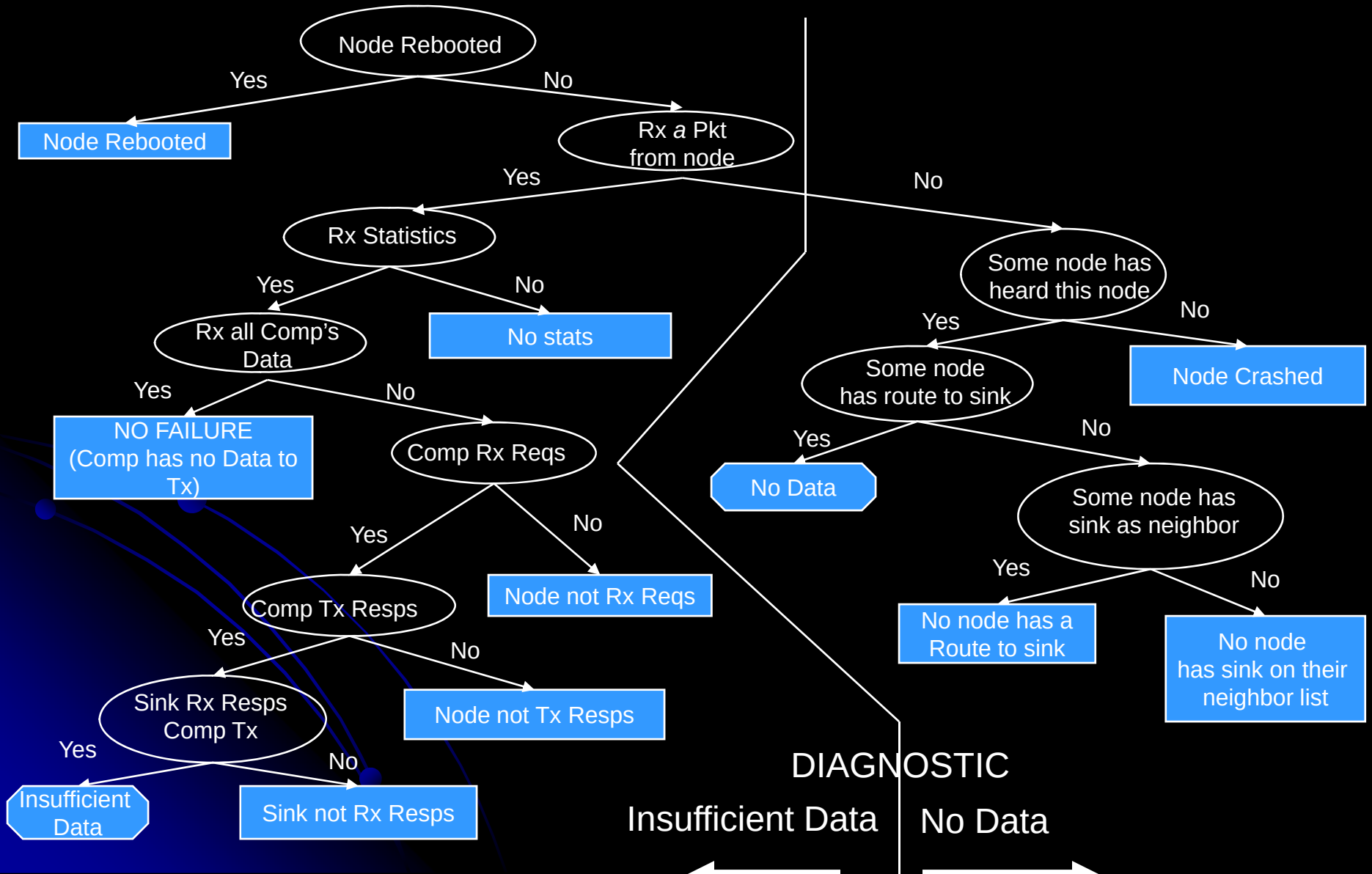
- Sympathy passes comp-specific statistics using a packet queue
- Components return ascii translations for Sympathy to print to the log file



Sympathy System



Failure Localization Algorithm



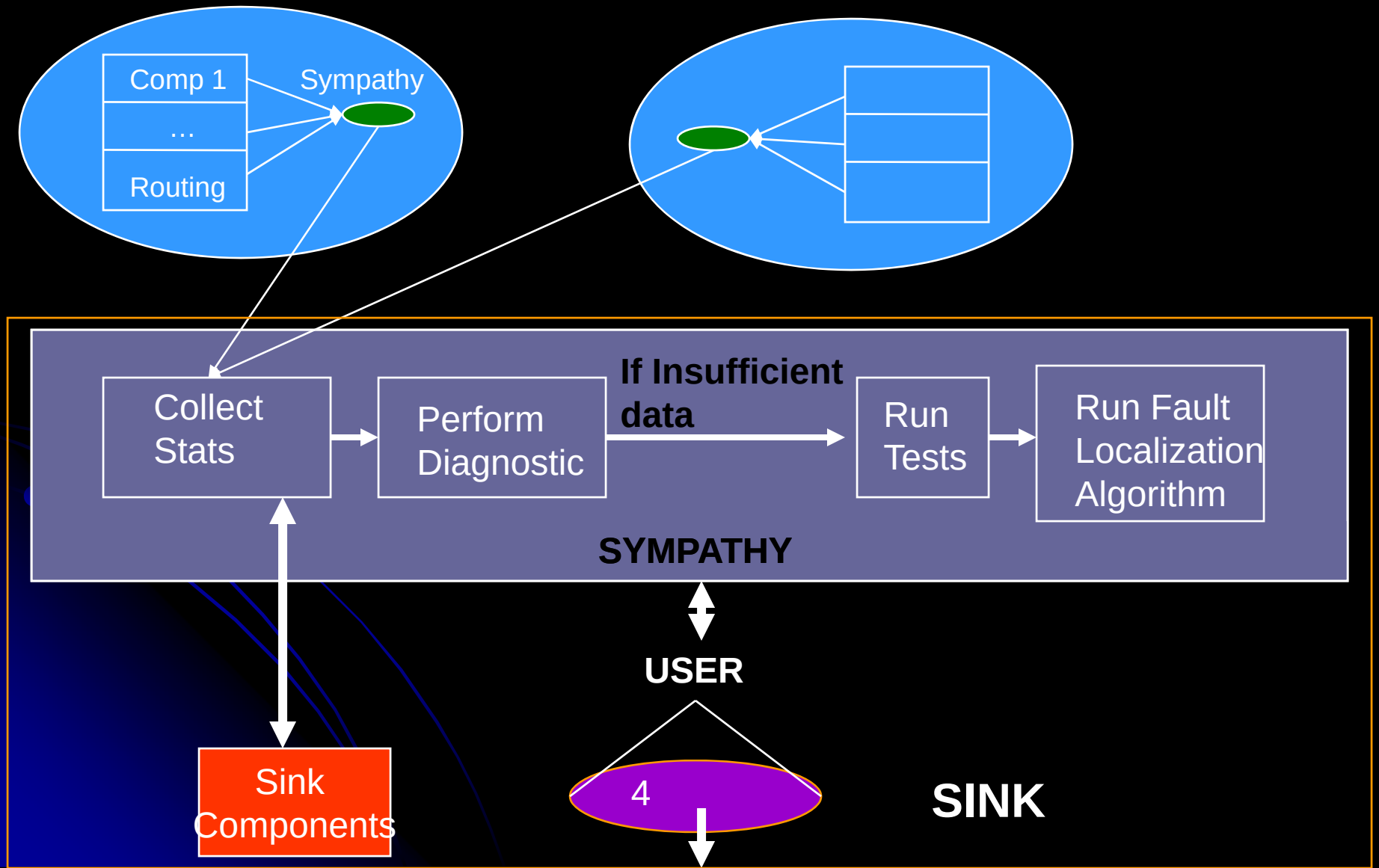
Functional “No Data” Failure Localization

Failure	Description
Node Crash	Node has crashed and not come back
No Route to Sink	No valid route exists to the sink from a node
No Data	No data received from a node, and Sympathy cannot localize the failure

Performance "Insufficient Data" Failure Localization

Failure	Description
Node Reboot	Node has rebooted
Congestion	Correlated failures on packet reception
No reqs rx	Component is not receiving requests from sink
No rsps tx	Component is not transmitting data in response to requests
No rsps rx	Sink is not receiving data transmitted by a component
No stats rx	Sink has not received Sympathy statistics on the component

Sympathy System



Informational Log File

Node 25, Time: Node awake(mins): 78 Sink awake: 78(mins)
Route: 25 -> 18 -> 15 -> 12 -> 10 -> 8 -> 6 -> 2
node 27, are children
Num neighbors heard this node: 6

Pkt-type	#Rx	Mins-since-last	#Rx-errors	Mins-since-last
1:Beacon	15(2)	0 mins	1(0)	52 mins
3:Route	3(0)	37 mins	0(0)	INF
Symp-stats	12(2)	1 mins		

Reported Stats from Components

**Sympathy:
#metrics tx/#stats tx/#metrics expected/#pkts routed: 13(2)/12(2)/13(1)/0(0)

Node-ID	Egress	Ingress
8	128	71
13	128	121
24	249	254

Failure Log File

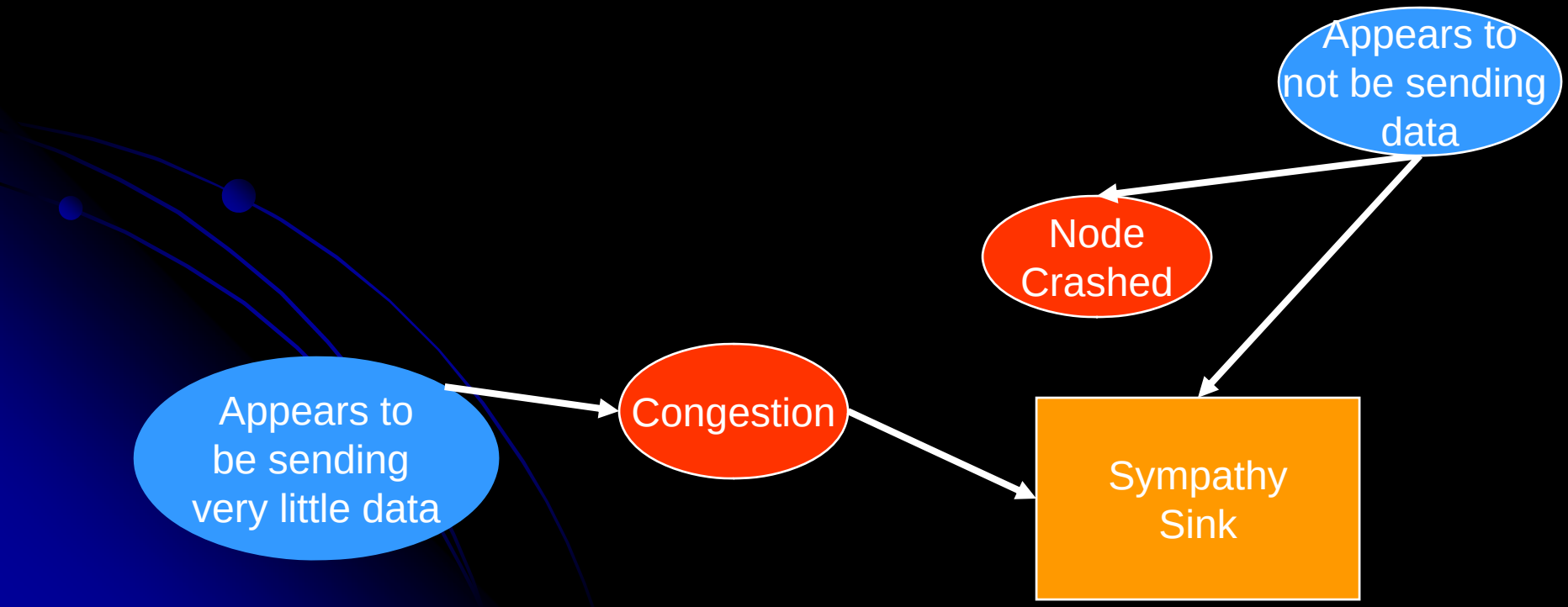
Node 18, Time: Node awake(mins): 0 Sink awake: 3(mins)
Node Failure Category: Node Failed!

TESTS
- Received stats from module [FAILED]
- Received data this period [FAILED]
- Node thinks it is transmitting data [FAILED]
- Node has been claimed by other nodes as a neighbor [FAILED]
- Sink has heard some packets from node [FAILED]
  - Received data this period: Num pkts rx: 0(0)
  - Received stats from module: Num pkts rx: 0(0)

Node's next-hop has no failures

Spurious Failures

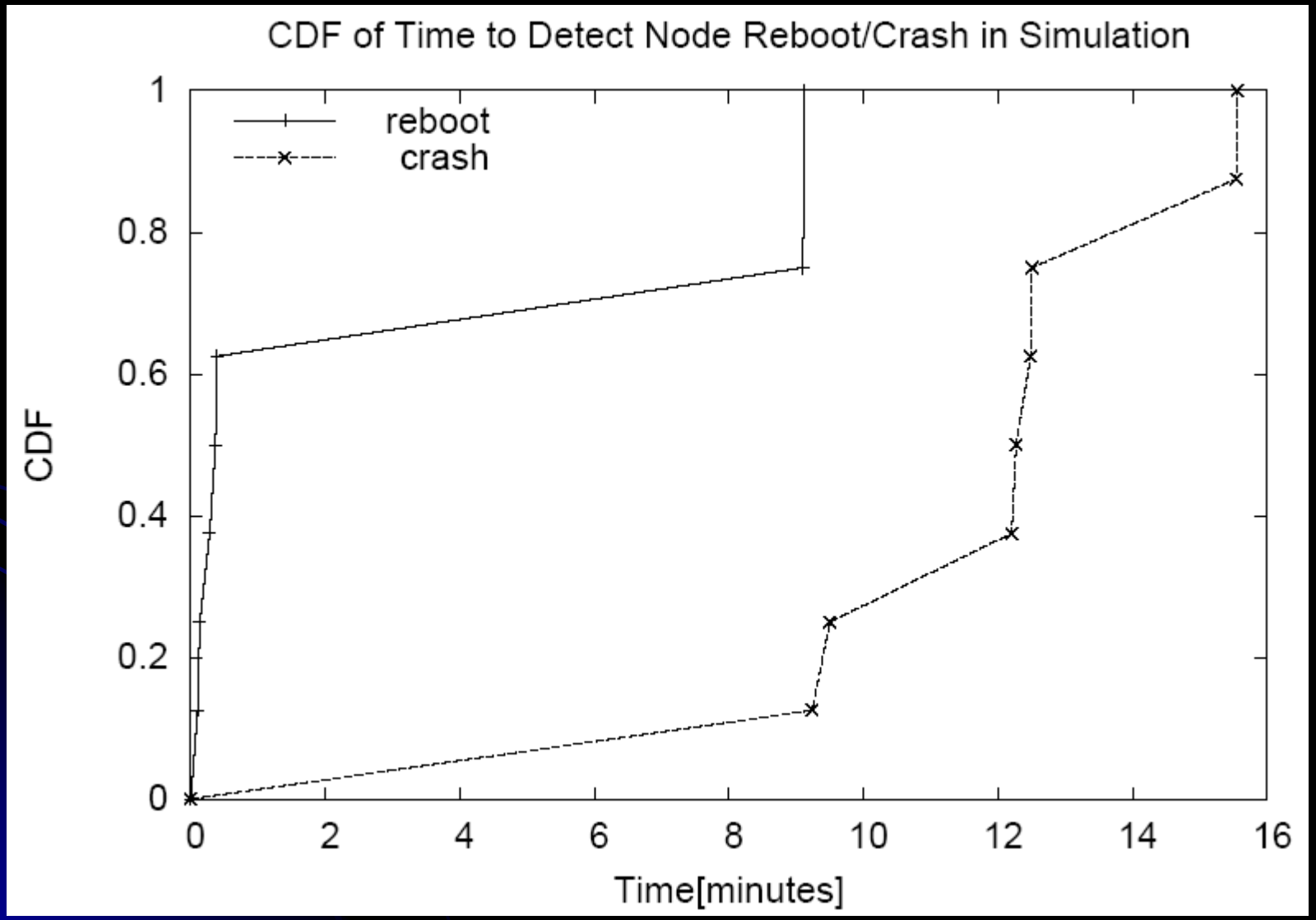
- An artifact of another failure
- Sympathy highlights failure dependencies in order to distinguish spurious failures



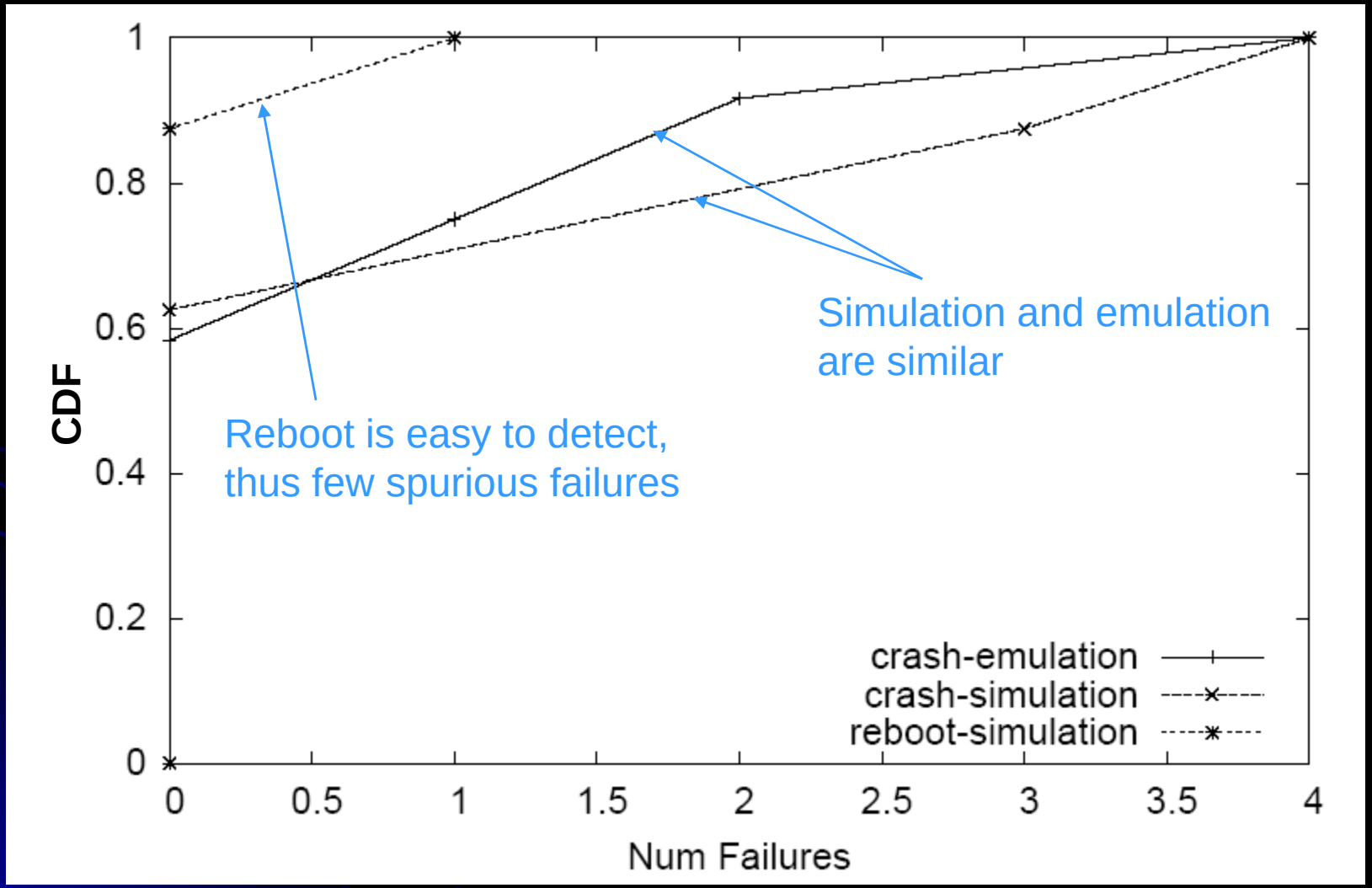
Testing Methodology

- Application
  - Run in Sympathy with ESS
  - In simulation, emulation and deployment
- Traffic conditions: no traffic, application traffic, congestion
- Node failures
  - Node reboot – only requires information from the node
  - Node crash – requires spatial information from neighboring nodes to diagnose
- Failure injected in one node per run, for each node
- 18 node network, with maximum 7 hops to the sink

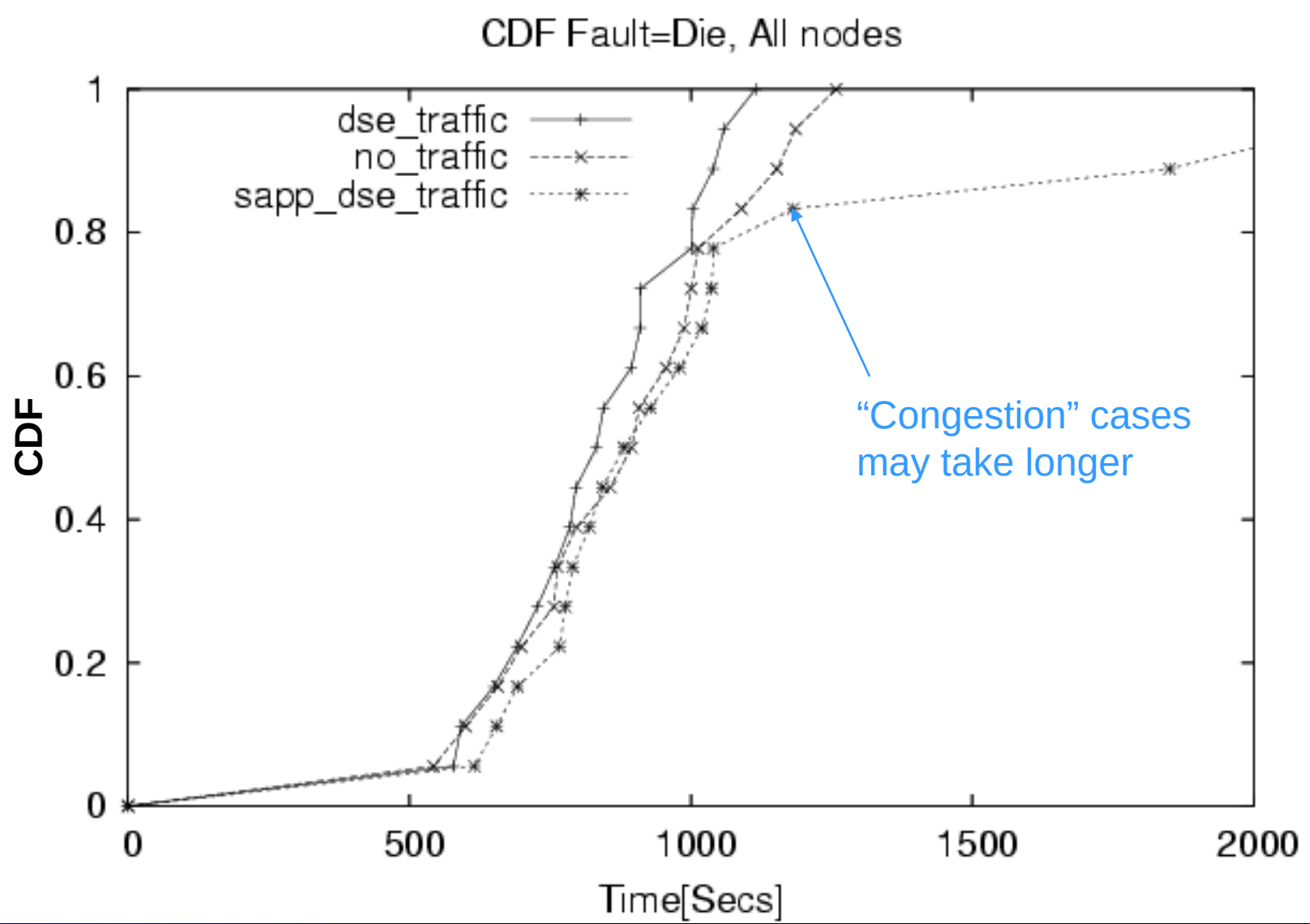
Time to Detect Node Crash/Reboot



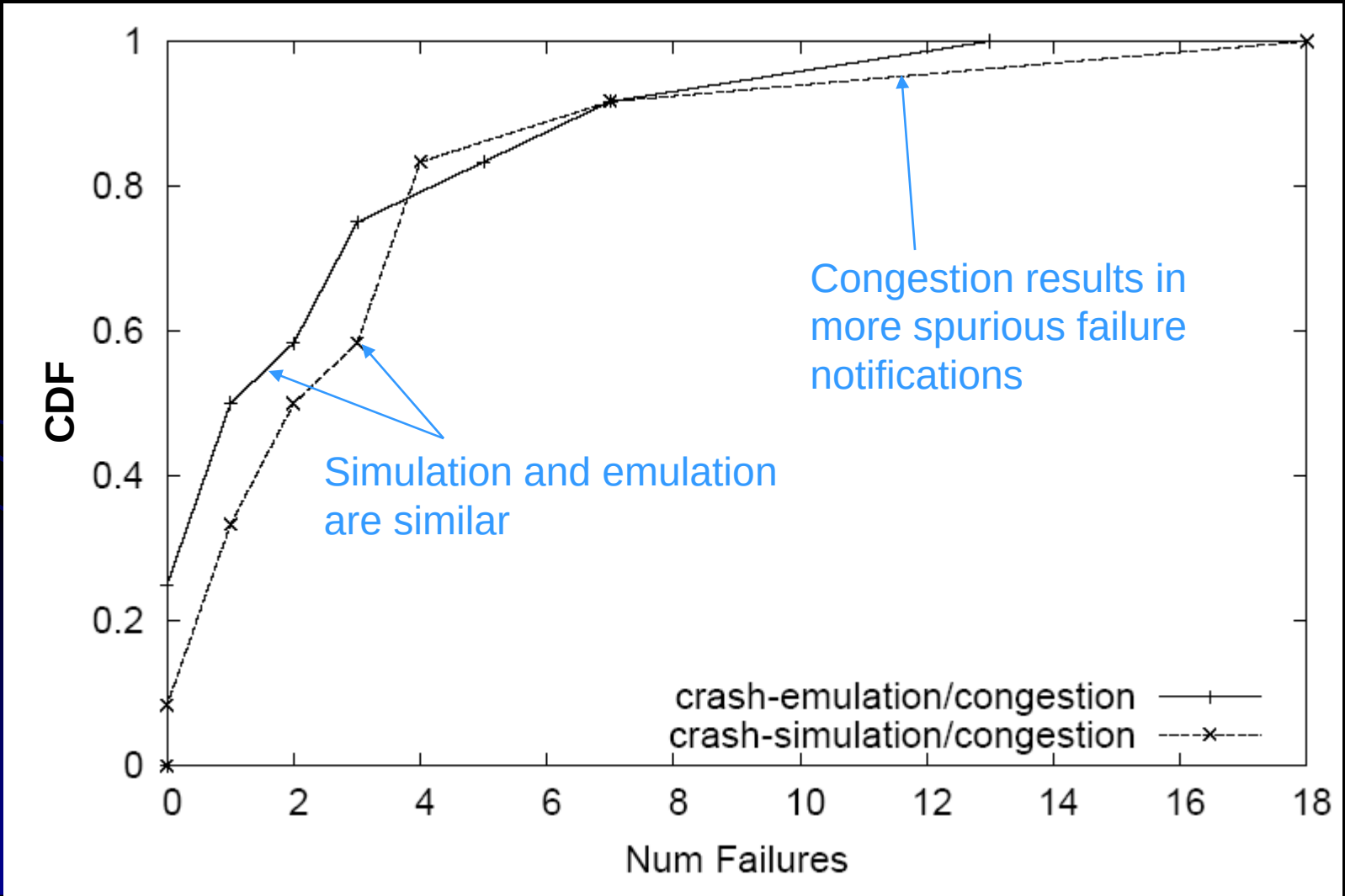
Spurious Failure Notifications



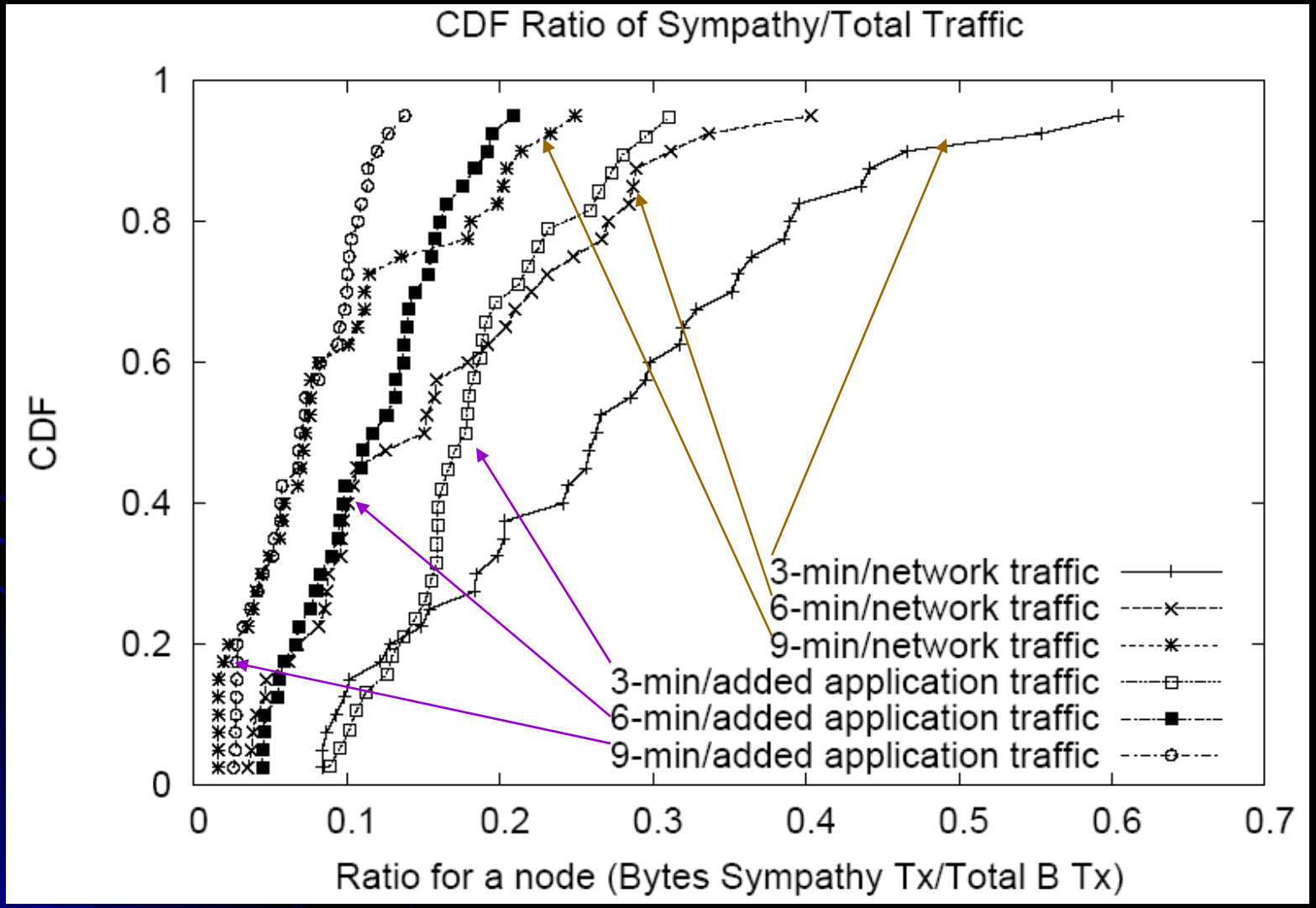
Time to Detect Node Crash



Spurious Failure Notifications w/ Congestion

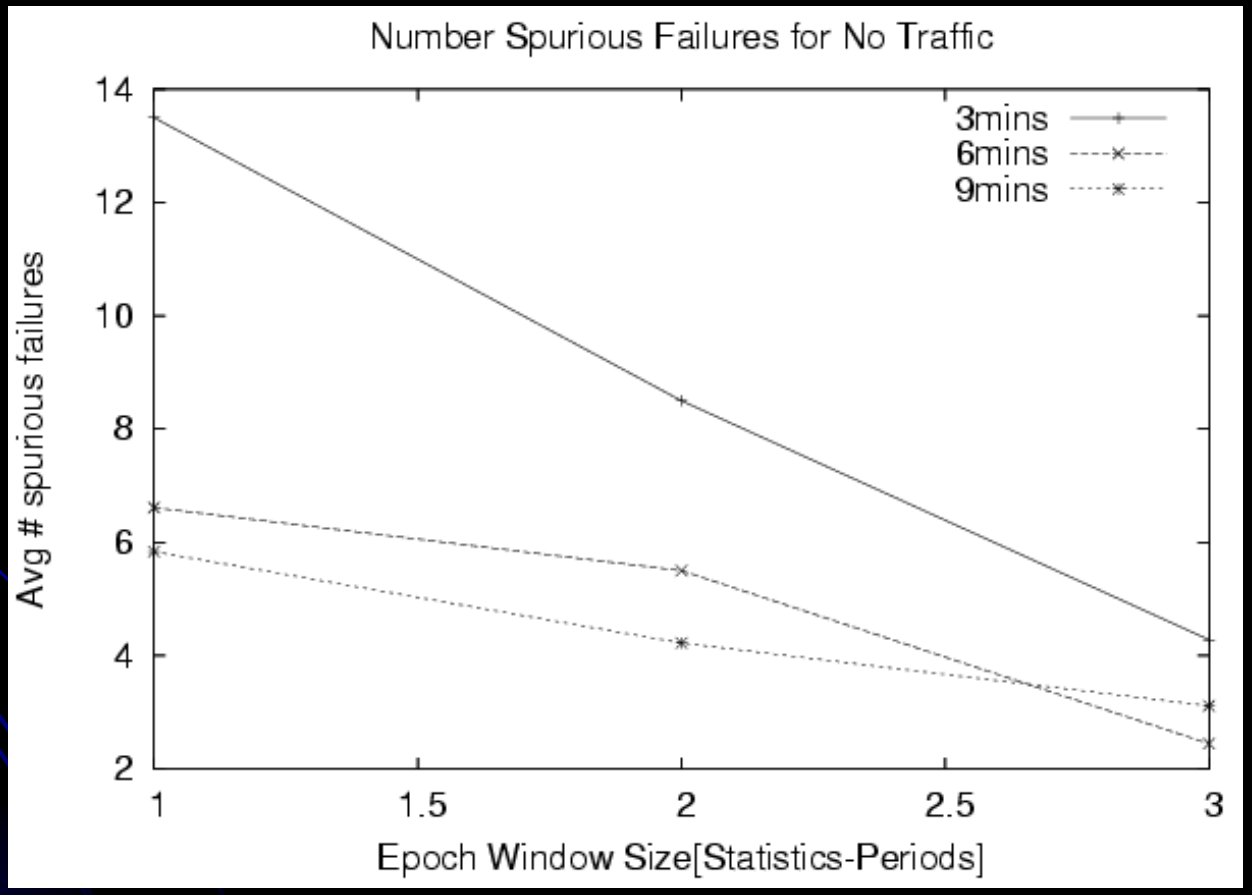


Sympathy Packet Overhead



Varying Epoch Window Size, No Traffic

- Window size: Number of statistics periods in the epoch



Memory Footprint

Binary	RAM	ROM
ESS w/o Sympathy	3089 B	96094 B
ESS w/ Sympathy	3160 B	104802 B
Difference	71 B	8708 B

Another Real World Example

- Temporal sink presence

Ongoing Work

- Using a Bayes engine to reduce the number of spurious failure notifications
- More deployments

Conclusion

- A deployed system that aids in debugging by detecting and localizing failures
- Small list of statistics that are effective in localizing failures
- Behavioral model for a certain application class that provides a simple diagnostic to measure system health

Thank You!

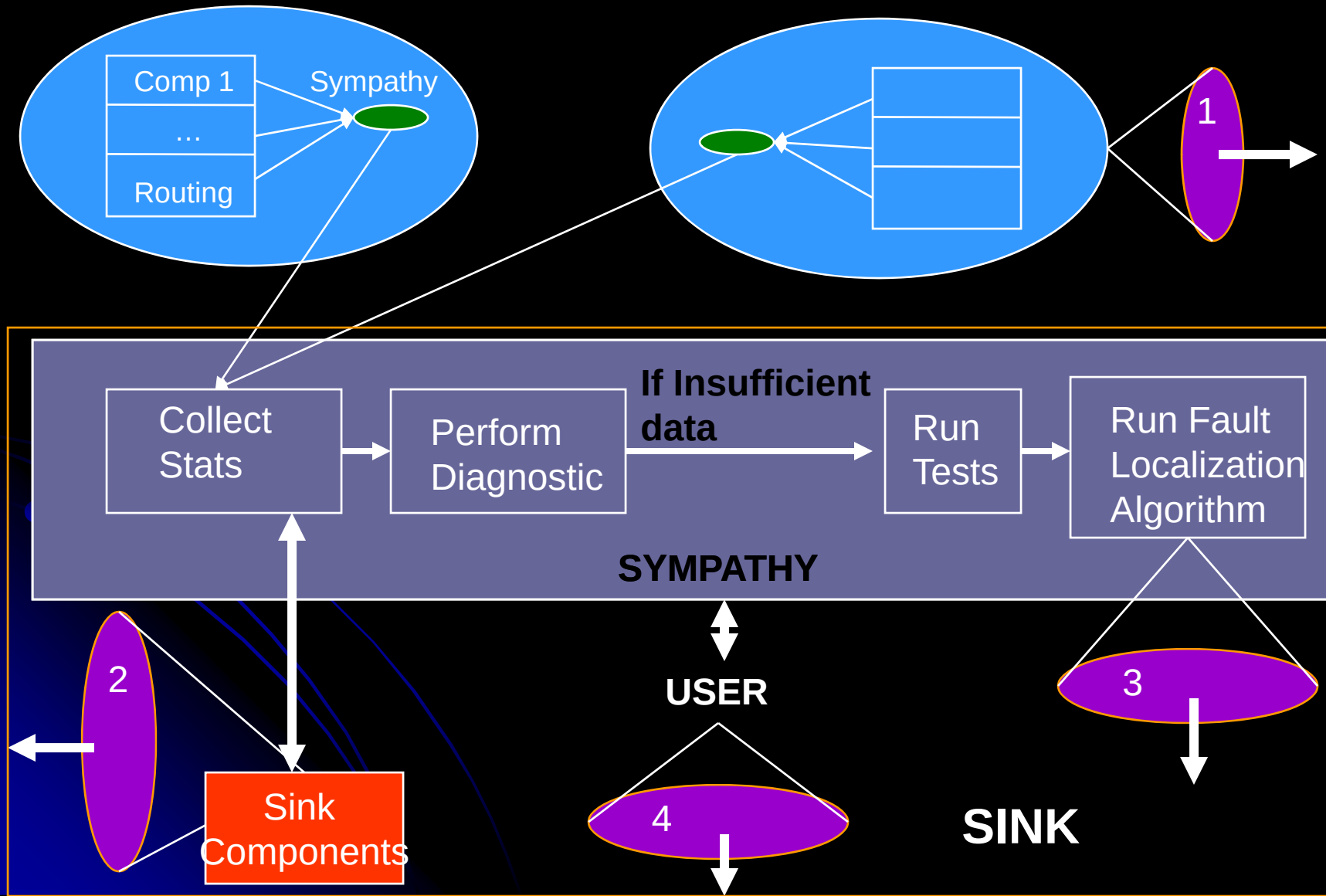
Iter_fail Variable

- For some failures, Sympathy must get information from all nodes within the epoch

OR

- Sympathy should not have heard from that node for iter_fail statistics periods in order to ignore the node

Sympathy System

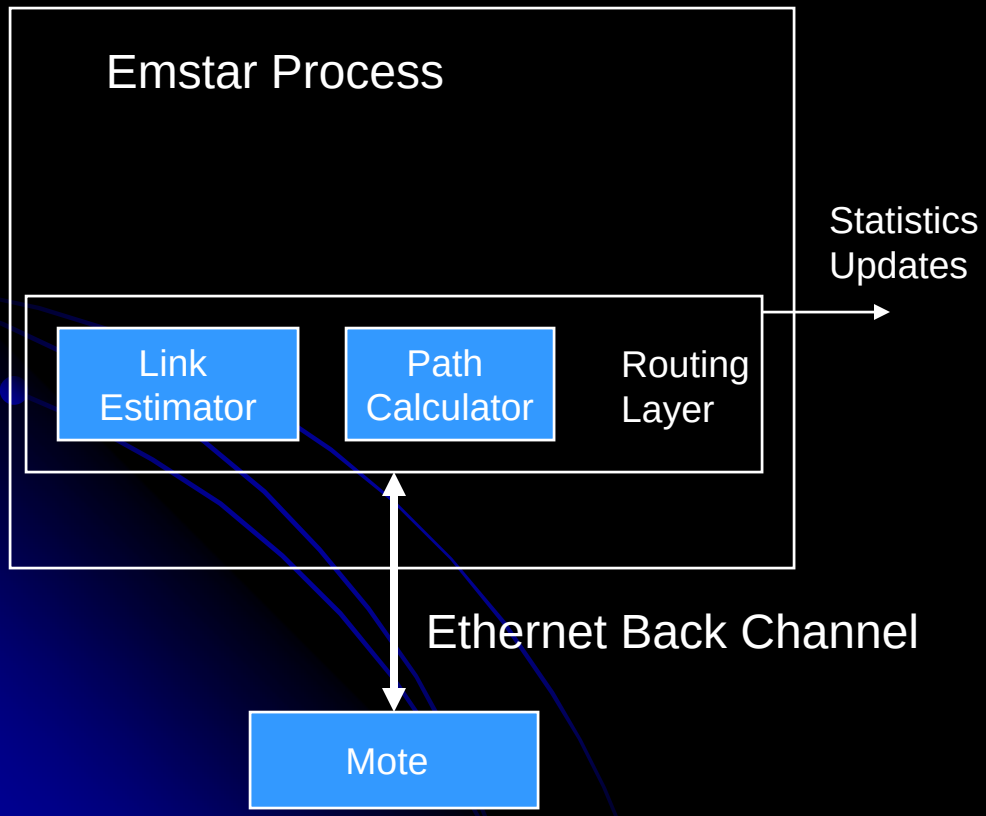


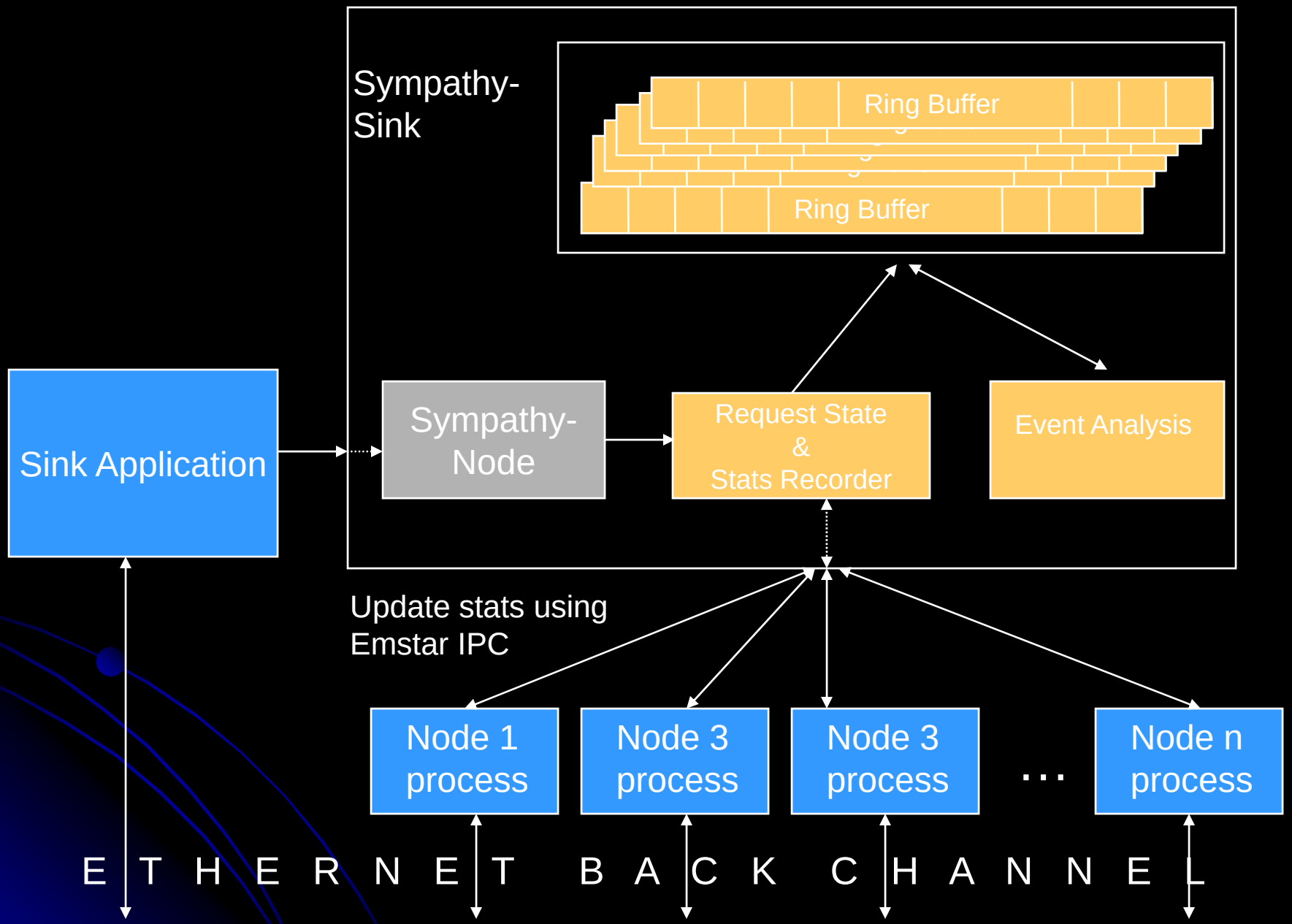
Failures Sympathy Detects[1,2]

- System Design / algorithm / protocol bugs
- Connectivity / topology

[1] R. Szewczyk, J. Polastre, A. Mainwaring, D. Culler “Lessons from a Sensor Network Expedition”. In EWSN, 2004

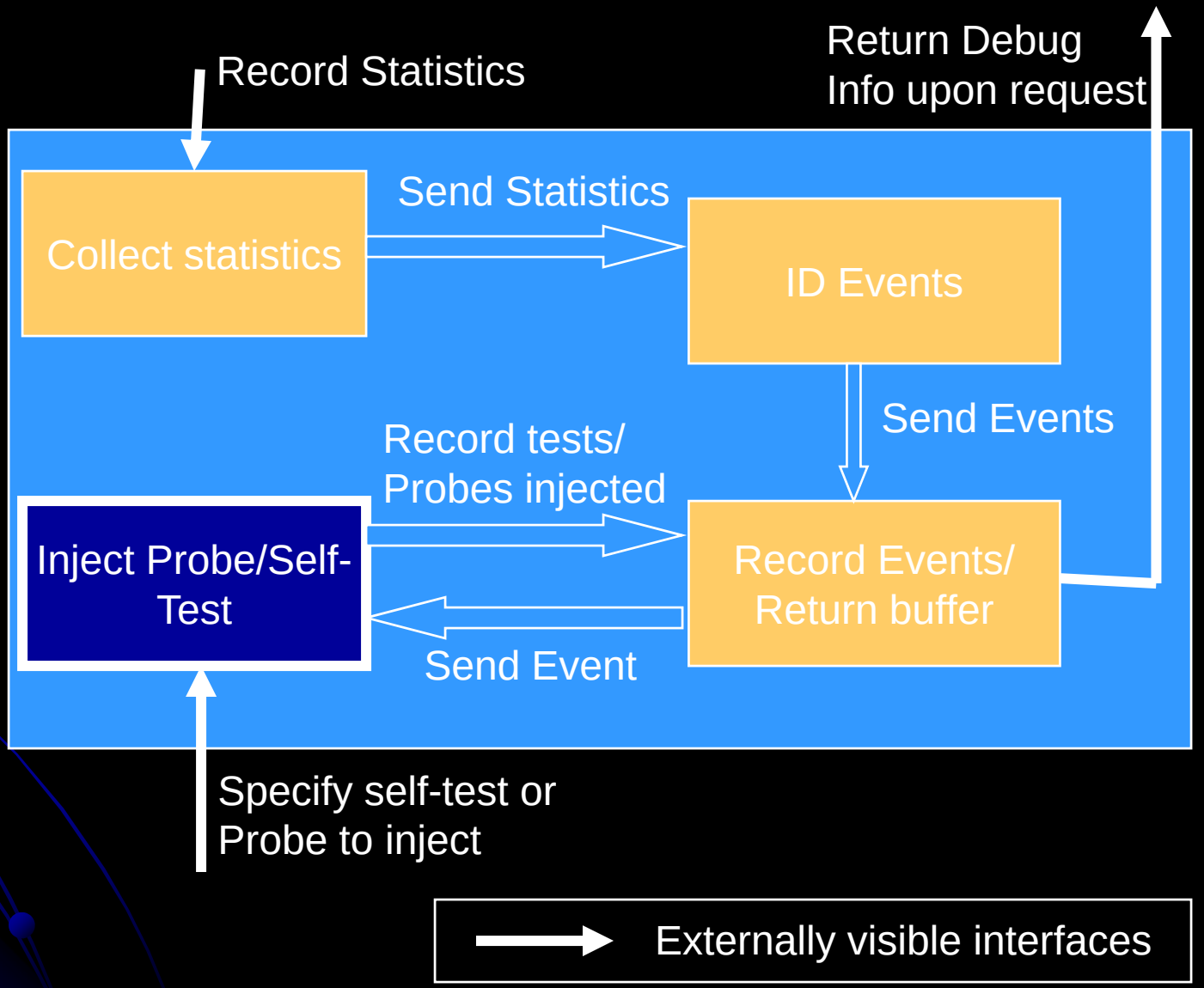
2 A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler “Wireless Sensor Networks for Habitat Monitoring”. In ACM International Workshop on Wireless Sensor Networks and Applications.


Emstar Process
Statistics
Updates
Link
Estimator
Path
Calculator
Routing
Layer
Ethernet Back Channel
Mote


Sympathy-
Sink
Ring Buffer
Ring Buffer
Sink Application
Sympathy-
Node
Request State
&
Stats Recorder
Event Analysis
Update stats using
Emstar IPC
Node 1
process
Node 3
process
Node 3
process
...
Node n
process
ETHERNET BACK CHANNEL

Regular Sympathy Peon

- Self-tests and probes can also be externally specified (e.g. by a neighbor)



SNMS/ Nucleus Management System[1]

- Enables interactive health monitoring of WSN in the field
- 3 Pieces
  - Parallel dissemination and collection
  - Query system for exported attributes
  - Logging system for asynchronous events
- Small footprint / low overhead
  - Introduces overhead only with human querying

[1] Gilman Tolle, David Culler, “Design of an Application-Cooperative Management System for WSN” Second EWSN, Istanbul, Turkey, January 31 - February 2, 2005

Model-Based Calibration[1,2]

- Use models of the physical environment to identify faulty sensors, e.g.:
  - Assume values from neighboring sensors in a dense deployment should be “similar”[2]
  - Plug sensor data into a pre-defined physical model; identify sensors that make the model inconsistent[1]

[1] Jessica Feng, S. Megerian, M. Potkonjak “Model-based calibration for Sensor Networks”. IEEE International Conference on Sensors, Oct 2003

[2] A Collaborative Approach to In-Place Sensor Calibration – Vladimir Bychovskiy Seapahn Megerian et al

Modeling For System Monitoring[1,2,3]

- Identify “anomalous” behavior based on externally observed statistics
  - Statistical analysis and Bayesian networks used to identify faults

[1] E. Kiciman, A. Fox “Detecting application-level failures in component-based internet services”. In IEEE Transactions on Neural Networks, Spring 2004

[2] A. Fox, E. Kiciman, D. Patterson, M. Jordan, R. Katz. “Combining statistical monitoring and predictable recovery for self-management”. In Procs. Of Workshop on Self-Managed Systems, Oct 2004

[3] E. Kiciman, L Subramanian. “Root cause localization in large scale systems”

Sympathy Sink