Introduction to compact routing

Dmitri Krioukov
UCSD/CAIDA
dima@caida.org

IDRWS 2004

Initial interest: theoretical (fundamental) aspects of routing on graphs

Interest crystallization history:

- Scalability concerns
  - Convergence
  - Routing table size
- Immediate causes
  - Routing policies
  - Increasing topology density
    - Multihoming
  - Address allocation policies
  - Inbound traffic engineering, etc.
- Various short-term fixes
  - Let’s consider one of them

Routing on AS#s (ISLAY,atoms)

Disregarding practical problems associated with it, this idea does not solve anything in the long run: small multihomed networks requiring $O(1)$ IP addresses will lead to the situation with the total number of ASs being of the same order as the number of IP addresses.

Crystallization history (contd.)

- Put aside routing policies (another interesting problem tackled by others☺)
- Level of abstraction: AS graph, which is a fat-tailed and scale-free small-world
- Problem becomes: theoretical lower and upper bounds for routing on massive fat-tailed scale-free small-world graphs

Fat-tailed scale-free small-worlds

- "Small-world" = there is virtually no long paths ('remote' nodes), i.e. the distance distribution has small average and dispersion
- "Fat tail" (e.g. power-law) of the node degree distribution = there is a noticeable amount of high-degree ('hubby') nodes ⇒ the graph has a 'core' ⇒ small-world
- "Scale-free" node degree distribution (e.g. power-law) = there is no 'hill' (characteristic scale) in it ⇒ there is a lot of low-degree ('edgy') nodes ⇒ the graph is 'hairy'
- Colloquially: scale-free = power-law

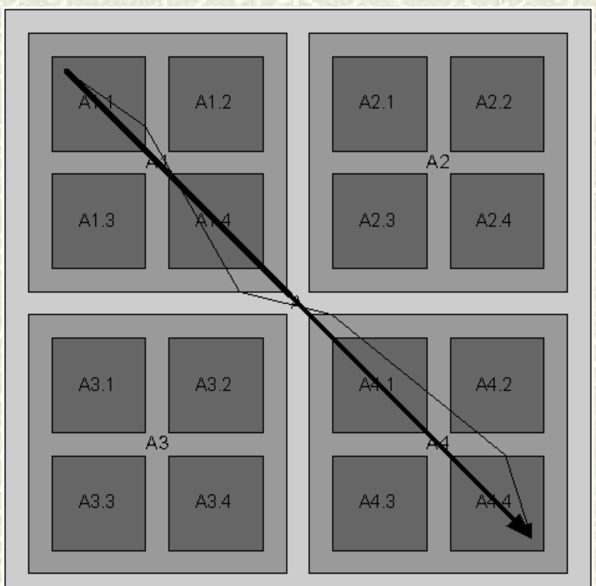
Assessment of known facts: networking community

- Hierarchical aggregation, multiple level of abstraction, i.e. Nimrod, MLOSPF, ISLAY, i.e. Kleinrock-Kamoun's hierarchical routing scheme of 1977 (KK).
- But: there is a cost associated with KK routing table size reduction: path length increase. It depends strongly on a particular topology

KK path length increase

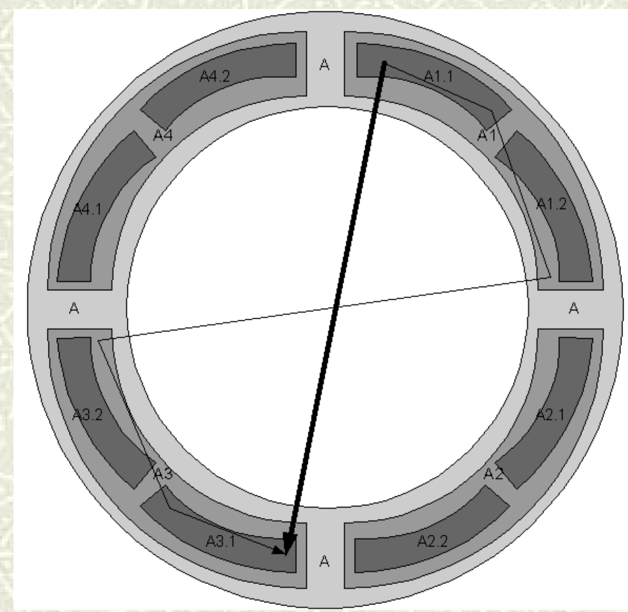
Sparse topology

- $\langle L(n)\rangle \to \infty$, $\langle L_{kk}(n)\rangle \to \infty$ s.t. $\langle L_{kk}\rangle / \langle L\rangle \to$ const.
- There are remote points



Dense topology

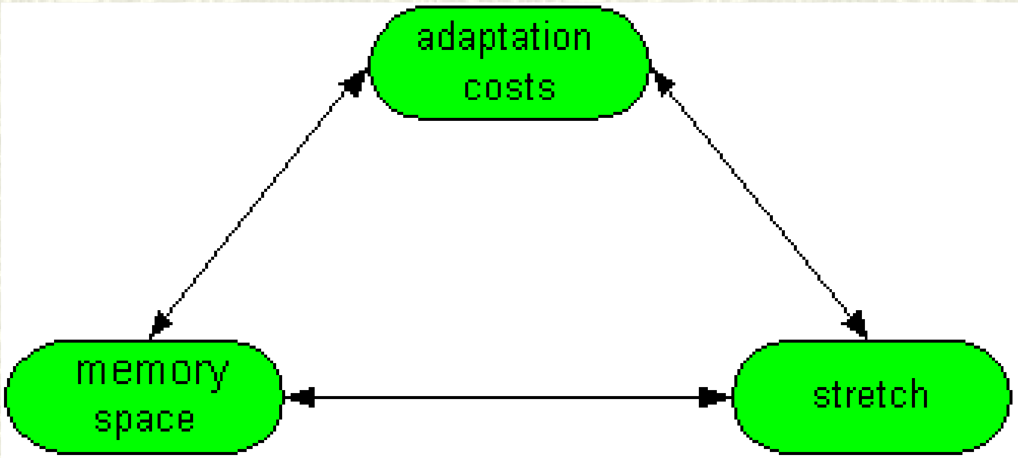
- $\langle L(n)\rangle$=const. ($\langle \text{degree}\rangle \to \infty$ instead) but $\langle L_{kk}\rangle \to \infty$ so that $\langle L_{kk}\rangle / \langle L\rangle \to \infty$
- There are no remote points, so that one cannot usefully aggregate, abstract, etc., anything remote—everything is close



What does path length increase mean in practice?

- Consider a couple of peering ASs. Their peering link is the shortest path between them. Non-shortest path routing may not allow them to use it, which is unacceptable.
- BGP is shortest path **if we ‘subtract’** policies (there is no view of global topology anyway). Distance and path vector algorithms are ‘shortest path’ algorithms by definition.
- Path length increase associated with routing table size decrease is a concern. On the AS topology, the KK scheme produces 15-times path length increase. Can anyone do better?

Assessment of known facts: distributed computation theory



Triangle of trade-offs:

- Adaptation costs = convergence measures (e.g. number of messages per topology change)
- Memory space = routing table size
- Stretch = path length inflation

Crystallization history (contd.)

- Simplify the task: put adaptation costs aside, i.e. assume they are unbounded, i.e. consider the static case. Reasons include:
  - BGP adaptation costs are unbounded (persistent oscillations)
  - The negative answer (memory space and stretch cannot be made simultaneously small on scale-free graphs) was expected. Reasons:
    - KK stretch on the Internet
    - High stretch of other schemes on complete network and classical random graphs
- Question: what is the "best" static routing scheme? Answer: stretch-3 routing by Thorup and Zwick (TZ). Reasons:
  - Maximum stretch of 3 is the minimum value of maximum stretch allowing for sub-linear memory space lower bounds
  - TZ is the only known nearly optimal (memory space upper bound = lower bound) stretch-3 routing

TZ scheme

- **Landmark set (LS) construction**: iterations of random selections to guarantee the right balance between the cluster size and LS size (as opposed to the greedy set cover algorithm by Lovasz in the Cowen case)
- **Routing table**: shortest paths to the local cluster nodes and landmarks
- **Labeling**: original node ID, its closest landmark ID, the ID of the port at the closest landmark towards the node
- **Forwarding at node *v* to destination *d*:**
  - If $v = d$, done
  - If *d* is in the routing table (cluster or landmark), route appropriately
  - If *v* is *d*'s landmark, the outgoing port is in the destination address in the packet
  - Default: *d*'s landmark in the destination address in the packet and the route to this landmark is in the routing table

End of story

- Done: considered the “best” static routing scheme (TZ) and analyzed its average memory-stretch trade-offs on Internet-like topologies.
- Found:
  - Both stretch and memory can be made extremely small simultaneously but **only** on scale-free graphs
  - A number of other unexpected interesting phenomena suggesting that there are some profound yet unknown laws of the Internet (and maybe some other networks) topology evolution

References

- Presentation: http://www.caida.org/~dima/pub/crig-ppt.pdf
- Infocom version: http://www.caida.org/~dima/pub/crig-infocom.pdf
- Technical report version: http://www.caida.org/~dima/pub/crig.pdf