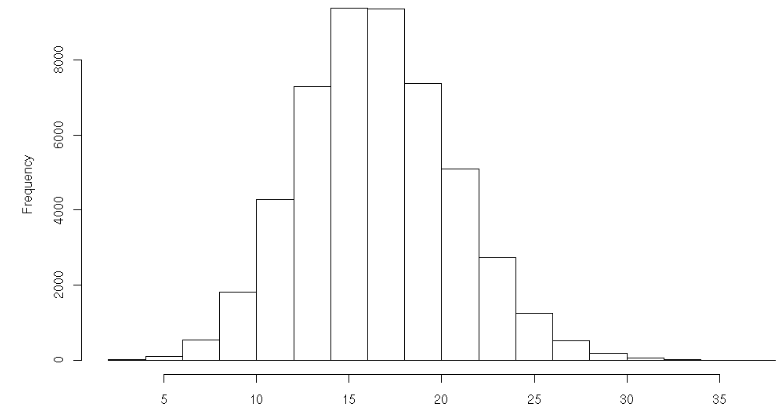
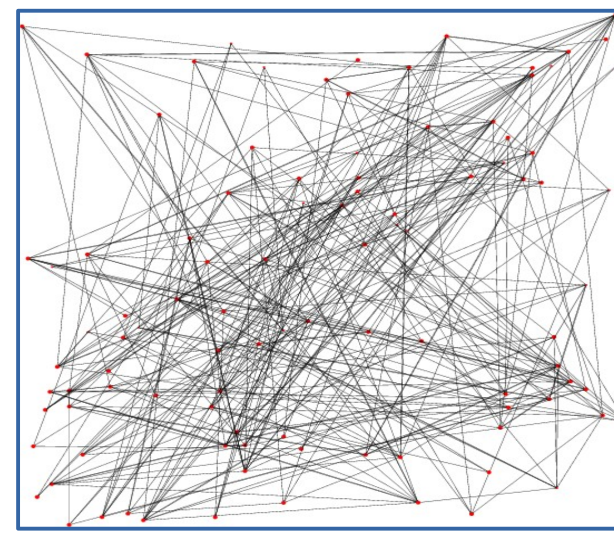
Load-balancing and redundancy

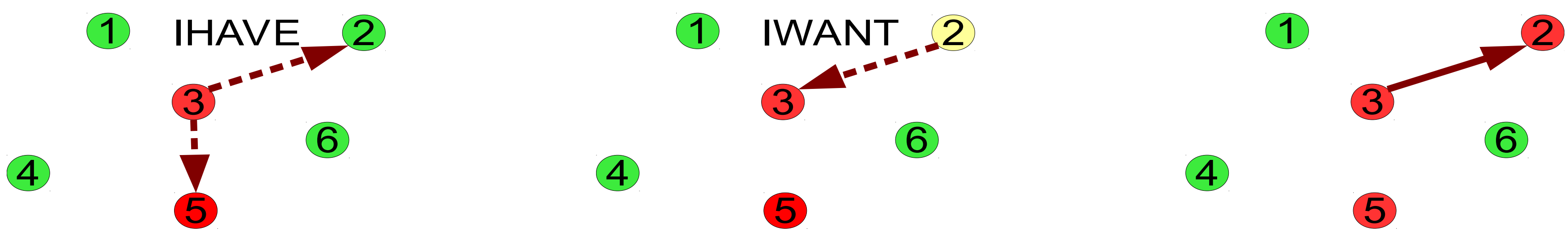
- Distribution of traffic on nodes and links:
  - Normal
  - Mean = fanout
- *f-1* useless payload copies for each destination!
- Not using nodes and links with higher capacity
- Using expensive links as much as cheap ones





Lazy dissemination

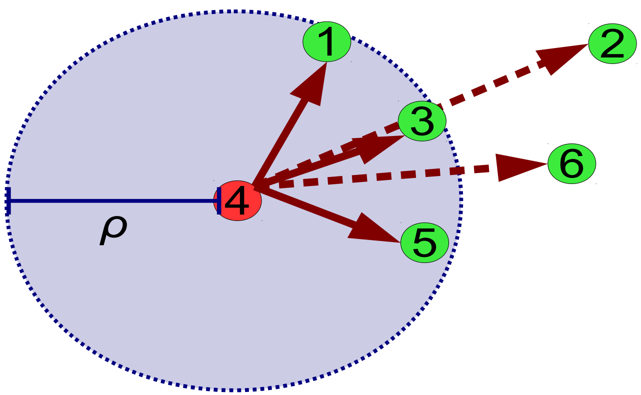
- Defer sending the payload (lazy transmission)
  - Equivalent to eager transmission (minor increase in probability of failure)
  - Saves bandwidth, assuming that the payload is much larger that a control message with the event id

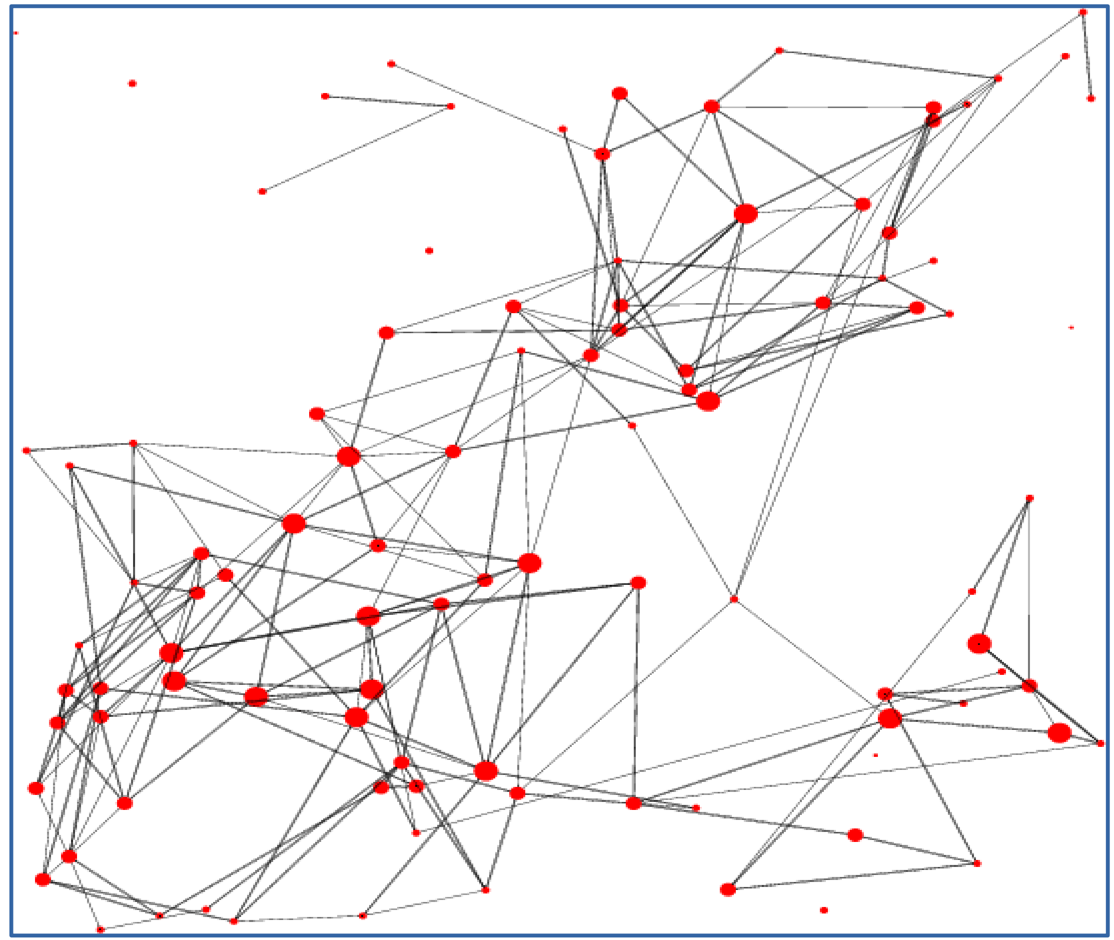


Emergent structure

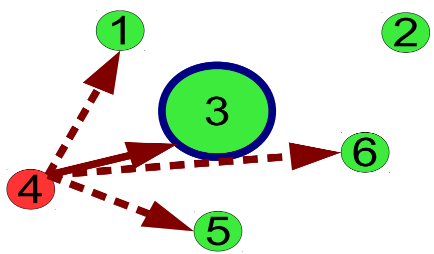
- Manipulate eager vs. lazy transmission such that the protocol achieves efficiency goals with high probability
  - Using cheap network links
  - Using large capacity nodes
  - Reduces redundant payload copies
- Decision based on local information:
  - Stateless: Decisions are independent
  - Stateful: Decision based on previous observations

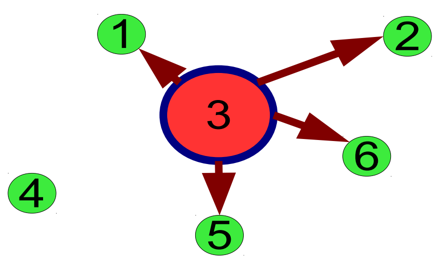
Radius (aka Mesh)

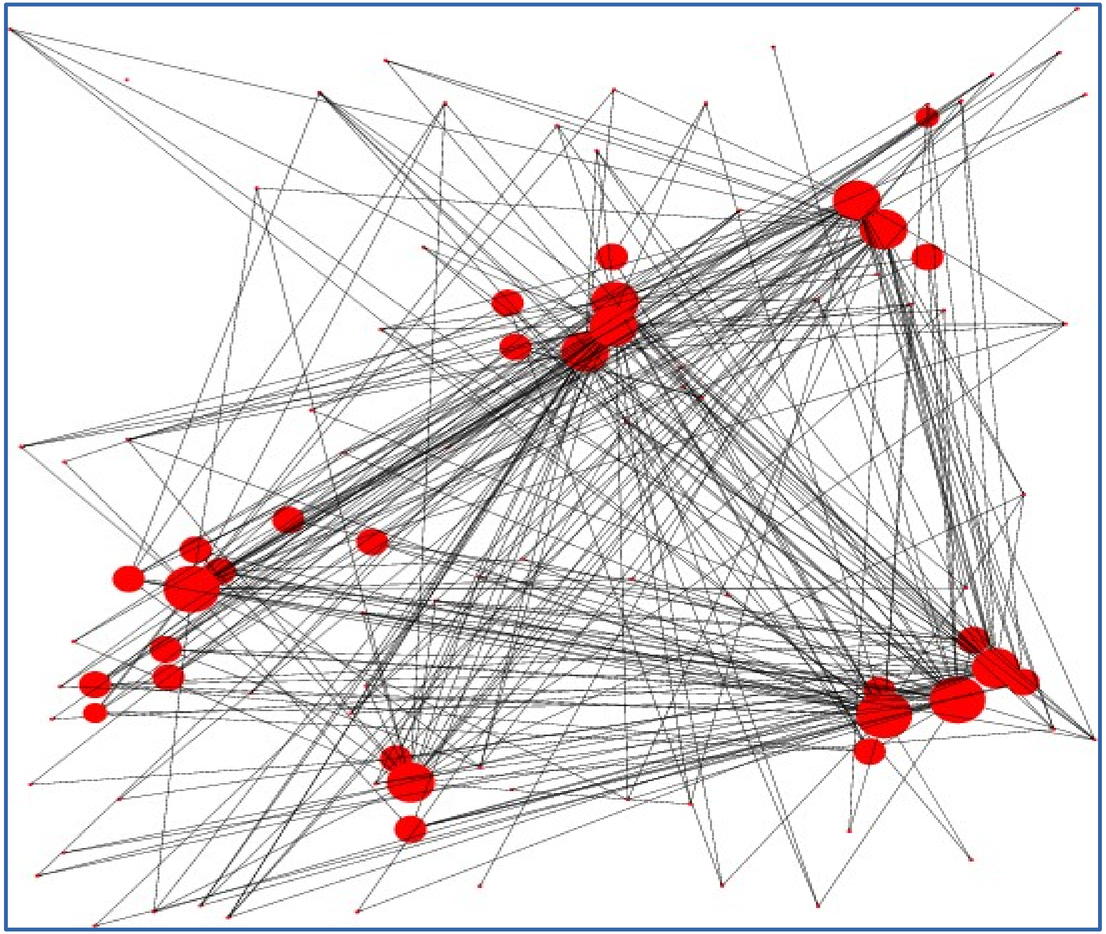




Ranked (aka Supernode)

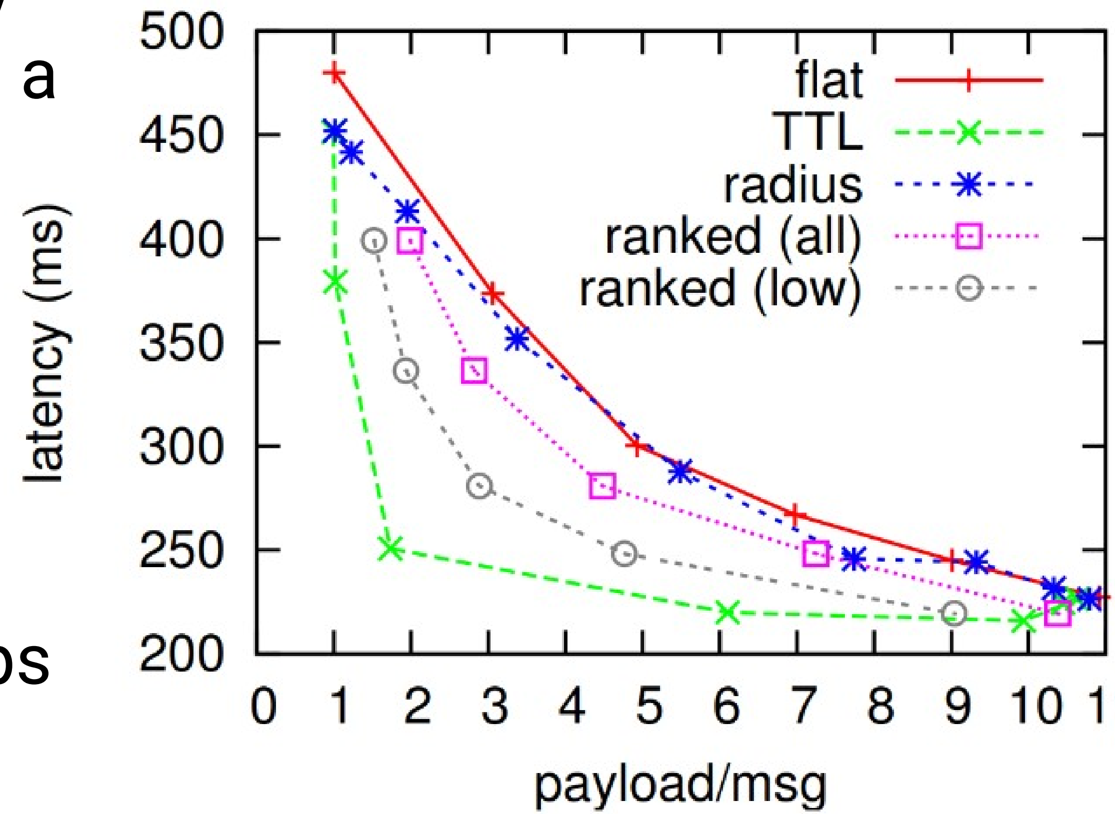






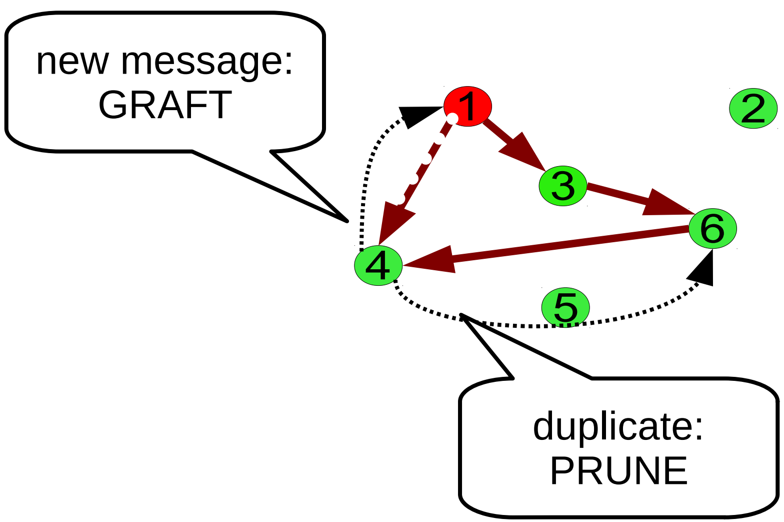
Time-to-live

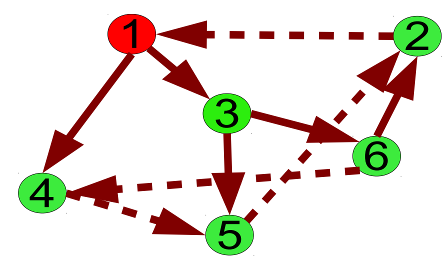
- Observation: Probability of a transmission being a duplicate is:
  - 0 in first round
  - minimal in 2nd round
  - …
  - 1 in latest round
- Use eager for first k hops achieves <2 payloads with good latency



Push-Lazy-Push Tree (aka “plumtree”)

- Start by using eager transmission:
  - When receiving duplicates, ask sender to PRUNE
  - When receiving annoucements for unknown messages, as sender to GRAFT





References

- N. Carvalho, J. Pereira, R. Oliveira, and L. Rodrigues, “**Emergent Structure in Unstructured Epidemic Multicast**,” in 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN’07), Jun. 2007, pp. 481–490. http://dx.doi.org/10.1109/DSN.2007.40
- J. Leitao, J. Pereira, and L. Rodrigues, “**Epidemic Broadcast Trees**,” in 2007 26th IEEE International Symposium on Reliable Distributed Systems (SRDS 2007), Oct. 2007, pp. 301–310. http://dx.doi.org/10.1109/SRDS.2007.27

Distributed aggregation

- Computation of aggregate functions:
  - Max, min
  - Count, sum
  - Average (reduces to sum / count)
- Avoid collecting all values at a single node

Idempotent

Epidemic strategy

- Each node i starts with estimate $e = v_i$
- A node disseminates its current estimate e
- When receiving e_j, make $e_i = f(e_i, e_j)$
- Repeat $T = \log n + c$ times
  - Enough for each “opinion” to impact the whole network

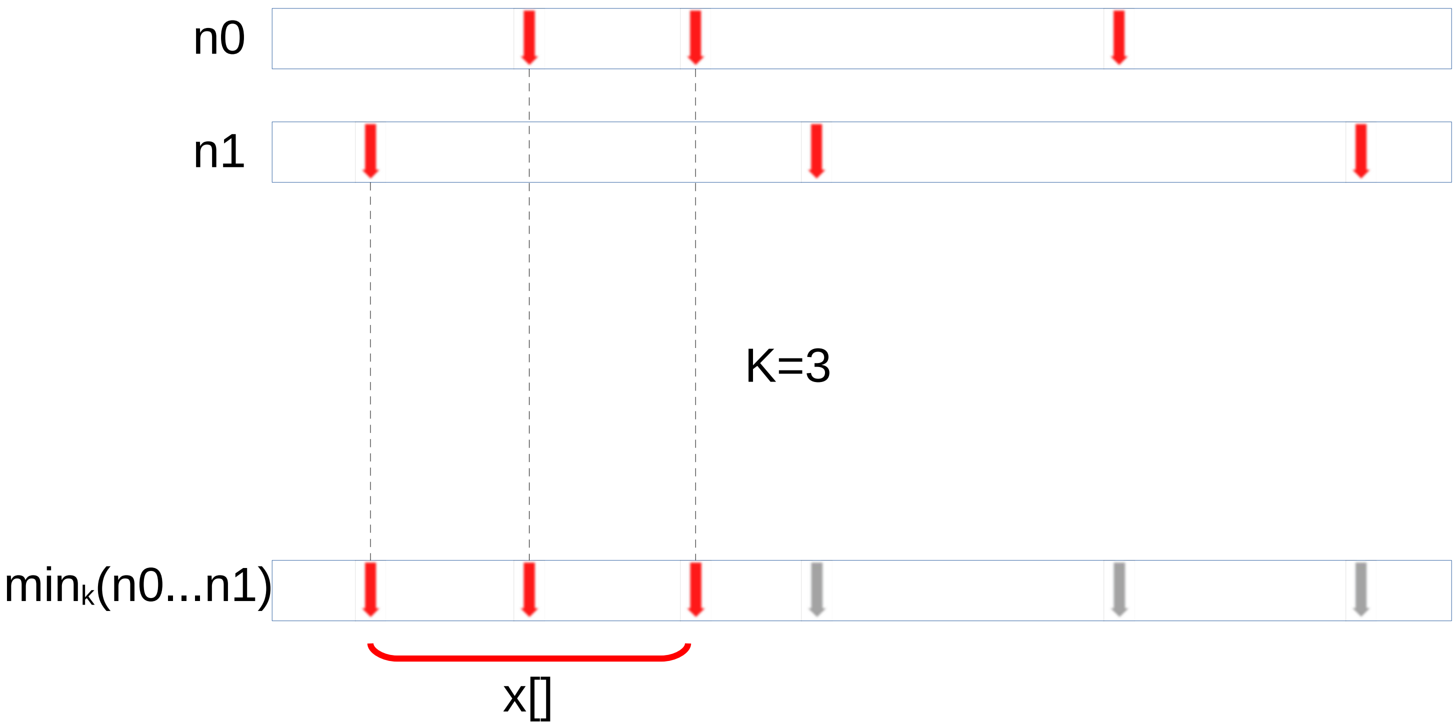
- Ok for min/max, but count and sum account opinions twice!

Extrema propagation

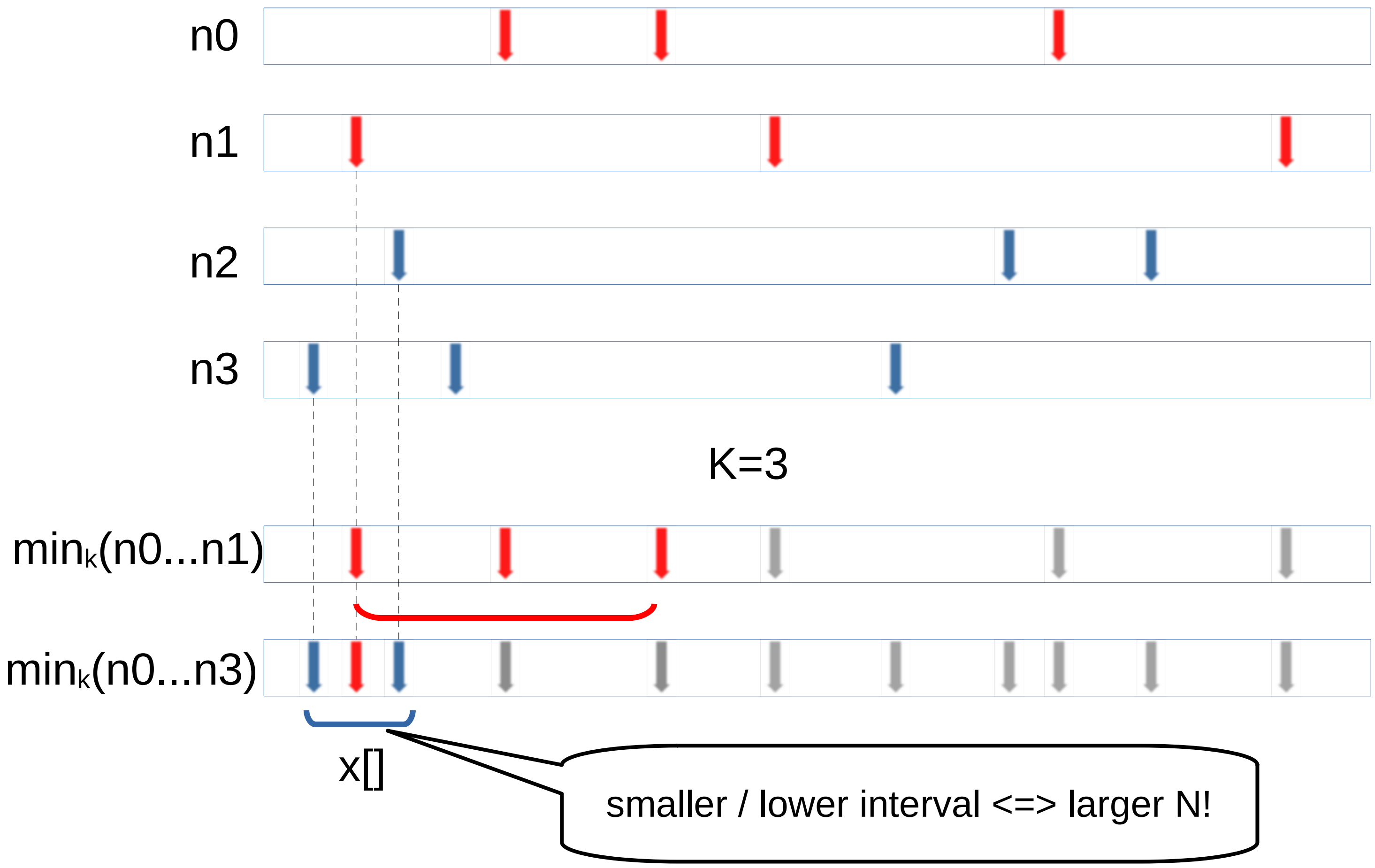
- Each of N nodes selects a vector x[] of K random values
- Nodes exchanges vectors and keep minimal values seen
  - Eventually, all nodes have the same values!
- Repeat until vector x[] is unchanged for T rounds
- Estimate count N from resulting vector x[]

- Can be generalized to compute Σv_i, therefore can also compute averages

Extrema propagation (intuition)



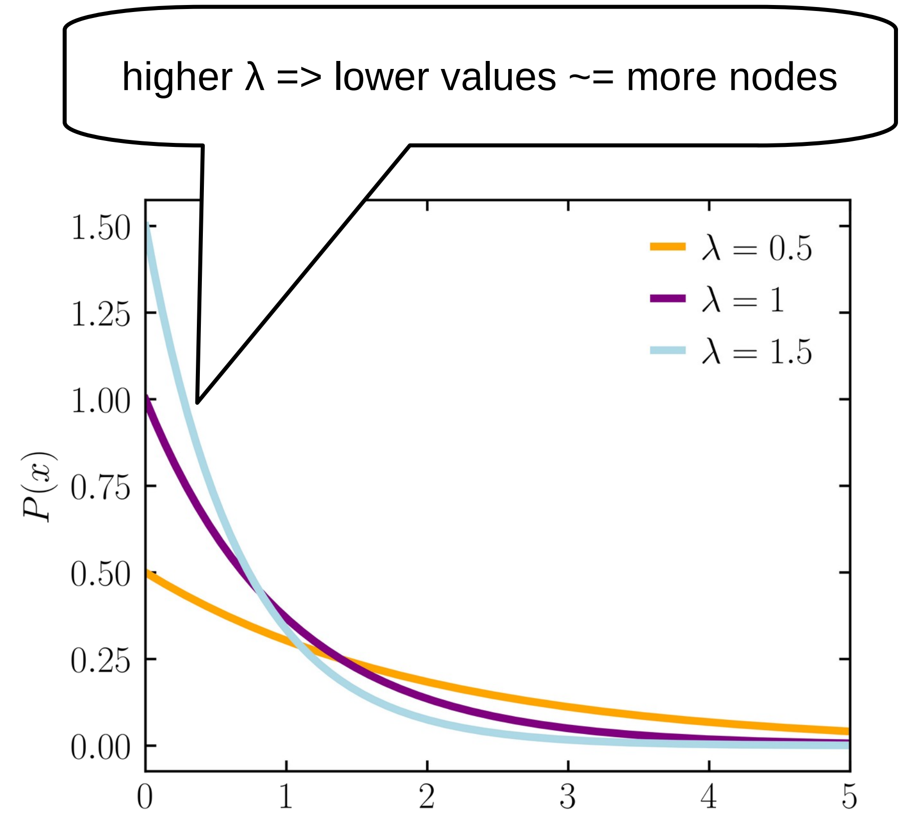
Extrema propagation (intuition)



Extrema propagation

- With exponential distribution:
  - Values from exp(1) in N nodes, the same as...
  - Values from exp(N) in 1 node
- If node i generates K values from $\exp(v_i)$, then result is an estimate of Σv_i

  Sum = (K-1) / Σ x[]



References

- C. Baquero, P. Almeida, R. Menezes, and P. Jesus, "**Extrema Propagation: Fast Distributed Estimation of Sums and Network Sizes**," IEEE Trans. Parallel Distrib. Syst., vol. 23, pp. 668–675, Apr. 2012. http://dx.doi.org/10.1109/TPDS.2011.209