Compact Routing: Challenges, Perspectives, and Beyond

TRILOGY Future Internet Summer School 2009

August 24-28, 2009

Louvain-la-Neuve, Belgium

Dimitri Papadimitriou

dimitri.papadimitriou@alcatel-lucent.be

Alcatel-Lucent Bell NV

Abstract

Title: Compact Routing: Challenges, Perspectives, and Beyond

Abstract: Compact routing schemes were until recently considered as the main alternative to overcome the fundamental scaling limitations of the Internet routing system. Such schemes have been introduced to address the fundamental and unavoidable trade-off between the stretch of a routing scheme and the routing table size it produces. Recent studies have shown that static routing on topology-dependent identifiers (that includes "some" topological information) scales (poly)logarithmically on the number of nodes in scale-free graphs such as the Internet. Nevertheless, evolution of the Internet routing system also demonstrates that i) topology-independence of the numbering scheme becomes a fundamental requirement for e.g. multi-homing, nomadicity, and mobility, and ii) "static" routing is unable to handle dynamic graphs, i.e., routing update messages (triggered by topology-driven, policy-driven, and other protocol-driven events) must be exchanged to timely inform remote routers such that each router maintains a consistent view of the non-local topology.

This tutorial starts by reviewing the fundamental dimensions of routing schemes. It then introduces the compact routing theory and its application to scale-free graphs. Next, it describes and analyses the consequences resulting from i) routing on topology-independent numbering space and ii) dynamic routing information exchanges and resulting costs. This tutorial will conclude by outlining the open research coupled-challenges (stretch x routing table size x dynamics) and possible direction(s) to address them.

Tutorial Outline

- Compact routing overview
- Motivations
- Objectives (Fundamental dimensions)
- Compact routing theory
- Compact routing schemes
- Open research challenges

Compact Routing Overview

Compact Routing (1)

Two base / solutions to the routing problem

1. Store a complete routing table at each node of the network
   - RT entries for any destination next-hop/link to which packets for that destination should be forwarded (shortest path)
   - Drawback: requires each node to store $\Omega(n \log(n))$ bits of routing information

2. Source-explicit/Source-directed routing
   - Source inserts for each packet carries in its header a complete description of the path along which it should be routed
   - Drawback: packet headers may need to be of size (n) and still processed at wire speed

In either case, we have problems with scaling: routing schemes that require nodes to store a linear number of bits are not scalable

Compact Routing (2)

- To reduce memory space and incur routing costs proportional to actual distances, two parameters that routing schemes aim to minimize: stretch and memory
  - Stretch: ratio between routing path length/cost and minimum cost/length path
  - Memory: number of bits of routing information stored locally for the routing scheme

- Definition: a routing scheme is said to be compact if it produces
  - Logarithmic node names/labels and header sizes
  - Sublinear upper bound on Routing Table (RT) size - local memory space
  - and Stretch bounded by a constant (remains constant independently of the network size growth)

Compact Routing (3)

Challenges

- Fundamental and unavoidable trade-off between stretch of routing algorithm and size of RT it produces (memory space: number of bits of routing information stored per node)

  => *Find memory – stretch trade-off toward routing schemes with substantially smaller RT, yet having headers of only logarithmic size*

- Construct in polynomial time a compact routing scheme that minimizes the stretch bound for any weighted graph while requiring only o(n) bits of routing information per node (trade-off between memory and stretch is efficiently balanced)

- Note: compact routing algorithms make RT sizes compact by omitting some topological details (in an efficient way) such that path length increase stays bounded

Motivations

Problem Statement (1)

Problem 1: **Topology vs aggregation**

- Originally, host addresses assignment (PA) based on network topological location
- Conditions to achieve efficient address aggregation and relatively small routing tables (tradeoff routing information aggregation vs granularity) are not met anymore [RFC4984]
- Deterioration root causes:
  - Host mobility, site multi-homing (~25% of sites), traffic-engineering (prefix de-aggregation)
  - RIR policy to allocate PI addresses (not topologically aggregatable) thus making CIDR ineffective

⇒ Super-linear growth of Routing Table (RT) even if network itself would not be growing (***routing protocol must not only scale with increasing network size !***)

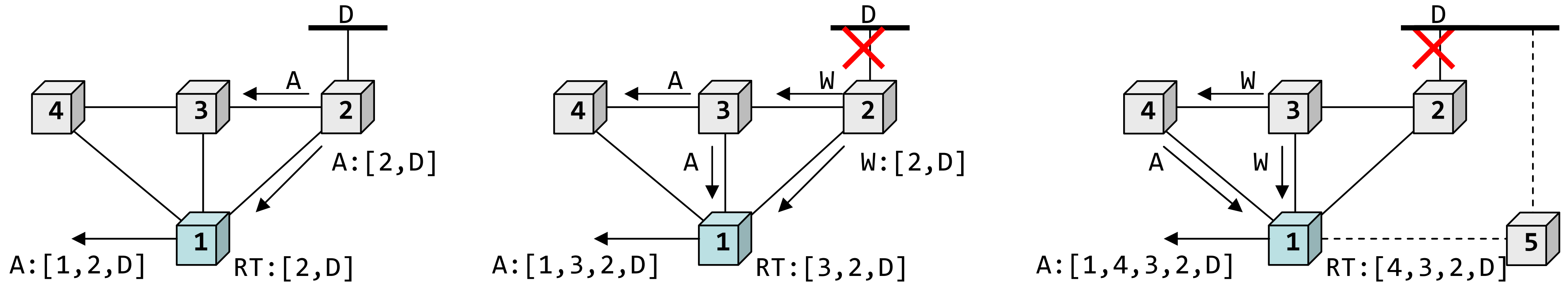
Problem Statement (2)

Problem 2: Inter-domain routing protocol (BGP)

Implementation specifics: may be circumvented

Architecture: shortest AS-path vector inter-domain routing (avoids AS-loops, eliminates DV count-to-infinity) BUT subject to Path Exploration

- BGP slow convergence due to uninformed path exploration:
  - Theoretical: upper bound ~ O(N!) and lower bound = Ω[(N-3) x MRAI timer] on state explored during BGP convergence (N = number of AS)
  - Practice: (Max_AS-Path - Min_AS-Path) x MRAI timer



Problem Statement (2)

Problem 2: Inter-domain routing protocol (BGP)

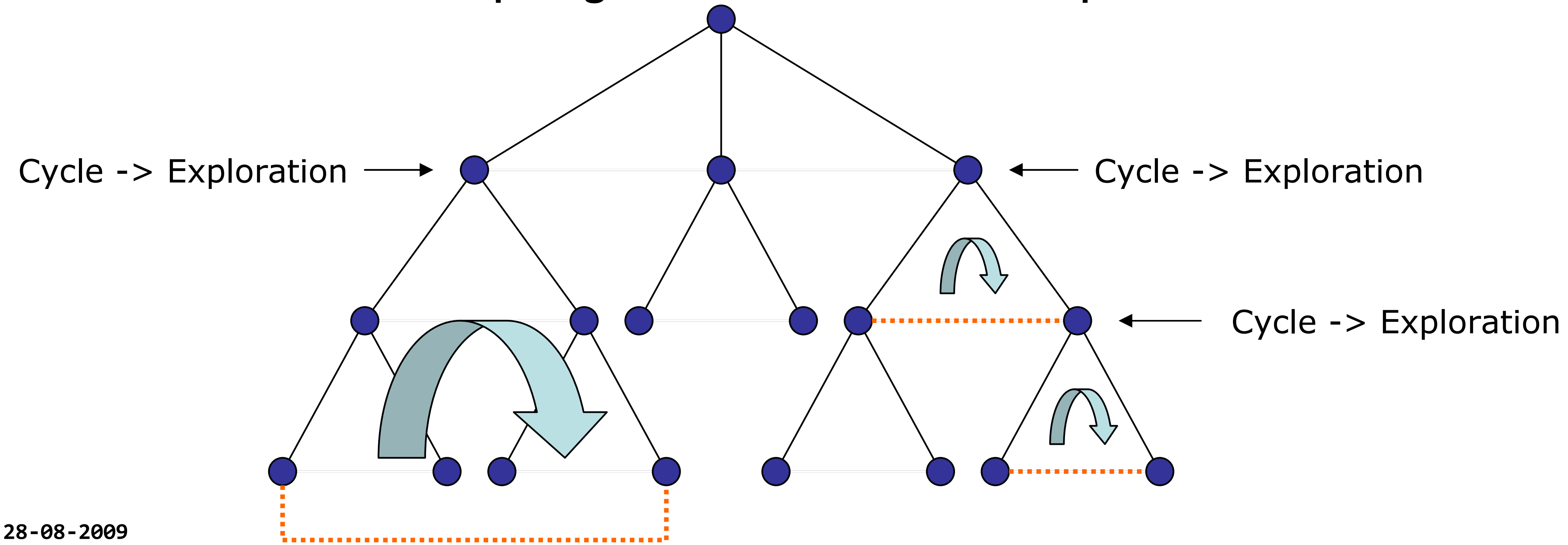
- Mitigation (examples):
  - Root cause notification (pin location/cause of updates)
  - Shorten MRAI (with or without RCN) to fasten convergence: affects routing updates rate limitation used to dampen some oscillations inherent to path vectors
  - Backup/Multi AS-path (increase AS-Path diversity): affects number of RIB states

Relationship to Topology

- Meshed AS topology (avg AS transit inter-connection degree ~2.5-3) with high 3-4 clustering coefficient
- BGP (uninformed) Path exploration
  - BGP listens w/o understanding (local BGP route selection)
  - BGP updates are neither coordinated in space nor in time

    -> **Topology dynamics and policy changes result in correlated BGP updates (AS correlation)**
  - BGP rate limits updates (MRAI timer)

    -> **State coupling between correlated updates**



Problem Statement (3)

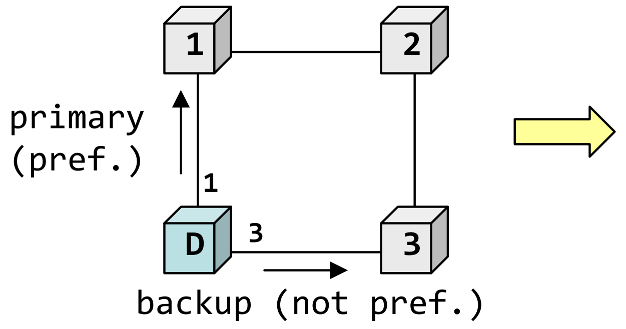
Problem 2: Inter-domain routing protocol (BGP)

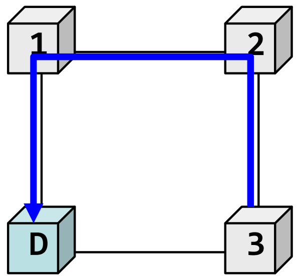
Policy-based routing (no policy distribution)

→ Intra-AS oscillations: MED-induced oscillations

→ Inter-AS oscillations: local pref. over shortest AS-Path

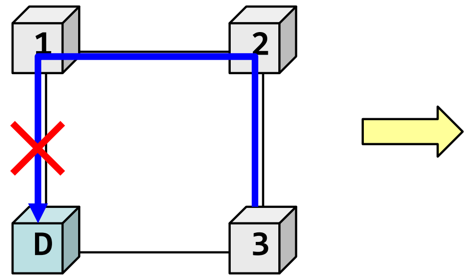
Conflicting policy interactions i) unintended stable state (wedgies)

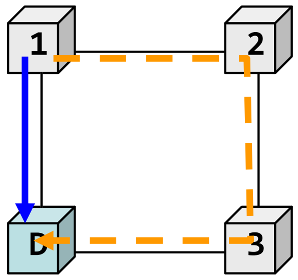




Configuration phase:

traffic primary (backup) incoming interface at D is 1 (3)





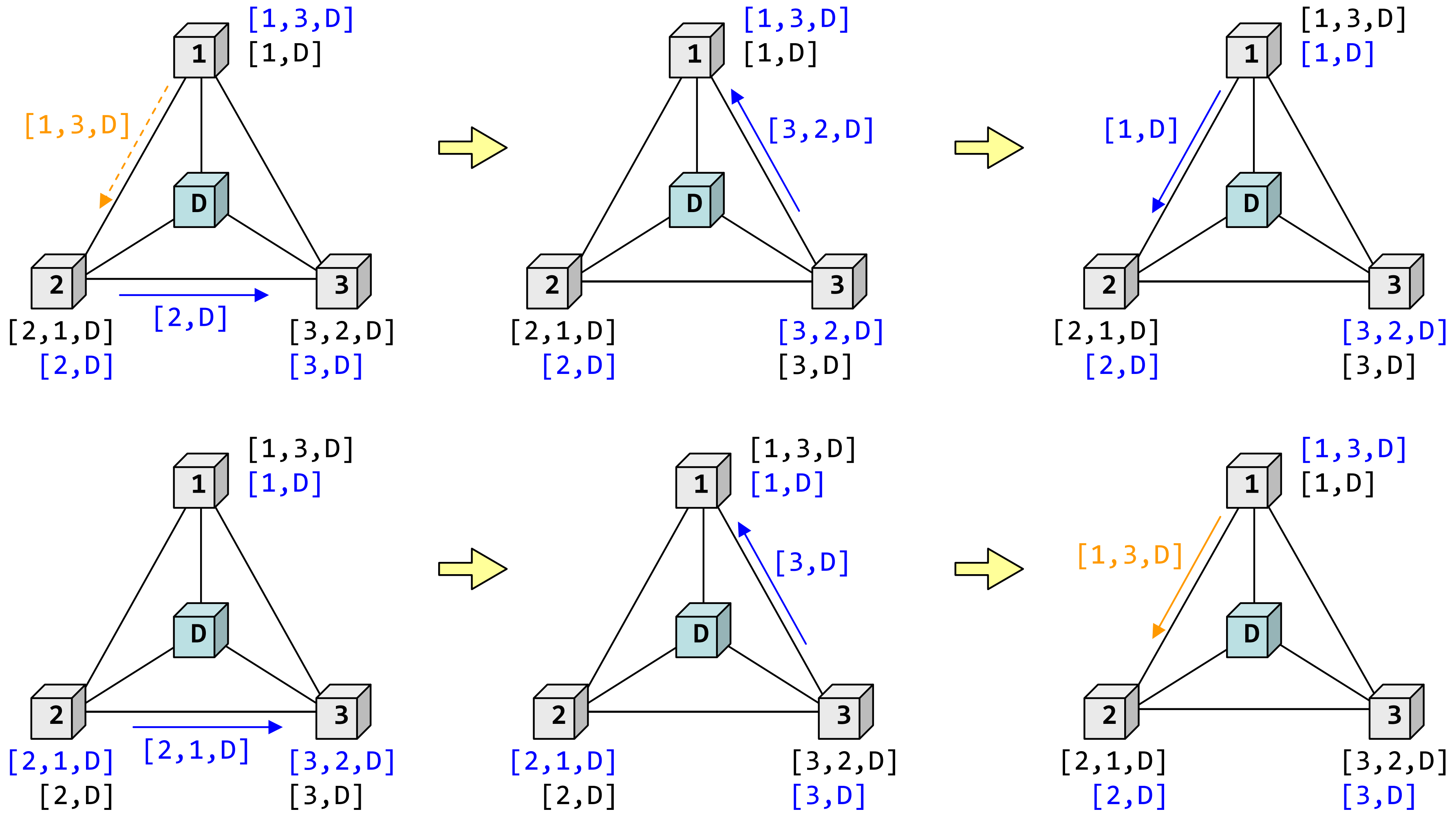
Failure occurrence at link (1,D): traffic incoming interface at D is 3

Failure recovery of link (1,D): traffic incoming interface at D is 3 AND 1

Problem Statement (4)

Problem 2: Inter-domain routing protocol (BGP)

Conflicting policy interactions ii) unintended unstable state (dispute wheels)



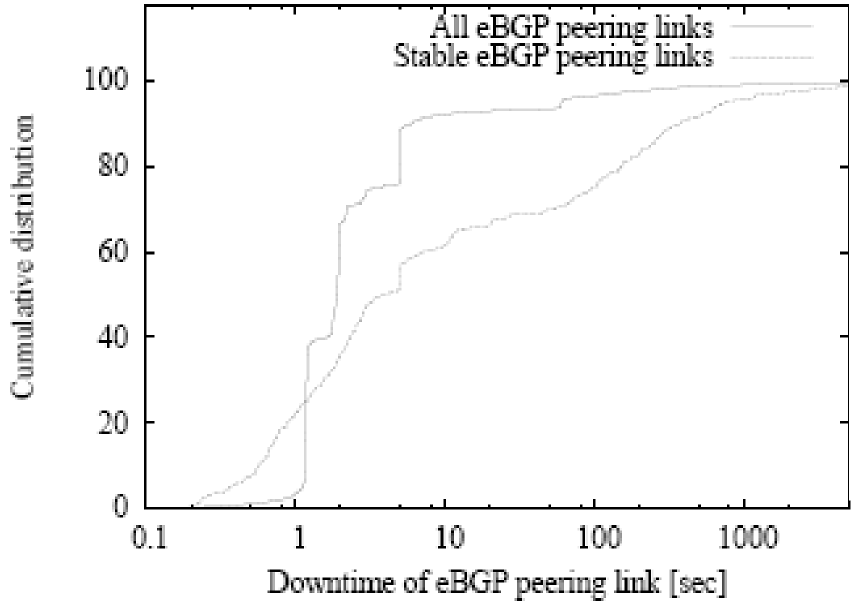
BGP Instability Causes

BGP peering link failures

- Common events (~70% of instability) that occurs everywhere but mostly at edge networks
- Transient events of duration ~ O(1s)-O(10s)
  - 82% of eBGP peering link failures last less than 180s
  - 22% of eBGP peering link failures lasted less than 1s
- Small number of links are responsible for large fraction of failures (flapping links)

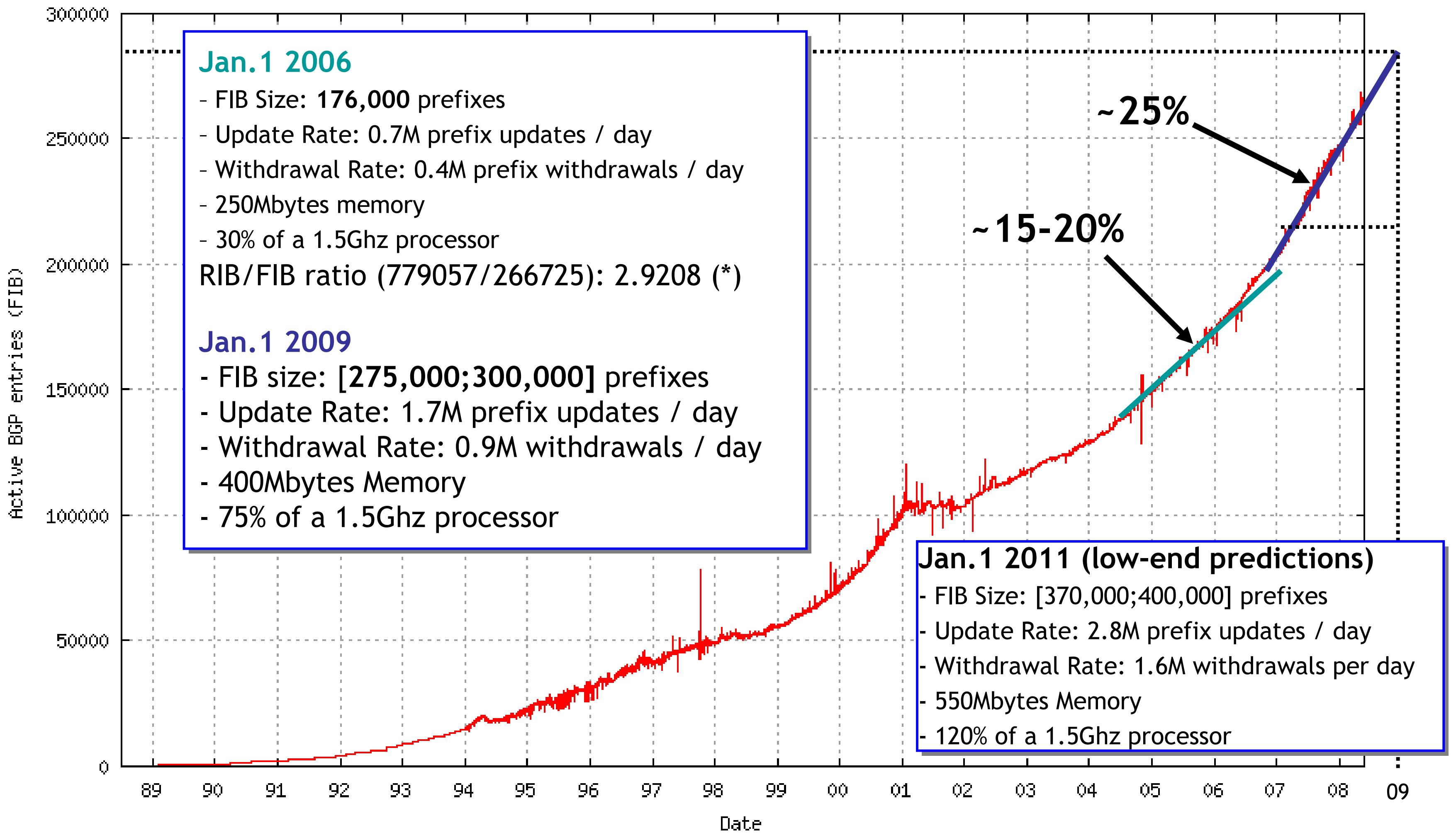
BGP operational instability

Instability	**Examples**
BGP Session availability	Session establishment/teardown/reset
BGP Session filters	Filter and/or BGP attribute changes usually imply session (soft-)reset or graceful restart
IGP costs changes	IGP metric changes
IP address changes	Renumbering
Originator changes route	Addition/deletion of network prefixes



Source: “Achieving Sub50 Milliseconds Recovery Upon BGP Peering Link Failures”, O.Bonaventure et al, ACM Co-Next 2005.

Growth of Active BGP Entries
(from Jan'89 to Mar'08)



Jan.1 2006
- FIB Size: **176,000** prefixes
- Update Rate: 0.7M prefix updates / day
- Withdrawal Rate: 0.4M prefix withdrawals / day
- 250Mbytes memory
- 30% of a 1.5Ghz processor

RIB/FIB ratio (779057/266725): 2.9208 (*)

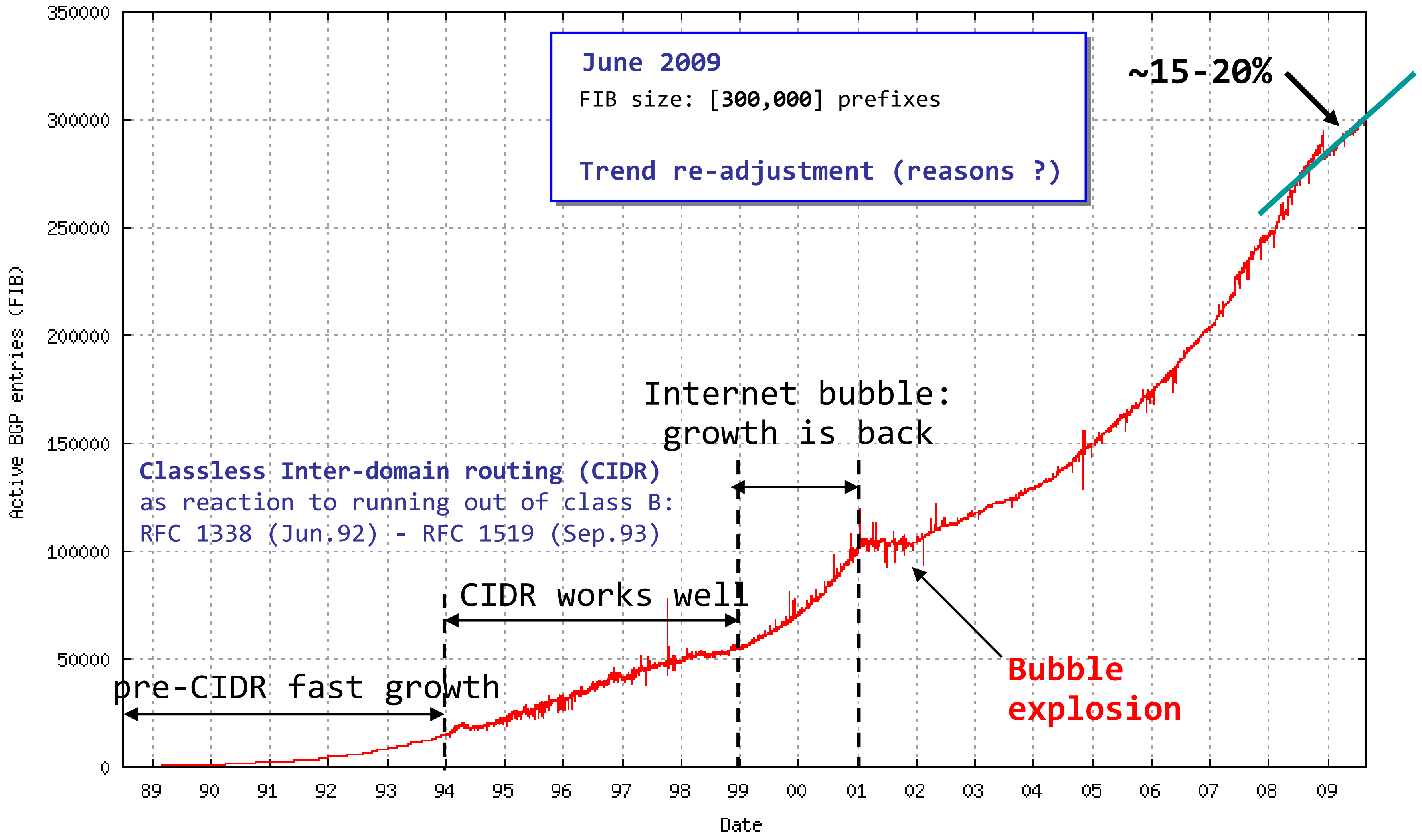
Jan.1 2009
- FIB size: **[275,000;300,000]** prefixes
- Update Rate: 1.7M prefix updates / day
- Withdrawal Rate: 0.9M withdrawals / day
- 400Mbytes Memory
- 75% of a 1.5Ghz processor

Jan.1 2011 (low-end predictions)
- FIB Size: [370,000;400,000] prefixes
- Update Rate: 2.8M prefix updates / day
- Withdrawal Rate: 1.6M withdrawals per day
- 550Mbytes Memory
- 120% of a 1.5Ghz processor

(*) RIB/FIB ratio can vary from ~3 to 30 (function of number of BGP peering sessions at sample point)

Source: BGP Routing Table Analysis Reports - http://bgp.potaroo.net/index-bgp.html

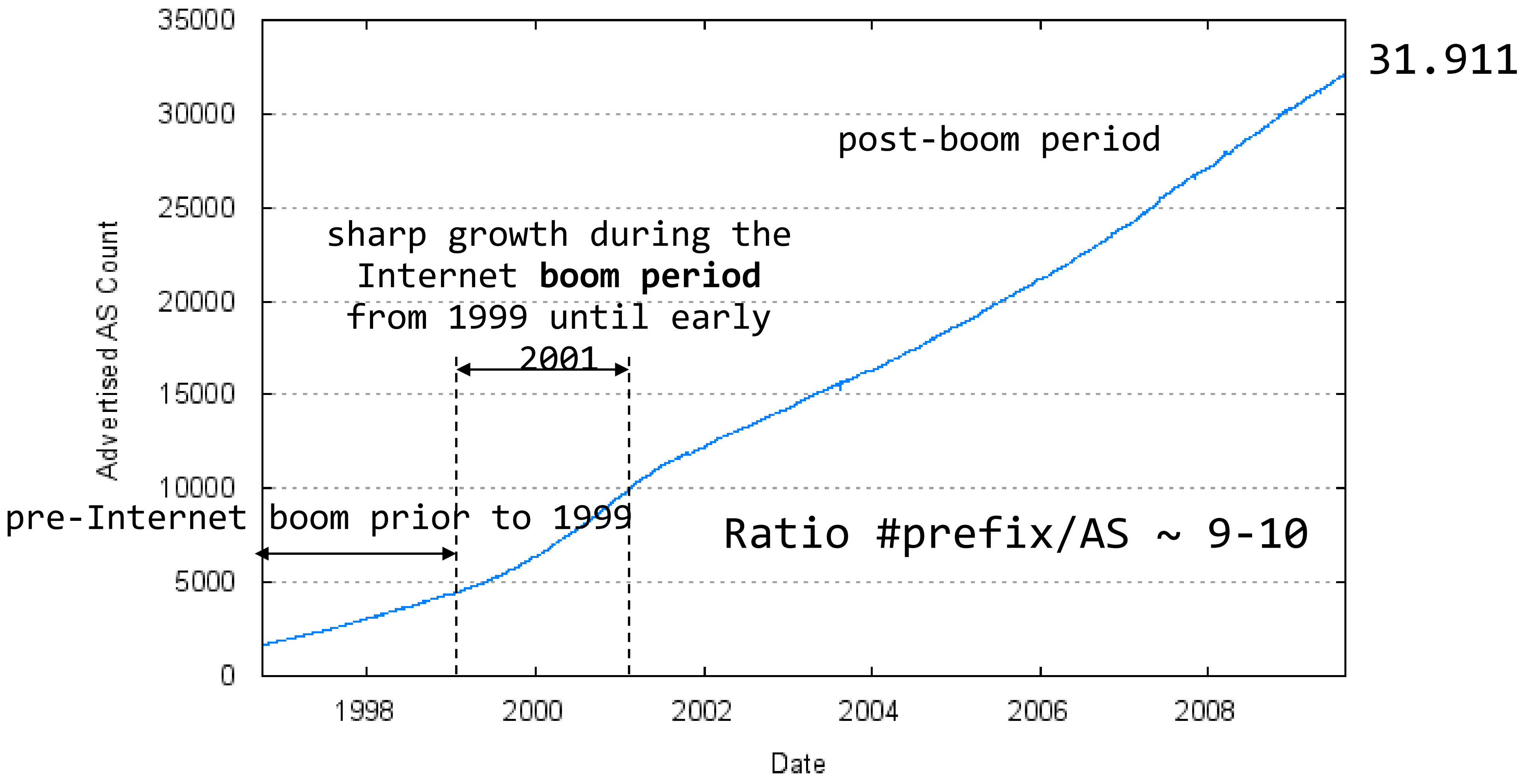
Growth of Active BGP Entries
(from Jan'89 to Aug'09)



Source: BGP Routing Table Analysis Reports - http://bgp.potaroo.net/index-bgp.html

Growth of number of AS
(from Jan'97 to Aug'09)

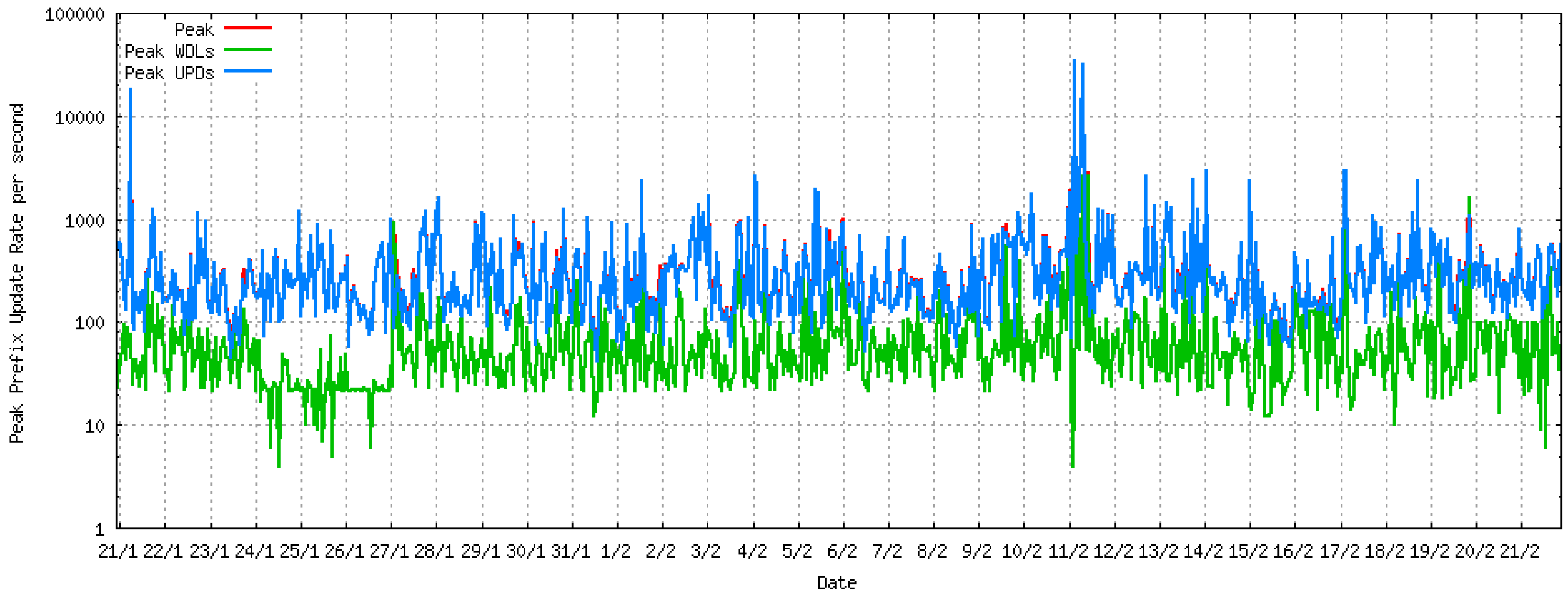
Number of unique ASN advertised in BGP routing table over Time



Source: http://www.potaroo.net/tools/asn32/

In practice...

- **Static**: DFZ routing tables
  - 300.000 prefix entries (growing at ~20-25% per year)
  - 30.000 ASs (growing ~15-20% per year)
- **Dynamics BGP updates** (routing convergence)
  - Average: 2-3 per sec. – Peak: O(1000) per sec.
  - BGP suffers from churn which increases load on BGP routers (due to link/nodes failures and traffic engineering)
  - BGP’s path vector amplifies these problems



BGP Scalability vs Convergence

RT size growth rate (~25% year) ⇒ routing engine system resource consumption *growth rate*

- Routing space size
  - ↑ #routing table entries ⇒ ↑ memory
  - ↑ #routing table entries ⇒ ↑ processing and searching (lookup)
- Number of peering adjacencies between routers
  - ↑ #peering adjacencies ⇒ ↑ memory (due to dynamics associated with routing information exchanges)

Impact on BGP convergence time

- BGP convergence time is limited by access speeds of DRAM (used for RIB storage)
  - DRAM capacity growth rate: ~4x every 3.3 years
  - DRAM access speed growth rate: ~1.2x every 2 years
- BGP convergence time degradation rate (estimation):

$$\frac{\textit{RT growth rate}\ [1.25]}{\text{DRAM access speed growth rate}\ [1.1]} \sim 10\%\ \text{per year}$$

Objectives
(Fundamental Dimensions)

Fundamental Dimensions

- **Routing scheme**

  Distributed algorithm that, given a destination node’s name, allows any node to route messages toward any destination node

- **Fundamental dimensions**
  - Distribution
  - Dynamicity
  - Stretch
  - Universality
  - Name independence
  - Policy
  - Security

  $\left.\begin{array}{l}\text{Policy}\\ \text{Security}\end{array}\right\}$ Not covered in this tutorial

Distribution

- **Source explicit routing**

  Each datagram carries in its header a complete description of the path along which datagram should be routed

  Local router does only need to maintain local routing information e.g. the address of its neighbors (next-hop)

  Drawback: datagram headers need to be of size (n) and still be processed at wire speed

- **Each node stores a complete Routing Table (RT)**

  Each node can perform an independent routing decision as RT include an entry for any destination: the next-hop/link to which packets for that destination should be forwarded

  Drawback: resulting RT size require to store $\Omega(n \log(n))$ bits of routing information where n is the number of nodes

Dynamicity

- **Updatefull routing**: routing in the light (with non-local knowledge about network topology)
  - Dynamics of routing information exchanges resulting from
    - Network topology updates (dynamic reaction to topological changes due to e.g. link/node failures)
    - Routing information updates (impacts number of inter-domain routing messages exchanged among BGP routers)
  - Updates requiring RT recalculation can lead to convergence delay, instabilities, and processing overhead
  - Communication cost lower bound for scale-free graphs is at best linear (up to logarithmic factors)

- **Updateless routing**: routing in the dark (w/o non-local knowledge about network topology)
  - Do not require to exchange routing updates
  - Do not maintain non-local knowledge about observed network topology for routing state to converge

Stretch

- **Definition**

  Stretch (of routing scheme): max. ratio over all source-destination pairs between the path length (cost) as produced by the routing scheme and the minimum length (cost) for the same source-destination pair

  – Algorithm stretch: maximum of this ratio among all node pairs (source, dest.) *in all graphs*

  – Average stretch: average of this ratio on subset of graphs

  – Intuitively: stretch is a quality measure of increase of paths length (cost) produced by a routing scheme compared to the minimum path length (cost)

- **Shortest path routing** (AS-path: BGP/path-vector, or cost metric: OSPF/link-state): **stretch = 1**

  Shortest path routing is **incompressible** (lower = upper bound): $\Omega(n \log n)$ bits required to store routing information in routing table

Universality

- **Universal** (routing scheme works correctly and satisfies scaling bounds on all connected graphs) vs **Specialized** (if it does so only on some specific graph classes)
- Internet topology shows properties of **scale-free graphs**
  - Node degree (k) distribution: power law $P(k) \sim k^{-\gamma}$, with scaling index $\gamma = 2.254$ (fewer nodes with large degree and large number of nodes with low degree)
  - Clustering: percentage of 3-cycles among all connected node triplets in the entire graph (number of triangular sub-graphs)

    Strong clustering means large number of triangular sub-graphs
  - Node degree correlation: negative correlation between node's degree k and its nearest-neighbors average degree (disassortative mixing: high-degree nodes tend to connect with low-degree nodes and visa versa)
  - Average shortest path length between all pairs of nodes on the Internet is small just over 3 hops (average AS-path length ~ 3,4)
- Internet topology not a tree or a grid but ~ scale-free graph (average hop distance between nodes grows prop. to log(n))

Name Independence

- Name-dependent schemes
  - Node addresses (or labels) encode some topological information (labels thus cannot be arbitrary)
  - Terms “node address” or “locator” refer to topologically informative node label ($\rightarrow$ PA address)
  - Addressing follows topology (topology-aware addressing):
    - Label encodes topological information useful for routing
    - Packet carries the chosen dest. label in its header, and
    - Note: Topology change $\Rightarrow$ Node label change (renaming)

- Name-independent schemes
  - Node addresses assigned from topology independent, arbitrary addressing space (no structure)
  - Terms “node name” or “identifier” refer to name-independent, topologically agnostic node label ($\rightarrow$ PI address)
  - Addressing does not follow topology (topology-unaware addressing) and topology does not follow addressing

- Note: scalable name-independent schemes are highly desirable for Internet routing (site multi-homing, mobility, etc.)

New routing scheme...

1. Scalability (memory): RT size scaling better than $\Omega(n \log n)$ – sub-linear RT size scaling (at best $O(\log n)$)
2. Quality (stretch): bound length increase of paths as produced by routing scheme compared to shortest path (does not grow with network size)
3. Reliability: fast convergence upon topology changes while minimizing communication costs to maintain coherent non-local knowledge about network topology
4. Name-independent routing: accommodate node addresses/labels assigned independently of the topology (*otherwise* need to split locator and ID parts in addressing architecture)
5. Specialization: take benefit of Internet topology properties (scale-free graph)

n = number of (abstract) nodes

Compact Routing Theory

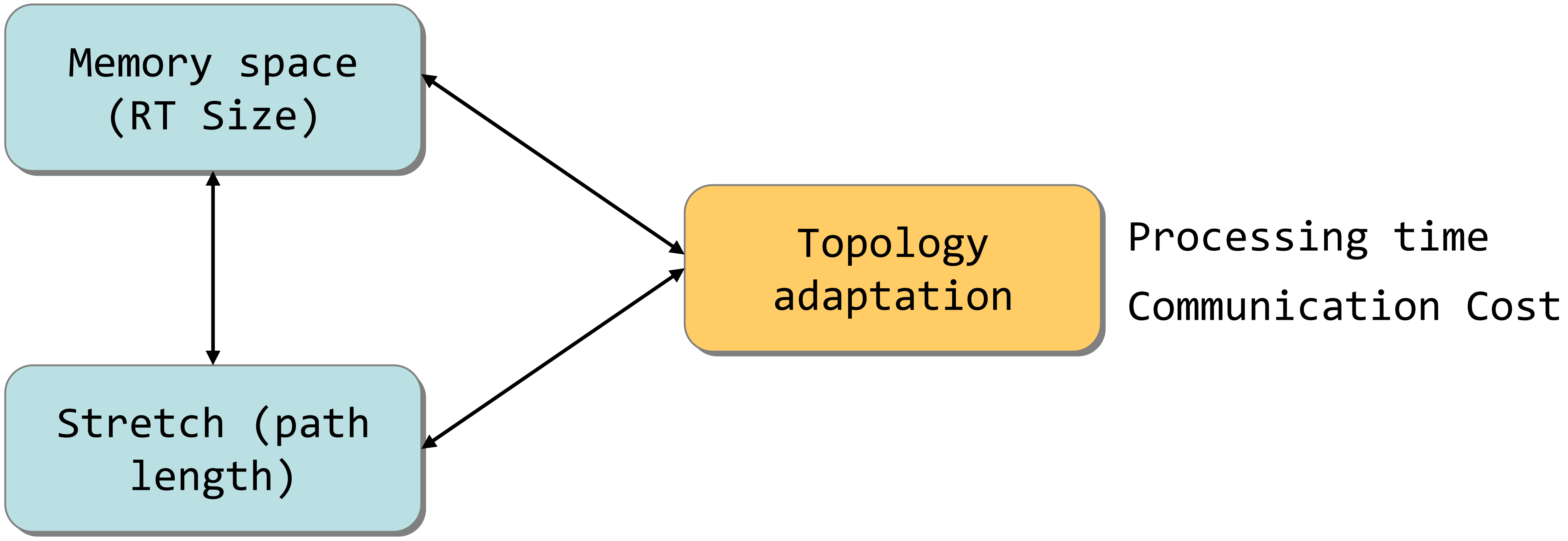
Definition

- A routing scheme is said to be compact if
  - Node names/labels and header sizes scales (poly-)logarithmically
  - Routing table size (local memory space in terms of number of bits) scales sublinearly
  - Stretch bounded by a constant (remains constant independently of the network size growth)

- Fundamental and unavoidable trade-off between stretch of routing algorithm and RT size it produces (local memory space: number of bits of routing information stored per node)

Fundamental Tradeoffs

- Static:



- Dynamic: topology adaptation
  - Communication cost: number of routing update per topology/policy change
  - Processing time: store → process (compute and possible add/remove, or replace RT entry) → forward

History of Compact Routing

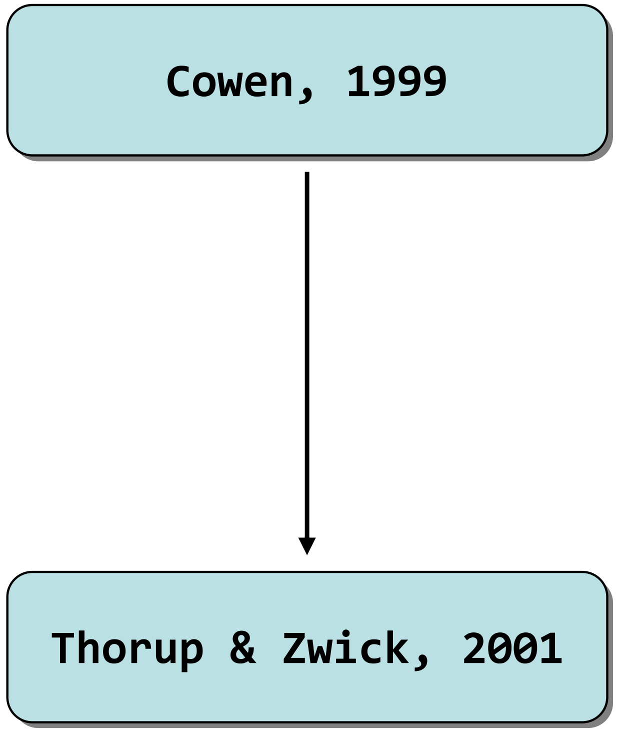
L.Cowen. *Compact routing with minimum stretch*. ACM SIAM SODA, 1999.

1) Non-hierarchical routing scheme (no network partitioning required)
2) Generic/universal (works for any topology)
3) Fixed stretch of 3 for any topology of any size
4) Sublinear RT size upper bound of $\tilde{O}(n^{2/3})$

M.Thorup and U.Zwick. *Compact routing schemes*. ACM SPAA, 2001.

TZ scheme: first generic ***nearly optimal*** routing scheme of stretch 3

1) Improves Cowen's RT size (memory space) upper bound to $\tilde{O}(n^{1/2})$
2) Fixed stretch of 3
3) Memory space upper and lower bounds nearly the same (up to a poly-logarithmic factor) NB: any routing with stretch below 5 cannot guarantee mem. space smaller than $\Omega(n^{1/2})$



History of Compact Routing

Main limitation of name-dependent scheme

When routing on topology-dependent labels (that encode topology-sensitive information useful for routing), topology change ⇒ label change: nodes needs to be relabeled

⇒ **Name-independent scheme** (routing on topology independent / arbitrary ID)

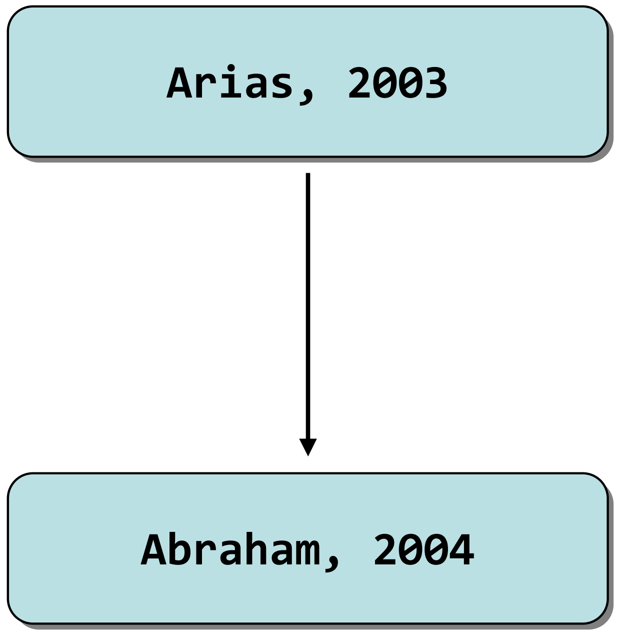
History of Compact Routing

M.Arias, L.Cowen, K.A.Laing, R.Rajaraman, and O.Taka. *Compact routing with name independence*. ACM SPAA, 2003.

1) Name-independent compact routing scheme
2) Memory space upper bound of $\tilde{O}(n^{1/2})$
3) Fixed **stretch of 5**

I.Abraham, C.Gavoille, D.Malkhi, N.Nisan, and M.Thorup. *Compact name-independent routing with minimum stretch*. ACM SPAA, 2004.

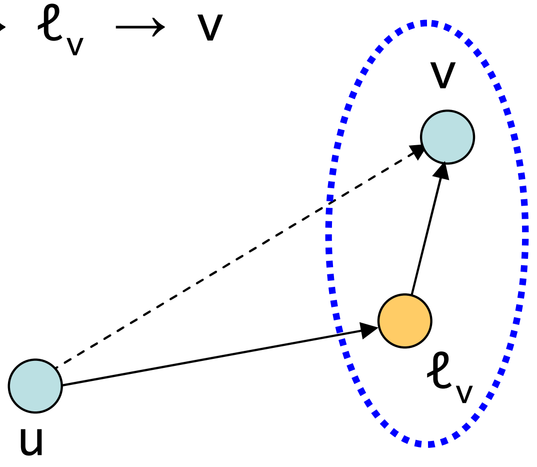
1) First generic nearly optimal *name-independent compact routing scheme*
2) Memory space upper bound of $\tilde{O}(n^{1/2})$
3) Fixed **stretch of 3**



Compact Routing Scheme "Principle"

- Within local neighborhood: store (and use) shortest path routing information
- Outside local neighborhood:
  - Store routing information to all landmarks
  - Route through a destination's closest landmark, and then use tree-routing from landmark to destination
- Stretch bound follows from the triangle inequality:

Routing: If $v \in B_u$, then route $u \to v$, else $u \to \ell_v \to v$



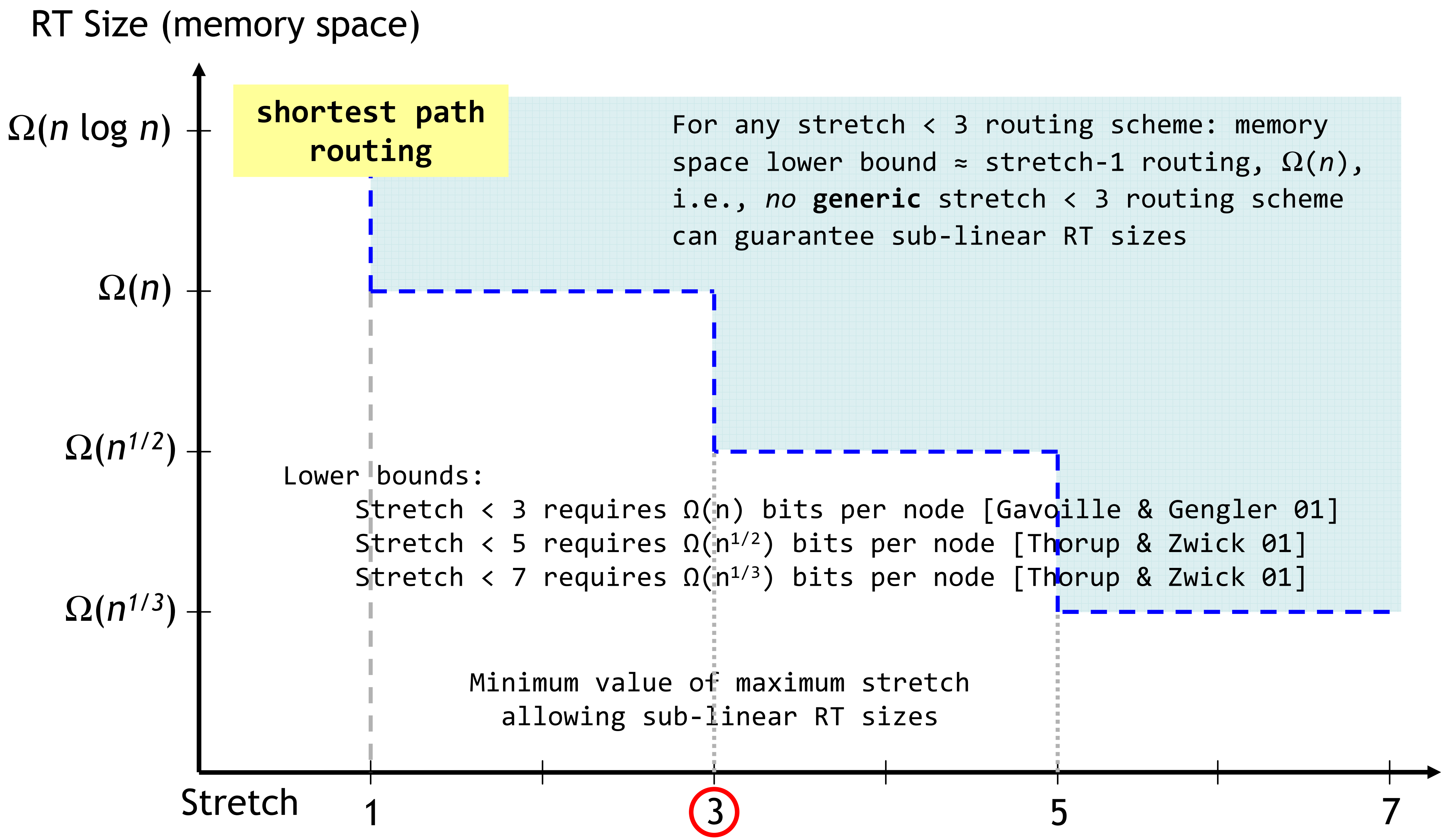
Suppose $d(\ell_v,v) \le d(u,v)$

Then $d(\ell_v,v) + d(u,\ell_v) \le d(u,v) + d(u,\ell_v)$

and $d(u,\ell_v) \le d(u,v) + d(v,\ell_v) \le 2\, d(u,v)$

Then $d(u,\ell_v) + d(\ell_v,v) \le d(u,v) + d(u,v) + d(v,\ell_v) \le \mathbf{3}\, d(u,v)$

Routing Scheme: Design Space



For any stretch < 3 routing scheme: memory space lower bound ≈ stretch-1 routing, $\Omega(n)$, i.e., *no* **generic** stretch < 3 routing scheme can guarantee sub-linear RT sizes

Lower bounds:

- Stretch < 3 requires $\Omega(n)$ bits per node [Gavoille & Gengler 01]
- Stretch < 5 requires $\Omega(n^{1/2})$ bits per node [Thorup & Zwick 01]
- Stretch < 7 requires $\Omega(n^{1/3})$ bits per node [Thorup & Zwick 01]

Minimum value of maximum stretch allowing sub-linear RT sizes

Routing Scheme: Design Space

- Name-independent routing: for k=1,2,3,5

  Stretch < 2k + 1

  Memory space: $\Omega(n^{1/k})$
- Labeled routing:

  Stretch ≤ 4k – 5, k ≥ 2 (≤ 2k – 1 with handshaking)

  Memory space: $\tilde{O}(n^{1/k})$

  Note: for k=4, stretch lower bound is not known to be 2k – 1 (Erdos conjecture proved only for k=1,2,3,5)

Stretch	Routing Table Size	Handshaking required?
3	$\tilde{O}(n^{1/2})$	no
5	$\tilde{O}(n^{1/3})$	yes
7	$\tilde{O}(n^{1/3})$	no
2k-1	$\tilde{O}(n^{1/k})$	yes
4k-5	$\tilde{O}(n^{1/k})$	no

Scaling Dependency on Topology (1)

- Properties of scale-free networks
  - They do not allow for efficient address aggregation like trees
  - But they do allow for extremely efficient *static* compact routing
- Scaling dependency on topology
  - Effectiveness of hierarchical network partitioning and aggregation depends
    - either of profusion of nodes at long hop-distances from each other
    - or on strong regularity of tree structure

  … but none are present in scale-free graphs

Scaling Dependency on Topology (2)

- Consequence 1: efficient application of hierarchical, aggregation-based routing to Internet-like topologies is hopeless
  - Hierarchical routing performs well for graphs with large distances between nodes when average distance d growths polynomially with network size (n):

    d(n) ~ n exp(m), m>0
  - Internet routing (average AS-path length ~3,4) average distance d growths at most logarithmically with network size (n):

    d(n) ~ log(n)

  - <u>Note</u>: applying hierarchical aggregation-based routing to an Internet AS-level topology would incur about 15-times AS-path length increase

Scaling Dependency on Topology (3)

- Consequence 2: routing protocols relying (explicitly or implicitly) on aggregation of topology-dependent locators (name-dependent routing) can not improve RT size scaling on Internet topology (example Loc/ID split) because
  - Such aggregation is impossible on scale-free topologies
  - Even when aggressive aggregation is not used, scaling can not be improved

    Reason: in addition to maintaining and updating RTs of topology-dependent locators, a distributed dictionary of identifier-to-locator mappings (for identifier to-locator name-resolution) must also be maintained and updated

Specialized Compact Routing (1)

- Universal compact routing (for any connected graph): stretch-1 with sub-linear RT size (memory space) does not exist

  ⇒ Specialized compact routing scheme: find "shortest path routing" with sub-linear RT sizes on scale-free graphs

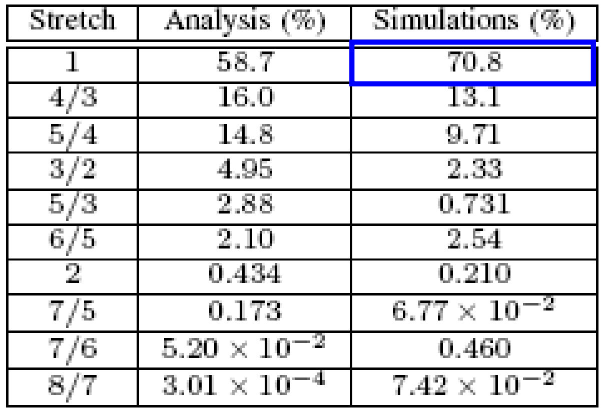
- Expectation: take benefit of scale-free networks specifics to obtain "non-generic algorithms" specialized (optimized) for Internet-like topologies

Specialized Compact Routing (2)

- Universal (any connected graph) => Specialized compact routing schemes (scale-free graph)
  - Average stretch on Internet-like graphs lower than 3 but how close can it be to 1?
  - [Krioukov04] showed that average performance of optimal stretch-3 TZ scheme on Internet-like topologies is much better than its worst case
    - Upper bounds are around 2200 for RT size and stretch 3
    - Average RT size on Internet topology around 50 entries
    - Average stretch = 1.1

Up to 70% of all pairwise paths being stretch-1 (shortest path)

On scale-free graphs, both stretch and memory can be made extremely small simultaneously



Stretch	Analysis (%)	Simulations (%)
1	58.7	70.8
4/3	16.0	13.1
5/4	14.8	9.71
3/2	4.95	2.33
5/3	2.88	0.731
6/5	2.10	2.54
2	0.434	0.210
7/5	0.173	6.77×10^{-2}
7/6	5.20×10^{-2}	0.460
8/7	3.01×10^{-4}	7.42×10^{-2}

D.Krioukov, K.Fall, and X.Yang, *Compact routing on Internet-like graphs*, IEEE INFOCOM 2004.

Compact Routing Schemes

Main Compact Routing Schemes

- TZ name-dependent universal scheme (2001)
- BC name-dependent specialized scheme (2006)
- Abraham name-independent universal scheme (2004)

TZ Scheme: Compact name-dependent routing scheme

- Construction:
  - Choose a set of landmarks (centers)
    - Landmark set construction: iterations of random selections of nodes to guarantee right balance between neighborhood size ($O(n^{1/2})$) and Landmark set size ($O(n^{1/2})$)
  - Clusters (of node v): set of vertices that are closer to v than to all landmarks
    - "Local" cluster (of node v) is a set of nodes closer to v than to neighbor’s closest landmarks

TZ Scheme: Compact name-dependent routing scheme

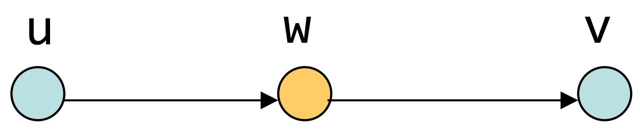
Formalism

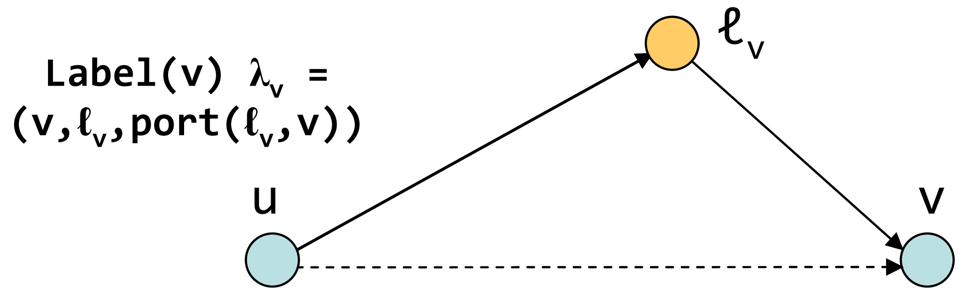
- $G = (V,E)$, $n = |V|$ and $m = |E|$, undirected graph with positive edge weights assigned to its edges
- For $u, v \in V$, $d(u,v)$: (weighted) distance between u and v in G
- Set $L \subseteq V$: subset of vertices referred to as landmarks with $d(L,v) = \min \{d(u,v) \mid u \in L\}$
- For every $w \in V$, cluster of w wrt set L:

  $B_L(w) = \{v \in V \mid d(w,v) < d(L,v)\}$
  - Note: clusters belonging to different vertices are not necessarily disjoint
- Center (landmark) ℓ_v: vertex from L nearest to v

TZ Scheme: Compact name-dependent routing scheme

- Routing table: shortest paths to local cluster nodes and all landmarks
  - RT includes original node ID, its closest landmark ID, the ID of the closest landmark's port lying on the shortest path from the landmark to the node



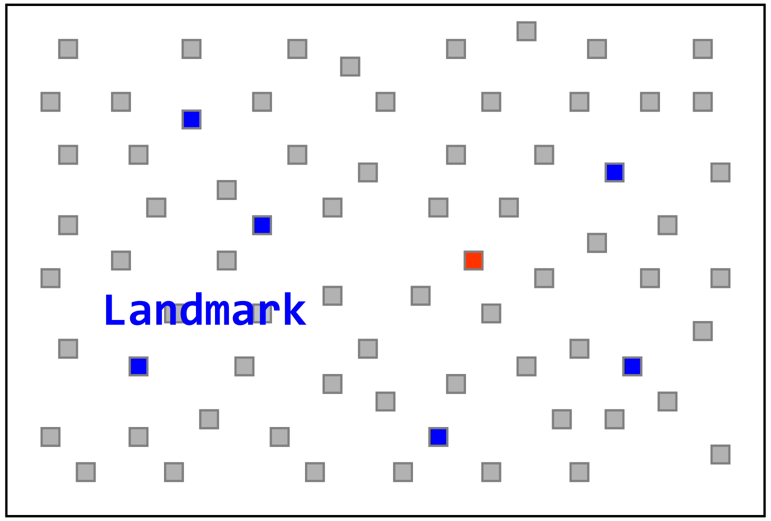


- Routing from u to v (at w)
  - **If** v belongs to local cluster of node u ($v \in B_L(u)$)

    Then route directly using shortest path to v ($\forall$ node w on the shortest path from $u \rightarrow v$, $v \in B_L(w)$)
  - **If** v does not belong to local cluster of node u ($v \notin B_L(u)$)

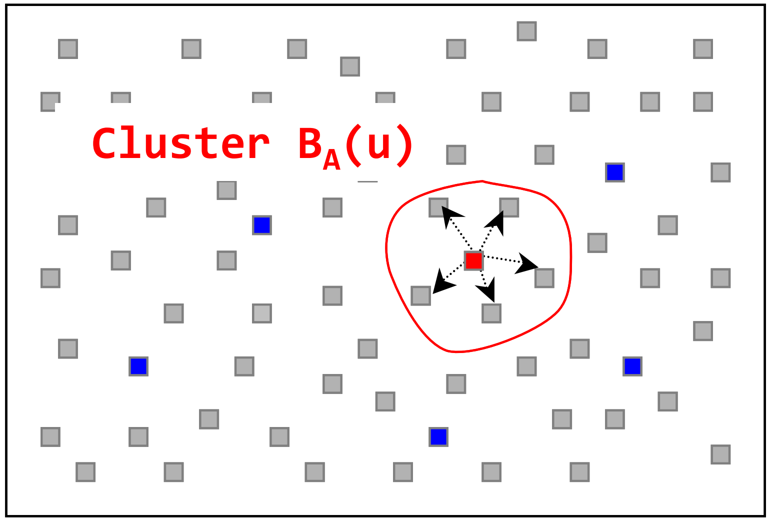
    Then route indirectly via landmark of node v (shortest path from u to ℓ_v, and then a shortest path from ℓ_v to v)

TZ Scheme: Compact name-dependent routing scheme

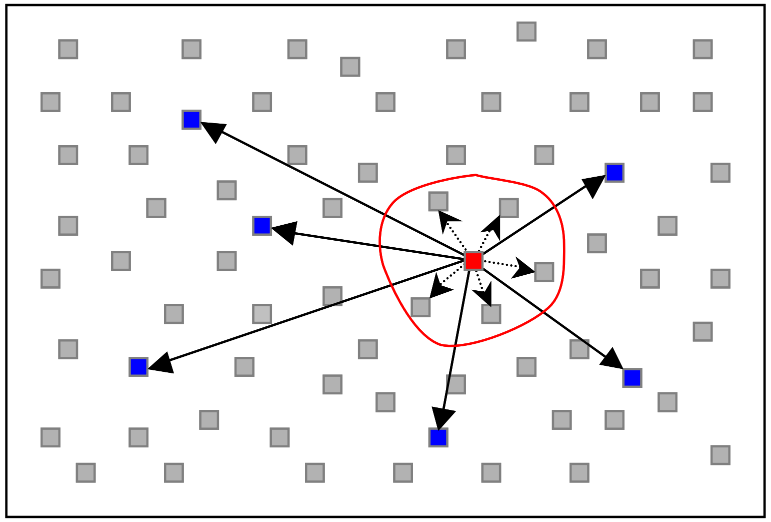


Choose a set of landmarks A

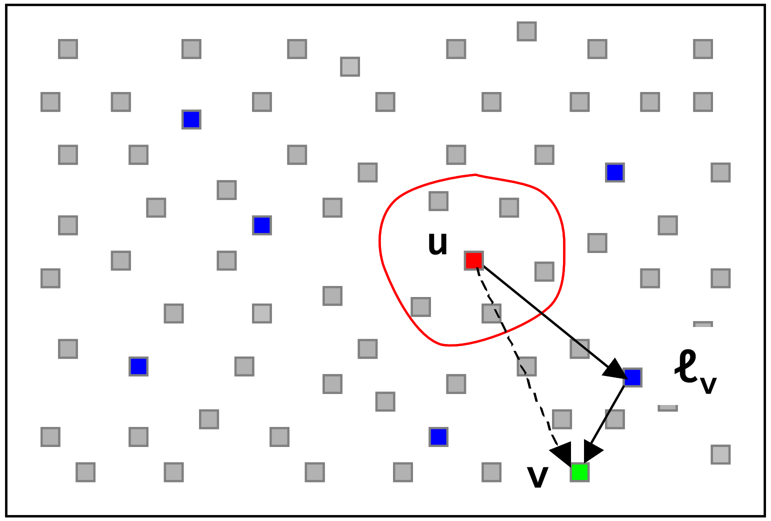
ℓ_u landmark closest to u



$B_A(u)$: set of vertices that are closer to u than to all other landmarks



u keeps routing information from u to $B_A(u)$ –local- and A –remote-



As $v \notin B_A(u)$, route indirectly from u -> ℓ_v (shortest path) then from ℓ_v to v (shortest path)

TZ Scheme: Compact name-dependent routing scheme

- How to choose the landmarks ($\ell \in L$) ?
  - Set L such that $|L| = \tilde{O}(n^{1/2})$
  - For $\forall$ $v \in V$, cluster of v, $|B_A(v)| = \tilde{O}(n^{1/2})$
- [Cowen99]: every weighted undirected graph has a set $L \subseteq V$ of size $\tilde{O}(n^{2/3})$ such that $|B_A(v)| = \tilde{O}(n^{2/3})$, for $\forall$ $v \in V$

  => Stretch 3 routing scheme that uses only $\tilde{O}(n^{2/3})$ memory at each vertex
- TZ scheme improves Cowen's construction: every weighted undirected graph has a set of landmarks $L \subseteq V$ such that $|L| = \tilde{O}(n^{1/2})$ and $|B_A(v)| = \tilde{O}(n^{1/2})$, for $\forall$ $v \in V$

TZ Scheme: Compact name-dependent routing scheme

```
Algorithm landmark(G,s)
$L \leftarrow \emptyset$;  $W \leftarrow V$;
while $W \neq \emptyset$ do
{
    $L \leftarrow L \cup$ sample(W,s);
    $B(w) \leftarrow \{ v \in V \mid d(w,v) < d(L,v)\}$, for $\forall w \in V$;
    $W \leftarrow \{ w \in V \mid |B(w)| > 4n/s \}$;
}
return L;
```

Subroutine sample(W,s) receives set W and returns a random subset of W obtained by selecting each element, independently, with probability s/|W|

If $|W| \leq s$, then **sample**(W,s) returns the set W itself

If $|W| \geq s$, the expected size of the sample is s

Theorem: the expected size of set L returned by algorithm landmark(G,s) is at most 2s log n

For $\forall\ w \in V$, $|B(w)| \leq 4n/s$ $(1 \leq s \leq n)$

-> Expected size of L is $O(n^{1/2}\log n)$ i.e. $\tilde{O}(n^{1/2})$

TZ Scheme: Compact name-dependent routing scheme

- TZ scheme performs optimally on scale-free graphs (Internet topologies)
  - Scale-free graphs yield best possible performance of the TZ scheme (compared to all other graphs)
  - First indicator that scale-free topologies are particularly ‘well-structured’ and allow for better routing performance

Routing Scheme	**Stretch**	**Memory**
Shortest-path routing	1	O(n log n)
TZ scheme (universal)	3	$\tilde{O}(n^{1/2})$
TZ scheme (scale-free)	shows an average stretch ~ 1.1 (~70% shortest path)	Small RT size (around 50 entries for O(10k) node networks (well below theoretical upper bounds)

⇒ Construction of name-dependent compact routing schemes specialized to utilize structural peculiarities of scale-free graphs (~Internet topology)

BC Scheme: first compact routing scheme specialized for scale-free graphs

- BC scheme achieves logarithmic RT scaling $O[(\log_2 n)^2]$ and infinitesimally small stretch on scale-free graphs
- <u>Idea</u>:
  - Build one shortest-path tree rooted at the highest-degree node and a small fixed number of additional trees to cover edges at the periphery of a power-law graph
  - Routing is then confined to this tree collection

BC Scheme: first compact routing scheme specialized for scale-free graphs

- Principle:
  - Step_1: Find the highest-degree node in the graph and grow the shortest path tree rooted at it
  - Step_2: Find the core, which is all nodes located within maximum distance (d) from the highest degree node
  - Step_3: Grow small number (e) of trees to cover all the edges that do not belong to the main tree and lie outside of the core (the larger d, the smaller e)
  - Use known compact routing algorithms to route on these trees

$\rightarrow$ Since RT sizes scale logarithmically, BC scheme achieves logarithmic scaling

Compact name-independent routing with minimum stretch

- Also referred to as "Abraham scheme"
  - Main references
    - I.Abraham, C.Gavoille, D.Malkhi, N.Nisan, and M.Thorup. *Compact name-independent routing with minimum stretch*. ACM SPAA, 2004
    - I.Abraham, C.Gavoille, D.Malkhi, N.Nisan, and M.Thorup. *Compact name-independent routing with minimum stretch*. ACM TALG, 2008
- Steps and mechanisms toward stretch 3 scheme
  - ▶ Vicinity balls
  - ▶ Coloring
  - ▶ Hashing names to colors
  - ▶ Labeled routing on trees
  - ▶ Landmarks
  - ▶ Partial shortest path trees

} Stretch 3 scheme

Preliminaries (1)

- V: set of n nodes labeled with an arbitrary unique identifier that can be represented by O(log n) bits

- Graph G = (V, E, ω) with positive edge cost ω. For u, v ∈ V,
  - d(u,v): minimum cost path from u to v in G

    where path cost = sum of the weights along its edges
  - Each node has, for each outgoing edge, a unique port name from the set of integers {1,...,n} (fixed-port model)

- Initially: source only knows the name of the destination node

  This destination is written in the message header

- Scheme requires writable packet headers: routing algorithm allowed to write a reasonable amount of information into message headers as they are routed

  Note: $O(\log^2 n / \log\log n)$-bit headers

Preliminaries (2)

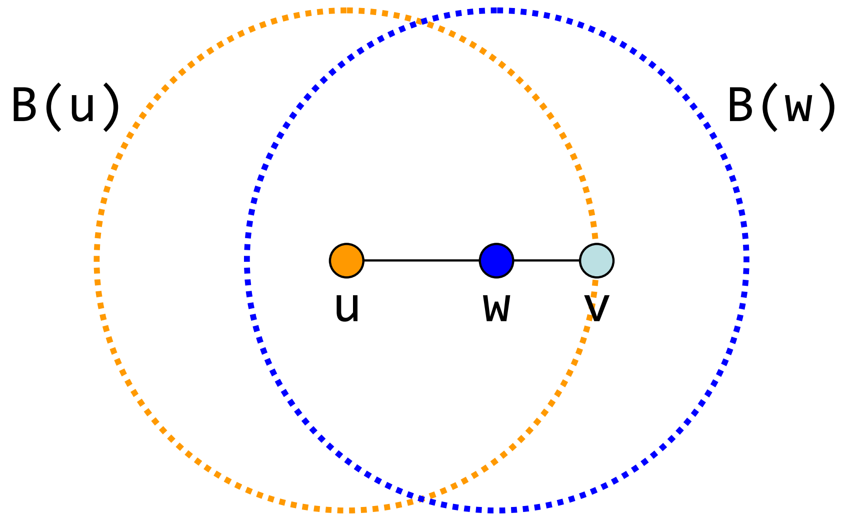
- Use of headers aims at useful tradeoffs between current techniques (source-directed routing, and routing with a fixed header) resulting in scaling problems
  - Writing a small amount of information in the header can significantly reduce the amount of information needed at the routers

- Abraham scheme (and all previous name-independent schemes) use writable packet headers
  - A scheme that does not rewrite packet headers must be loop free and thus must have stretch 1 on any tree (in a tree that is a star the center would have to code a permutation using $\Omega(n \log n)$ bits on the average)

Vicinity Balls (1)

- $\forall$ κ ≥ 1, and for a node u ∈ V, vicinity of u, $B_\kappa(u)$: set consisting of node u and κ closest nodes to u

- Vicinities satisfy monotonicity property [Awerbuch90]:
  If v ∈ $B_\kappa(u)$ and w is on a minimum cost path from u to v, then v ∈ $B_\kappa(w)$

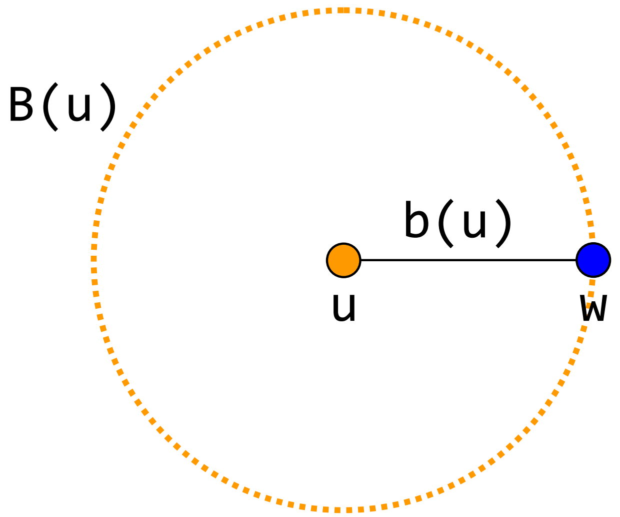


- Size of the vicinities set to κ = [4 √n log n]

Vicinity Balls (2)

Notation:

- $B(u) = B_K(u)$
- $b(u)$: radius of $B(u)$ with $b(u) = \max_{w \in B(u)} d(u,w)$

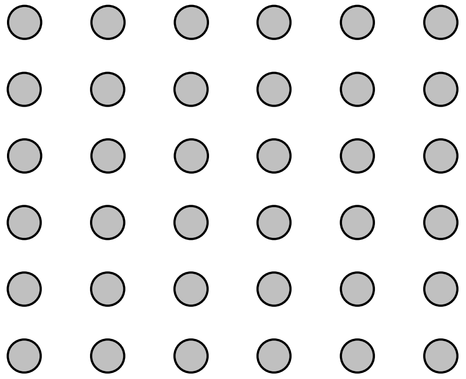


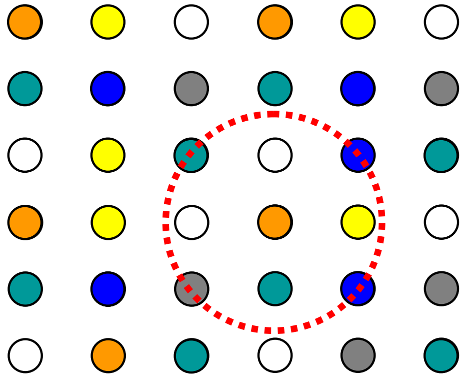
Assumptions:

- Each node u knows its vicinity B(u)
- Node u has access to standard dictionary over the names in B(u) so that (in constant time) u can check membership and look up associated information

Coloring (1)

- Partition nodes into color-sets $C_1,\ldots,C_{\sqrt{n}}$, with the following two properties
  1. Every color-set has at most $2\sqrt{n}$ nodes
  2. Every node has in its vicinity at least one node from every color-set
- If node $u \in C_i$, it has "color i", $c(u) = i$





- Constructing a polynomial-time coloring satisfying these properties

Coloring (2)

- By standard Chernoff bounds and union bound Properties holds with high probability
  - If Every node u independently chooses a random color c(u)
  - Then with high probability

    Every color set has $O(\sqrt{n})$ nodes

    Every node has in its vicinity at least one node from every color set

- Constructing a polynomial-time coloring satisfying these properties
  - Derandomization via method of conditional probabilites using pessimistic estimators can be computed in $\tilde{O}(n^2)$

Hashing Names to Colors (1)

- Assume mapping function h from node names to colors that is balanced in the sense that at most $O(\sqrt{n})$ names map to the same color

- Each node u should be able to compute h(w) for any destination w

- Note: if names were a permutation of {1,..,n}, one could just extract ½ log n bits from the name, but here arbitrary names such as IP addresses

Hashing Names to Colors (2)

- Function h that can be computed in constant time from n arbitrary names to $\sqrt{n}$ colors
  - Constant time assumes performing standard arithmetic operations on names in constant time

    => Each name is stored in a constant number of words
  - Representation of function h takes $O(\sqrt{n})$ space

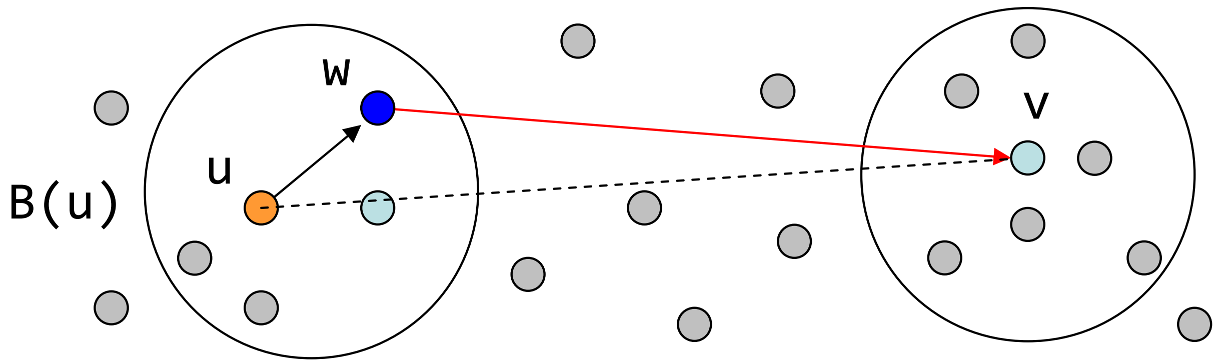
    => Representation can be stored at each node without violating space bounds
  - With the constant time hash function, each routing decision is made in constant time

Example: stretch-3 for complete graphs

- Simple stretch 3 scheme with Õ(√n) bits per node given a complete graph whose edge weights satisfy the triangle inequality
- Every node u stores the following information:

  (1) Names of all nodes in the vicinity B(u) and what port number to use to reach them

  (2) Names of all nodes v such that c(u) = h(v) and what port number to use to reach them



- Routing from u to v is done in the following manner:

  If v ∊ B(u) or c(u) = h(v)

  Then u routes directly to v with stretch 1

  Otherwise u forwards the packet to w ∊ B(u) such that c(w) = h(v)

  Then from w the packet goes directly to v

Example: stretch-3 for complete graphs

Stretch is at most 3

- Suppose $d(u,w) \leq d(u,v)$

  Then $d(u,w) + \underline{d(w,v)} \leq d(u,v) + \underline{d(w,v)}$
- TI: $d(w,v) \leq d(w,u) + d(u,v)$

  Then $d(u,w) + d(w,v) \leq \underset{\downarrow}{\boxed{d(u,v) + \underline{d(u,v)}}}\underline{\;+ d(w,u)}$

  and $d(u,w) + d(w,v) \leq 2d(u,v) + \underline{d(w,u)}$

  with $d(u,w) \leq d(u,v)$
- Thus $d(u,w) + d(w,v) \leq 3d(u,v)$

Labeled Routing on Trees (1)

- Based on DFS Interval-based routing scheme improved by [Fraigniaud01] and [Thorup01]

<u>Principle</u>

- For every weighted tree T with n nodes (fixed-port model), there exists a labeled routing scheme that, given any destination label, routes optimally on T from any source to the destination
- Storage per node in tree T, size of label λ, and header size are $O(\log^2 n/(\log\log n))$ bits
- Given a tree T containing node v

  $\mu(T,v)$, stored routing information of node $v \in T$, and $\lambda(T,v)$, destination label of node v in tree T,

  routing decisions take constant time

Labeled Routing on Trees (2)

- A node is called "heavy" if its sub-tree contains more than half of the nodes of its parent’s sub-tree

- Each node stores i) its DFS interval and ii) the DFS interval of its heavy child (if it has one)

- A node’s label consists of the names of the non-heavy nodes on the path from the root

- Routing to v on node w:

  If v is not in w’s interval then send to parent

  If v is in w’s heavy child interval then send to heavy

  Otherwise, w’s label contains the appropriate child

Labeled Routing on Trees (3)

- Stretch 2k-1: each node participates in $\tilde{O}(n^{1/k})$ trees

  Stretch 2k-1, i.e, $\forall$ pair (u,v), there is a tree with a path of stretch at most 2k-1 between them

  In total, tree routing information per node is $\tilde{O}(n^{1/k})$

- Stretch 3 (k=2): each node participates in $\tilde{O}(n^{1/2})$ trees

  In total, tree routing information per node is $\tilde{O}(n^{1/2})$

Landmarks

- Designate one color to be special and call it the landmark color
- L: set of nodes ℓ such that their color $c(\ell)$ is the landmark color
- By coloring property
  - $|L| \leq 2\sqrt{n}$
  - $\forall\ v \in V,\ B(v) \cap L \neq \emptyset$
- For a node $v \in V$, ℓ_v: closest landmark node in $B(v)$ (breaking ties by lexicographical order of node names)

Partial Shortest-Path Trees

- For any node u, T(u): single source minimum-cost-path tree rooted at u
- In a partial shortest path tree, every node v maintains $\mu(T(u),v)$ iff $u \in B(v)$
  - Note: set of nodes that maintains $\mu(T(u),\cdot)$ is a subtree of T(u) that contains u
- If $x \in B(y)$, then given the label $\lambda(T(x),y)$, x can route to y along a minimum cost path
  - Proof follows monotonicity property [Awerbuch90]: For any node w along the minimum cost path of T(x) between x and y, node $x \in B(w)$ Thus, every node w on this path maintains $\mu(T(x),w)$

Stretch-3 Scheme (1)

Every node u stores:

- [Vicinity]

  For every node w ∈ B(u): name w, port name (u→y), y being next hop along a minimum cost path (MCP) from x to w

- [Landmarks]

  For every landmark ℓ ∈ L: routing information μ(T(ℓ),u) and label λ(T(ℓ),ℓ) of the tree T(ℓ)

- [Partial Shorthest Path Trees]

  For every node x ∈ B(u): routing information μ(T(x),u) of the tree T(x)

Stretch-3 Scheme (2)

Every node u stores: for every node v such that $c(u)=h(v)$, either option i) or ii) that produces MCP

i. Labels $\{\lambda(T(\ell_v),\ell_v), \lambda(T(\ell_v),v)\}$ where ℓ_v closest landmark node in B(v)

   Routing path

   - $u \rightarrow \ell_v \in B(v)$ using $\lambda(T(\ell_v),\ell_v)$ along the tree $T(\ell_v)$
   - and $\ell_v \rightarrow v$ using $\lambda(T(\ell_v),v)$ along the same tree $T(\ell_v)$

ii. p(u,w,v): path from u to v composed of MCP from u to w and MCP from w to v with the following properties

   - u and edge (x,y) exists along the MCP from w to v such that $x \in B(w)$ and $y \in B(v)$
   - if such paths exist, choose the lowest cost path p(u,w,v) and store labels $\{\lambda(T(u),w), x, (x\rightarrow y), \lambda(T(y),v)\}$

   Routing path

   - $u \in B(w) \rightarrow w$ on T(u) using $\lambda(T(u),w)$ – Partial SPT
   - $w \rightarrow x \in B(w) \rightarrow y$ using stored port number $(x \rightarrow y)$
   - $y \in B(v) \rightarrow v$ on T(y) using $\lambda(T(y),v)$ – Partial SPT

Stretch-3 Scheme (3)

- Routing from u to v using u routing information
  - Case 1a) If $v \in B(u)$: vicinity routing
    - name v, port name $(u \rightarrow w)$, w next hop along a minimum cost path (MCP) from u to v
  - Case 1b) If $v \in L$ (v is a landmark node): tree routing on T(v) using $\mu(T(v),u)$ and $\lambda(T(v),v)$ of tree T(v)

If $v \in B(u)$ or $v \in L$, u routes on minimum cost path (MCP) directly to v

Stretch-3 Scheme (4)

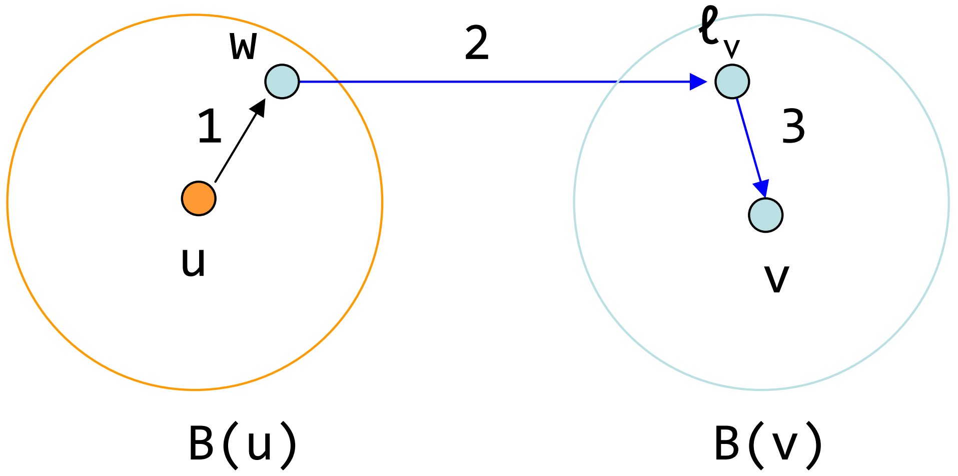
Otherwise, u forwards packet to $w \in B(u)$ such that $c(w)=h(v)$ using vicinity routing

Then, from w the packet is forwarded to v using w routing information

- Case 2a) uses w stored Labels $\{\lambda(T(\ell_v),\ell_v),\lambda(T(\ell_v),v)\}$

  where ℓ_v closest landmark node in B(v)

  WHEN on every MCP from u to v, there is a node y | $y \notin B(u)$ and $y \notin B(v)$ – note: $b(u) + b(v) \leq d(u,v)$



Routing path

1. $u \rightarrow w \in B(u)$ (vicinity routing) s.t. $c(w)=h(v)$

w stored labels $\{\lambda(T(\ell_v),\ell_v), \lambda(T(\ell_v),v)\}$

2. $w \rightarrow \ell_v \in B(v)$ using $\lambda(T(\ell_v),\ell_v)$ along tree $T(\ell_v)$
3. $\ell_v \rightarrow v$ using $\lambda(T(\ell_v),v)$ along the same tree $T(\ell_v)$

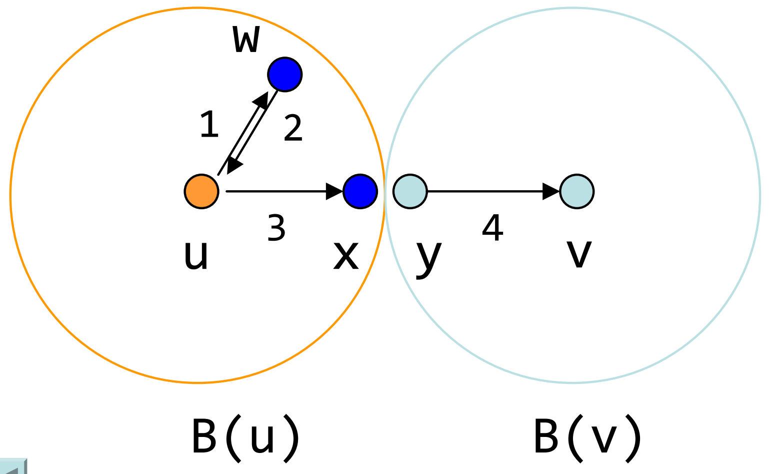
Stretch-3 Scheme (5)

Otherwise, u forwards packet to w ∈ B(u) such that c(w)=h(v) using vicinity routing

Then, from w the packet is forwarded to v using w routing information

- Case 2b) uses w stored labels {λ(T(u),w), x, (x->y), λ(T(y),v)}

  WHEN there exists a MCP in which every node is in B(u) ∪ B(v)



Routing path

1. u → w ∈ B(u) (vicinity routing) s.t. c(w)=h(v)
2. w → u on T(w) using λ(T(w),u) – Partial SPT routing
3. u → x ∈ B(u) → y using stored port number (x->y)
4. y ∈ B(v) → v on T(y) using λ(T(y),v) – Partial SPT routing

Dynamic Compact Routing

Dynamic Compact Routing (1)

- Dynamic nature of routing protocols allows each router to maintain non-local knowledge about topological changes (resulting from failures, addition/ removal of routes and ASs, etc.)
- This information is exchanged between routers by means of routing information updates (each router timely distributes to its own peers following specific selection criteria the routing information received from other peers)
- Routing updates lead to re-computation/re-selection of RT entries that in turn result in convergence delay, instabilities, and processing overhead

Dynamic Compact Routing (2)

- [Afek89] showed that communication cost cannot scale better than $\Omega(n)$

  Results applied to universal schemes

  => Expectation: specialized schemes for scale-free graphs might behave better ($\Omega(n)$ bound attained only on some marginal worst-case graphs) ?

- [Korman06] recently showed that communication cost lower bound for scale-free graphs is at best linear up to logarithmic factors: $\tilde{\Omega}(n)$

- Implication: as far as communication costs are concerned, Internet-like graphs belong to the class of worst-case graphs, across all possible network topologies

Demonstration

- Demonstrate that communication cost lower bound for scale-free graphs is $\tilde{\Omega}(n)$

Dynamic Compact Routing (3)

- Communication cost (number of routing update messages per topology change) of routing schemes cannot grow slower than linearly on Internet-like topologies
  - Timely distribution of updates to remote routers –upon topology and/or routing information changes– such that each router reaches and maintains a consistent view of the non-local network topology

- Combined with the cost resulting from processing routing updates for (re-)computation of RT entries (leading to convergence delay, instabilities, etc.)
  - Time complexity is defined as the number of cycles needed to (re-)compute a RT entry for a given destination and insert it as part of the RT (or replace/remove an existing entry in the RT)

Summary

Compact Routing (1)

- **Principle**: given coherent full view of the network topology build routing algorithm that balances efficiently fundamental (and unavoidable) trade-off between stretch and size of RT it produces (memory)

- **Compact routing scheme**:
  - Stretch bound by constant: does not grow with the network size at all
  - RT sizes scale sublinearly: at most o(n) bits of routing information stored per RT (per node)
  - Logarithmic address and header sizes

- Compact routing algorithms make routing table sizes compact by omitting "some" network topology details such that resulting path length increase stays small

Compact Routing (2)

- Name-dependent vs independent compact routing scheme

Stretch	Name dependent (labeled)	Name independent
1	Shortest-path: n log n	Shortest-path: n log n
3	Universal TZ scheme: $n^{1/2}$ log n	Universal Abraham scheme: $n^{1/2}$ log n Note: same upper-bound scaling as name-dependent scheme

- On Internet-like scale-free graphs

Stretch	Name dependent (labeled)	Name independent
1	Shortest-path: n log n	Shortest path: n log n
3	TZ scheme: shows an average stretch ~ 1.1 (~70% are shortest path)	Abraham scheme: shows an average stretch ~ 1.5 Note: name-independent scaling on Internet topologies is worse on average

- Specialized name-dependent schemes exploiting topological properties of scale-free graphs
  - Example: BC scheme: logarithmic RT scaling $O[(\log_2 n)^2]$ and infinitesimally small stretch)

Compact Routing (3)

… but compact routing schemes assumes a coherent full view of the graph (representing topology) at any given instant in time

-> Achieve coherent full view of the network topology (or paths to all other network nodes) and support network topology dynamics requires exchange of timely routing updates

Example: TZ scheme is not optimal for dynamic routing case since it labels nodes with topology-dependent information (as soon as the topology changes, nodes need to be re-labeled)

=> Associated **communication cost**:

Routing updates cannot grow slower than linearly with network size on scale-free graphs (... but still much better than exp.-like communication cost of "existing" routing schemes)

Compact Routing (4)

Routing "design" space

- **Evolutionary**: BGP re-considered (is it fundamentally possible ?), new candidate protocol like HLP (or similar), etc. but no improvement possible on RT size scale from aggregation
- **Disruptive**: dynamic compact name-independent routing schemes specialized for scale-free graphs

Bottom line

- Routing requires coherent full-view (network graph topology or path to destination) and support of topology dynamics $\Rightarrow$ Timely routing updates
- Routing updates communication cost and resulting processing cost cannot grow slower than linearly on Internet

Compact Routing (5)

Objective: ***find compromise between routing algorithm stretch, routing table size scaling (memory space), and dynamics (processing and communication cost)***

Construct in polynomial time a dynamic compact routing scheme that minimizes the stretch bound for Internet-like graph while

i. requiring at most o(n) bits of routing information per node (RT size scaling sublinearly)

ii. minimizing communication and processing costs

Open Research Challenges

Trends and Alternatives (1)

- Current trend in "inter-domain" routing system: short-term incremental fixes to attenuate symptoms instead of addressing root causes

- This approach works
  - Until system engineering can ‘absorb’ protocol deficiencies (stretch x RT size scale x dynamics)
  - Until network engineering can ‘absorb’ executions problems: operational workarounds to resolve policy, monitoring, etc.

  ... but is subject to **"unknown but limited elasticity"**

Trends and Alternatives (2)

- **Alternatives**
  - **Evolutionary**
    - Patch "a la CIDR": still to be invented (does it exist?)
    - Loc/ID split protocol: no improvement possible on RT size scale from aggregation
    - BGP x machine learning (x data/graph analysis and mining)
  - **Disruptive**
    - Construct in polynomial time a name-independent dynamic compact routing scheme that minimizes stretch bound for Internet-like graphs while requiring at most O(n) bits per RT
    - (Of course) other alternatives are possible !

Trends and Alternatives (3)

- Open coupled research challenges (stretch x RT size scale x dynamics) for new distributed routing schemes specialized for scale-free graphs

- Possible directions to address challenges of dynamic name-independent compact routing scheme
  1. Greedy routing (dual metrics)
  2. Minimize effects of topology adaptation (processing and communication cost)
     - Traffic-driven: spatial characteristics
     - Utility-driven: usefulness of disseminated routing updates (path backtracking, selective routing)

Trends and Alternatives (4)

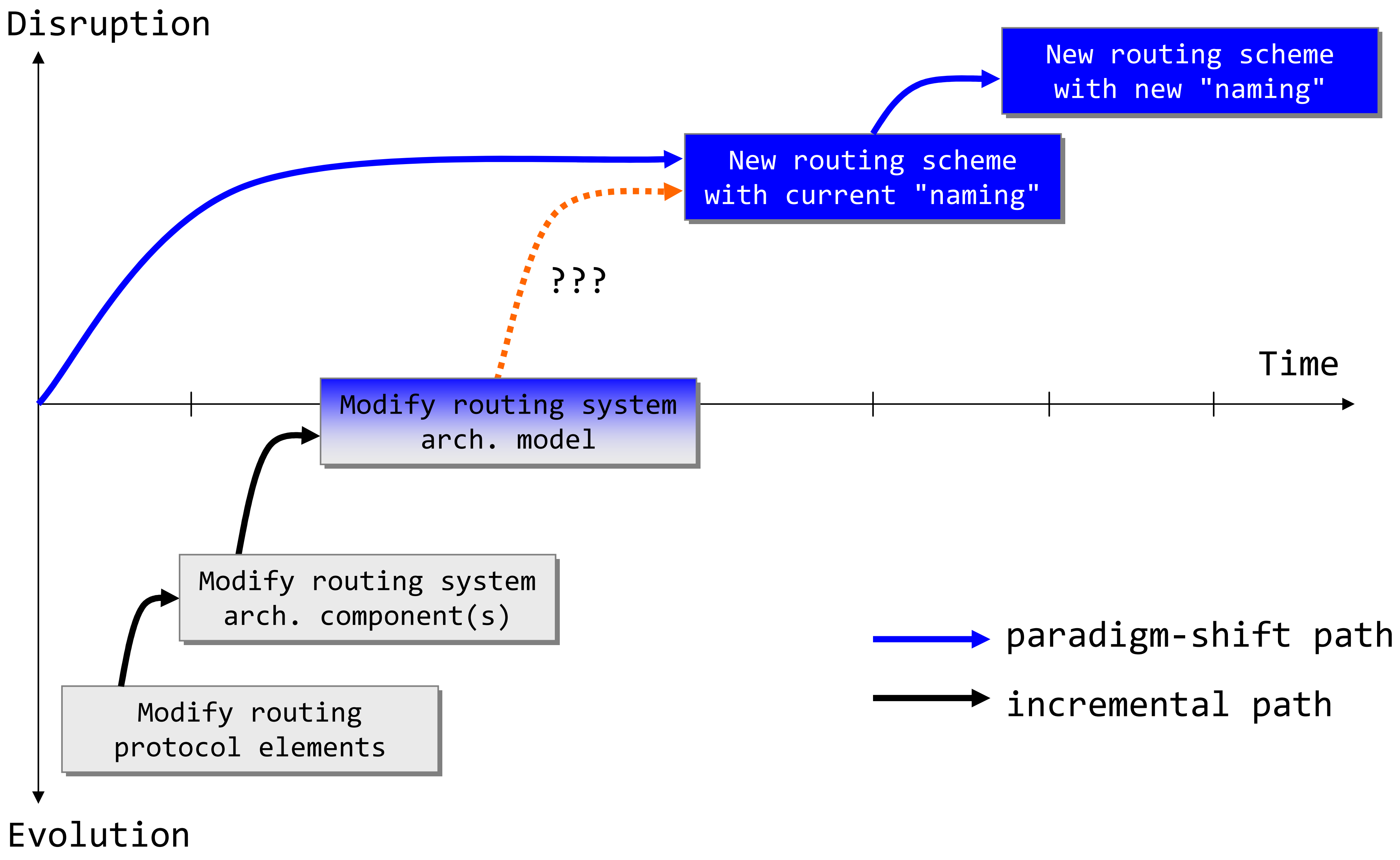
- Compact routing combined with
  - Policy
  - Security

  } unaddressed challenges

- AnyTraffic compact routing (for both unicast and multicast traffic)

- Performance evaluation (stretch x RT size x communication cost x processing time complexity) of dynamic routing schemes on Internet-like topologies is a challenge on its own (large-scale simulation)

Trends and Alternatives (5)



Disruption
New routing scheme with new "naming"
New routing scheme with current "naming"
???
Time
Modify routing system arch. model
Modify routing system arch. component(s)
Modify routing protocol elements
paradigm-shift path
incremental path
Evolution

References

References

- I.Abraham, C.Gavoille, D.Malkhi, N.Nisan, and M.Thorup, *Compact name-independent routing with minimum stretch*, In Proc. of ACM SPAA 2004.
- Y.Afek, E.Gafni, and M.Ricklin, *Upper and lower bounds for routing schemes in dynamic networks*, In Proc. of FOCS, 1989.
- M.Arias, L.Cowen, K.A.Laing, R.Rajaraman, and O.Taka, *Compact routing with name independence*, In Proc. of ACM SPAA 2003.
- A.Brady and L.Cowen, *Compact routing on power-law graphs with additive stretch*, In Proc. of ALENEX 2006.
- L.Cowen, *Compact routing with minimum stretch*, In Proc. of ACM SIAM SODA 1999.
- A.Korman and D.Peleg, *Dynamic routing schemes for general graphs*, In Proc. of ICALP 2006, Part I, LNCS 4051, pp. 619-630, Springer-Verlag Berlin Heidelberg 2006.
- D.Krioukov, K.Fall, and X.Yang, *Compact routing on Internet-like graphs*, In Proc. of IEEE INFOCOM 2004.
- D.Krioukov, and kc claffy, *Toward Compact Inter-domain Routing*, 2005.
- P.Pedroso, O.Pedrola, and D.Papadimitriou, *Anytraffic routing algorithm for label-based forwarding*, IEEE Globecom, Nov.2009.
- M.Thorup and U.Zwick, *Compact routing schemes*, In Proc. of ACM SPAA 2001.
- M.Thorup and U.Zwick, *Approximate distance oracles*, In Proc. of ACM STOC 2001.