Towards More Adaptive Internet Routing

Mukund Seshadri
(mukunds@cs.berkeley.edu)

Prof. Randy Katz

Motivation – Inter-domain Routing

- Inter-domain routing failures often last several minutes [Labovitz et al.]
  - Often due to slow BGP convergence
  - Can take up to 20 min to recover
- Reachability failures can be circumvented by using alternate routes [RON]
  - But these alternate routes are not made available via BGP
  - So, overlays were used – small-scale solution only.
- Can we modify inter-domain routing to utilize alternate routes (when available)?

Motivation – Intra-domain Routing

- Typically done by setting (OSPF) weights to achieve desired utilizations (for known traffic matrix)
- Higher reactivity, greater stability and better capacity planning than inter-domain routing
    - But can Performance be a problem?
- Cannot adapt to changes in traffic load
    - Currently addressed by heavy over-provisioning

- If high overprovisioning is not feasible, and variations in demand are on a faster timescale than that of traffic engrs.: can we automatically adapt to changes in the load?
    - Packet-switching, no reservation-based models
        - Does not change the interface to end-hosts or other networks

Inter-domain: Approach

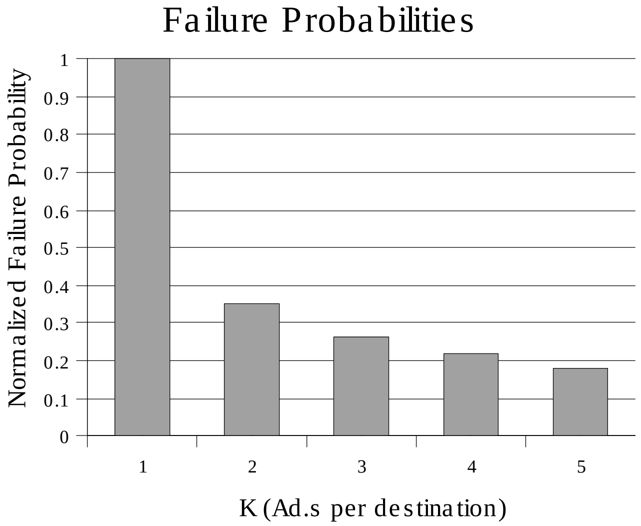
- Extend BGP's path vector protocol to advertise *k* (~2) routes per destination instead of 1.
  - Factor k increase in advertisement overhead
- The first of the k routes is the default BGP route as computed today.
- The remaining k-1 routes are selected to be maximally link-disjoint (at the AS-level).
  - Sequential greedy selection of routes
  - Heuristic to reduce probability that a change to the default route will be accompanied by a change to the alternate routes (assuming random single-link failure)

Service Model

- How will the use of the alternate routes be triggered?
  - Network node can automatically switch to alternate route when the default route changes; and continue to use alternate routes for a certain period, until the default routing entry stabilizes
    - Alternate routing entry will not be changed when the default entry is changing
  - Ultimately, the only way to validate a routing entry is to send and receive packets via that route
    - End-hosts already do this – can indicate reachability failures via a flag in the packet-header.

Results

- Construct AS-level topologies and default paths from BGP Routeviews data.
  - Inaccuracies due to symmetry assumption and hidden edges
  - ~500 nodes, 100 src/dest.
- Construct routing tables using *k*-path vector.
- Find reachability/failure probability of all destinations for a given node (under random single link failure)



- Clearly, just using *k*=2 greatly improves reachability

Our approach – Route granularity

- One route per destination network node => high-volume unit of re-routing => harder to load-balance
- Therefore a node A divides the traffic through it to a particular destination node into *B* buckets
  - Division into buckets can be done independently by the network node, by using a hash of source and destination IP header fields, thus not affecting the interface to other networks or end-hosts.
  - Small B is desirable; otherwise it would devolve into per-flow routing state.
  - Dual – fixed bucket size, variable B.
- One route is maintained for each bucket

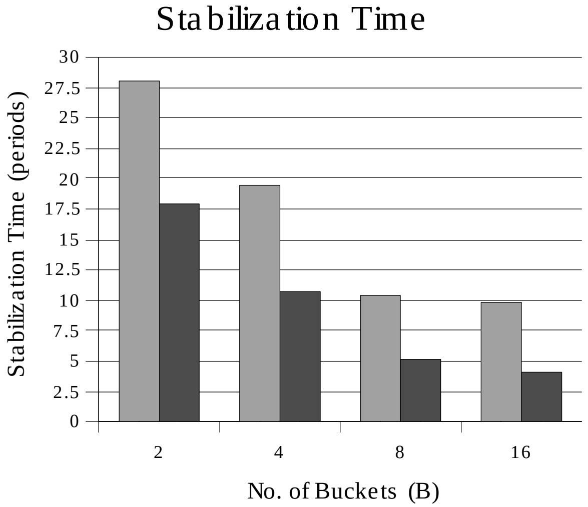
Our Approach – Randomization

- Link state is inherently stale
  - This can cause herd behaviour, leading to instability and imbalance
- We introduce randomness into the routes selected across different buckets for the same destination
  - Randomly choose from r best routes.
  - Best of r random routes (selected proportional to static costs)
- [Mitzenmacher97] showed that "best-of-2 random selection" was ideally suited for server load-balancing with stale info.

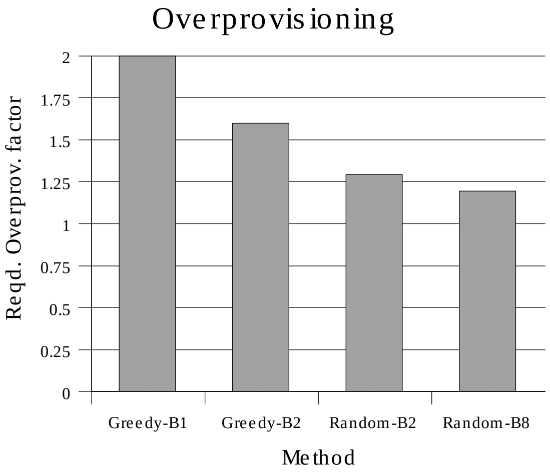
- Link state used is a load-based metric
  - Without randomization and bucketing, this can be extremely unstable.

Results

- “Random fork/t-s topologies
  - Flow-level simulation.
- Bucketization” improves stabilization times (and loss rates) even with moderately low values of ***B***
  - Since the unit of traffic change becomes significantly lower than total link loads.
- Random selection is a significant improvement over best-first selection.



Random: Red bars
Best-first: Blue bars



Overprovisioning required to reduce stabilization time to less than 10 periods.

Future Work

- Better, Dynamic Evaluation Scenario
  - Failure location/time data for inter-domain routing
  - Traffic matrix and topology for intra-domain routing
- Better metric for load-sensitive routing
  - Use model of state change.
  - Effect of filtering, incorporate delay info.
- Inter/intra interactions?