eecs589 Advanced Computer Networks

Floyd and Jacobson, "The Synchronization of Periodic Routing Messages," *IEEE/ACM Transactions on Networking*, 2(2):122–136, Apr. 1994

Observed: Periodic Packet Losses

Experiment sending 1000 pings between Berkeley and MIT, at 1 sec. interval

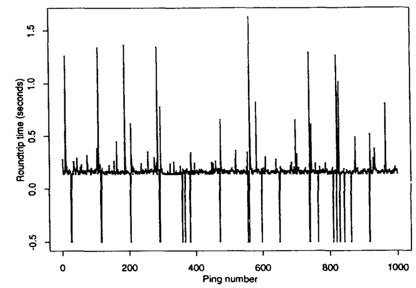


Fig. 1. Periodic packet losses

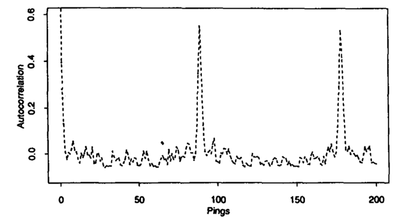


Fig. 2. The autocorrelation of roundtrip times.

Observed: Periodic Packet Losses

Similar periodic losses have also been observed on other networks running other routing protocols

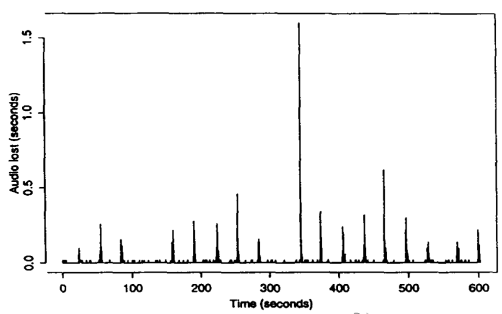


Fig. 3. Periodic packet losses at 30-second intervals on network running RIP.

What could cause the periodic losses?

Suspect: synchronized routing updates

Example of Synchronized Processes

Two pendulum clocks hanging on the same wall end up swinging in synch

Male Thai fireflies gathering at dusk in trees by the riverside flash on and off unsynchronized, but as the night progresses whole trees of fireflies flash in synch for hours

Weak Coupling and Synchronization

Pulse-coupled oscillator systems, e.g., pendulum clocks on wall, fireflies on tree, exhibit weak coupling between components

Weak-coupling leads to synchronization of dynamic systems

What kind of weak coupling causes synchronization of periodic routing messages?

Weak Coupling of Route Updates

Hypothesis: setting periodic route update timers after processing updates from other routers provides the weak coupling between routers that lead to synchronized route updates

Hard to test this hypothesis on real system: too many uncontrolled variables

Approach: create a simulation model

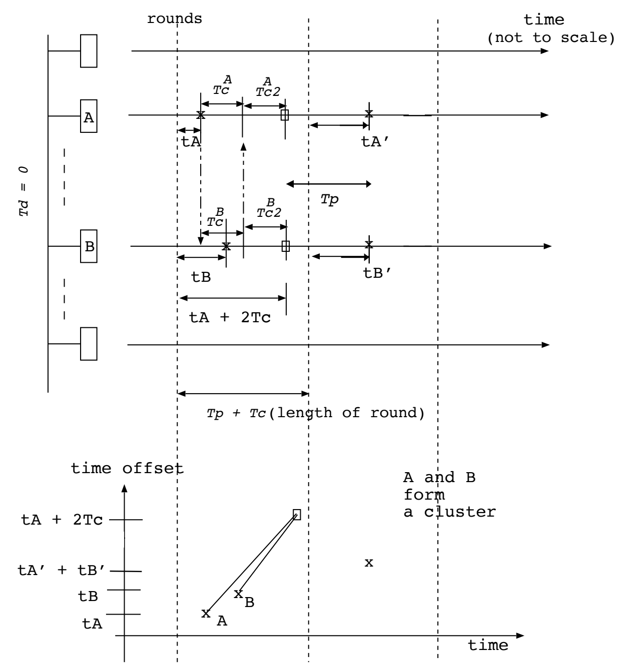
Timing Model of Route Updates

When an update timer expires,

- router A prepares and sends its routing updates (we'll assume that updates are streamed out)
- neighbor B receives first packet of updates T_d $(= 0)$ secs later
- it takes T_c secs for A to process an outgoing update
- it also takes T_c secs for B to process A's updates
- if B's route update timer expires during T_c, it waits until the end of T_c before handling the timer (takes $T_{c2} = T_c$ secs)
- if a neighbor's update arrived within T_c, processes update at the end of T_c (also takes $T_{c2} = T_c$ secs)
- after finish processing both updates, A and B set next update timer to T_p $(= 30)$ secs later

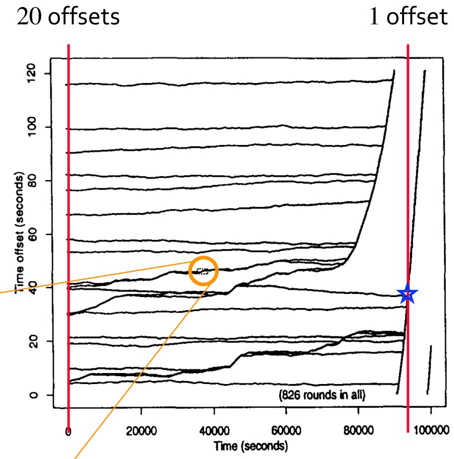
Cluster Formation

Processing routing updates before setting periodic timer induces a weak-coupling between routers that lead to synchronization of routing messages



Simulation

Simulation of $N = 20$ routers shows that clustering forms and breaks up, but eventually the routers all synched up



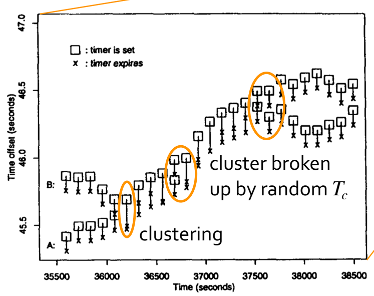


Fig. 5. An enlargement of the simulation in Fig. 4.

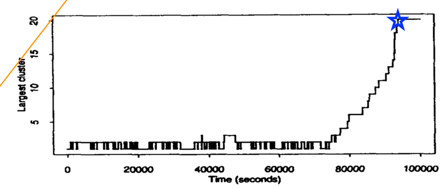
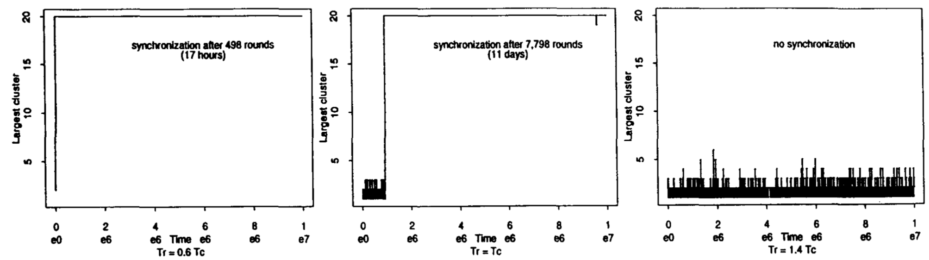


Fig. 6. The cluster graph, showing the largest cluster for each round.

How to Prevent Synchronization?

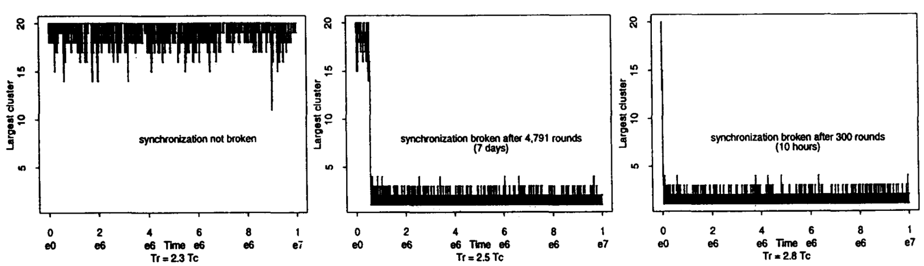
Add a random factor (T_r) to the periodic timer

Starting with unsynch-ed updates, $T_r = 1.4T_c$ manages to prevent synchronization



How to Break-up Synchronization?

Starting with synch-ed updates, perhaps due to triggered updates or routers reboot, $T_r = 2.8T_c$ manages to break up synchronization



How big must T_r be?

Use a Markov Chain model to formally analyze how long it takes for a cluster of size k to form and to break up

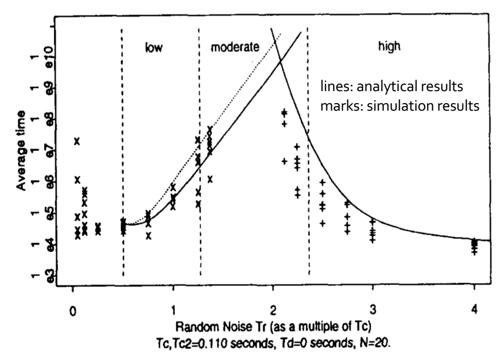


Fig. 14. Expected time to go from cluster size 1 to cluster size N, and vice versa, as a function of T_r.

Results explanatory not predictive, analytical model shows the same qualitative behavior as simulation model (but off by several factors)

How big must T_r be?

Analytical model simplifies reality even further, it only approximates the simulation model, which in itself is a simplification of reality, but the analytical model illustrates some significant properties of the simulation model (and reality)

Choosing T_r as $.5T_p$, i.e., max randomization, larger than T_p will become in-phase again, should eliminate synchronization

Synchronization is a Phase Transition

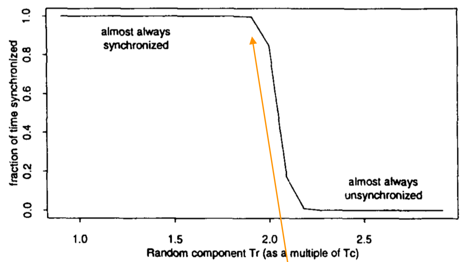


Fig. 18. The fraction of time synchronized versus the random component T_r.

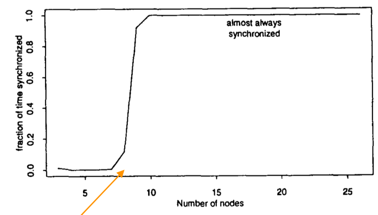


Fig. 19. The fraction of time synchronized versus the number of nodes, for $T_r = 0.11$ seconds.

Abrupt change

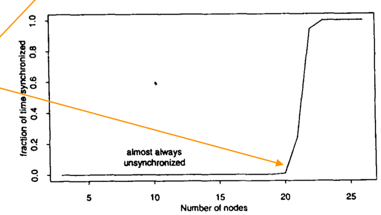


Fig. 20. The fraction of time synchronized versus the number of nodes, for $T_r = 0.30$ seconds.

Again, qualitatively descriptive, not quantitatively prescriptive

Models and Network Dynamics

Simple, innocuous behavior can cause (unsuspected) emerging coordination among entities that leads to complex global structure

Given the observed complex behavior/structure, need to isolate the simple innocuous behavior that gives rise to it

Model the behavior

Hypotheses on network dynamics can only be studied within a very simplified model of the real system

The model must be simple enough to isolate the suspected behavior(s), yet realistic enough to allow the complex behavior to emerge

Confirm observations from model on real network

Reporting Results

Simulation results can depend on random seed used

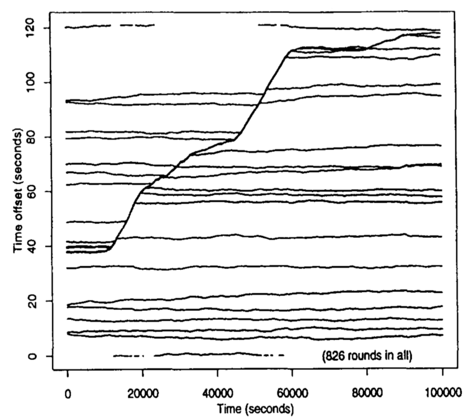


Fig. 7. A simulation showing unsynchronized routing messages.

Be honest in reporting negative cases

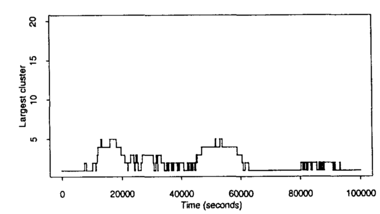


Fig. 8. The cluster graph, showing the largest cluster for each round.

Reporting Results

Analytical model may be over simplified and quantitatively off by several factors, but is still useful to explain behavior

Be honest in reporting loose bounds

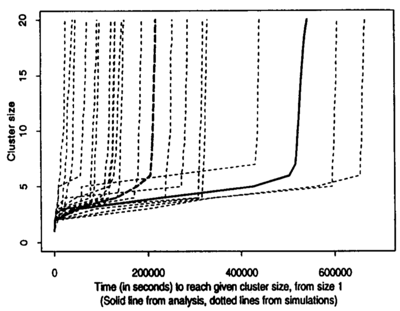


Fig. 12. The expected time to reach cluster size i, starting from cluster size 1, for $T_r = 0.1$ seconds.

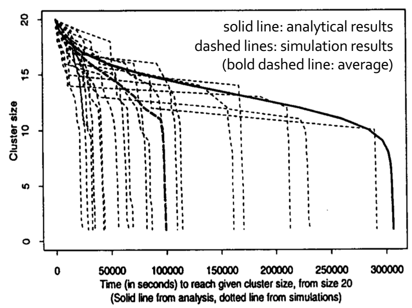


Fig. 13. The expected time to reach cluster size i, starting from cluster size N, for $T_r = 0.3$ seconds.

Reporting Results

Double check sensitivity of results to parameter values:

- different network topologies: point-to-point networks
- size of N and T_c
- non-zero T_d's: $0 < T_d < T_c$ (no synch if $T_d > T_c$)

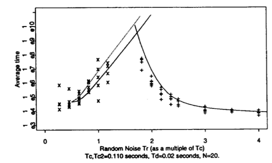


Fig. 16. Time to go from a cluster of size 1 to a cluster of size N, and vice versa, for $T_d = 0.02$ seconds.

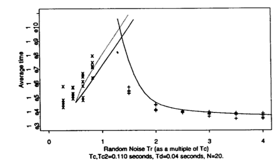


Fig. 17. Time to go from a cluster of size 1 to a cluster of size N, and vice versa, for $T_d = 0.04$ seconds.

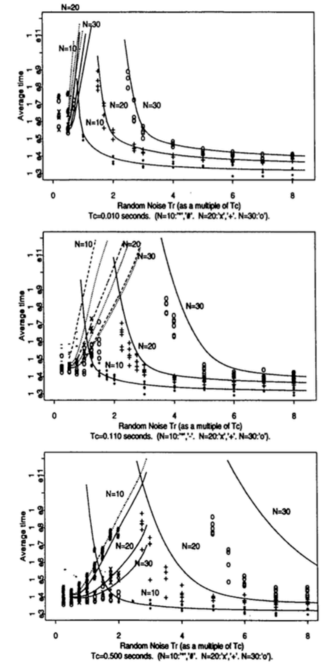


Fig. 15. Expected time to go from cluster size 1 to cluster size N, and vice versa, as a function of N and of T_r.