Interdomain Routing Design for MobilityFirst

October 6, 2011

Z. Morley Mao, University of Michigan
In collaboration with Mike Reiter's group

Interdomain routing design requirements

- Mobility support
  - Network mobility and disconnection
- Efficiency and scalability
  - Reduced routing table size
  - Multi-homing of devices across multiple networks
  - Multipath delivery across multiple networks
  - Efficient multicast to destinations in different networks
  - Edge network resource awareness (BW, storage, availability, etc.)
- Security
  - Defense against known routing attacks, e.g., prefix hijacking
  - No new security holes introduced
- Flexibility in policy
  - Dynamic creation and removal of networks
  - Flexible boundaries for routing domains
  - Alternate paths and policy choice in transit networks

Proposed mechanisms

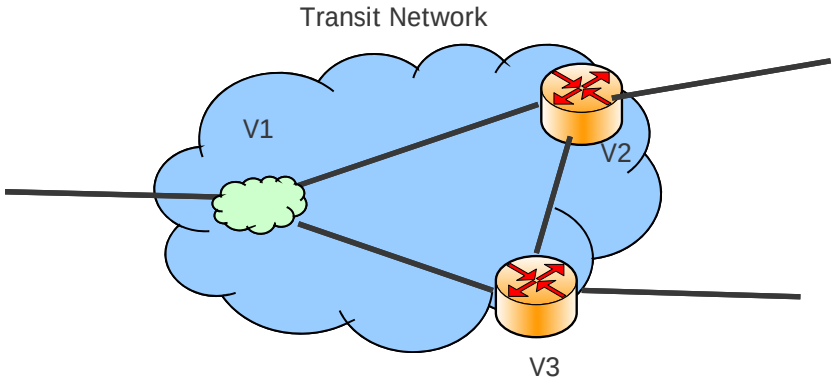
- Inherent support for mobility
  - Separate locators from identifiers
- Reduce routing table size
  - Use loose source routing by embedding routing paths into locators
- Support multiple interfaces and multicast
  - An end host can bind to multiple locators, each associated with a particular interface.
  - A locator can be mapped to multiple interfaces or multiple networks

Mechanisms

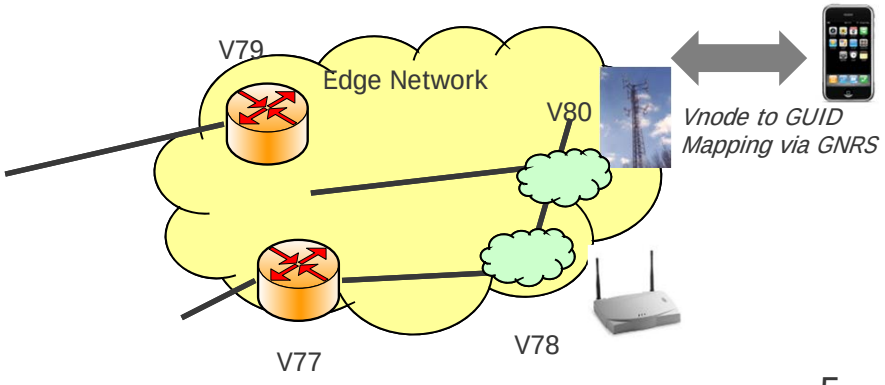
- Flexible routing policy
  - Support **both sender-driven and receiver-driven routing**
  - Place some routing information into a distributed database called Global Name Resolution Service (GNRS)
  - Different organizations can publish their own routing information
    - Through customized GUID to locator mappings
- Security: data plane security
  - Eliminate source spoofing
  - Source accountability: use public keys as host identity (GUID) and network identity
- Security: control plane security
  - Provide S-BGP equivalent guarantees at lower cost and improved deployability

Protocol design: Vnode/Vnet interdomain routing

- Flexible interdomain routing scheme based on generalization of “Vnode” from Pathlet routing
  - Basic idea is to summarize relevant properties of each network’s graph and expose it using a path vector protocol
- A typical network will expose the topology between ingress and egress/access nodes
- In addition to summarized topology, the routing information includes “Vnode” and “path” properties

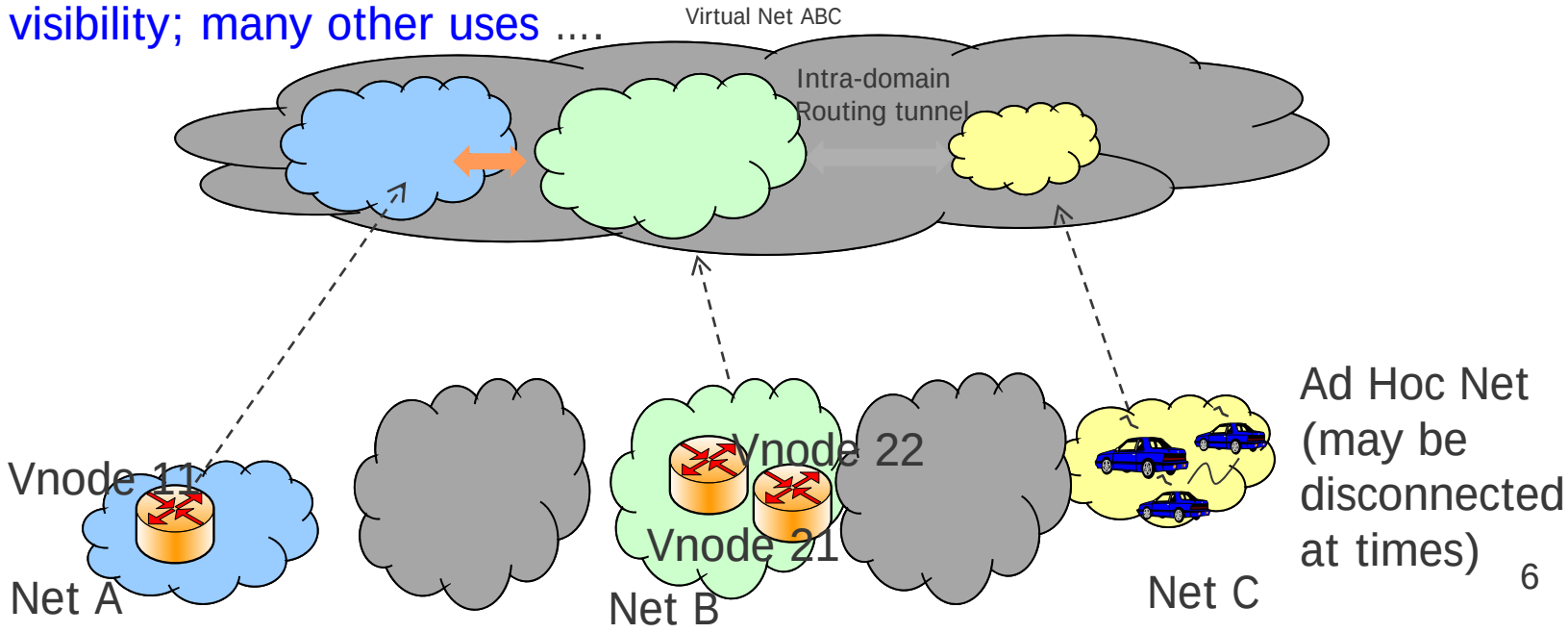


{Network #, Vnode list, vnode properties, Vnode links, link properties}



Protocol design: virtual routing domains

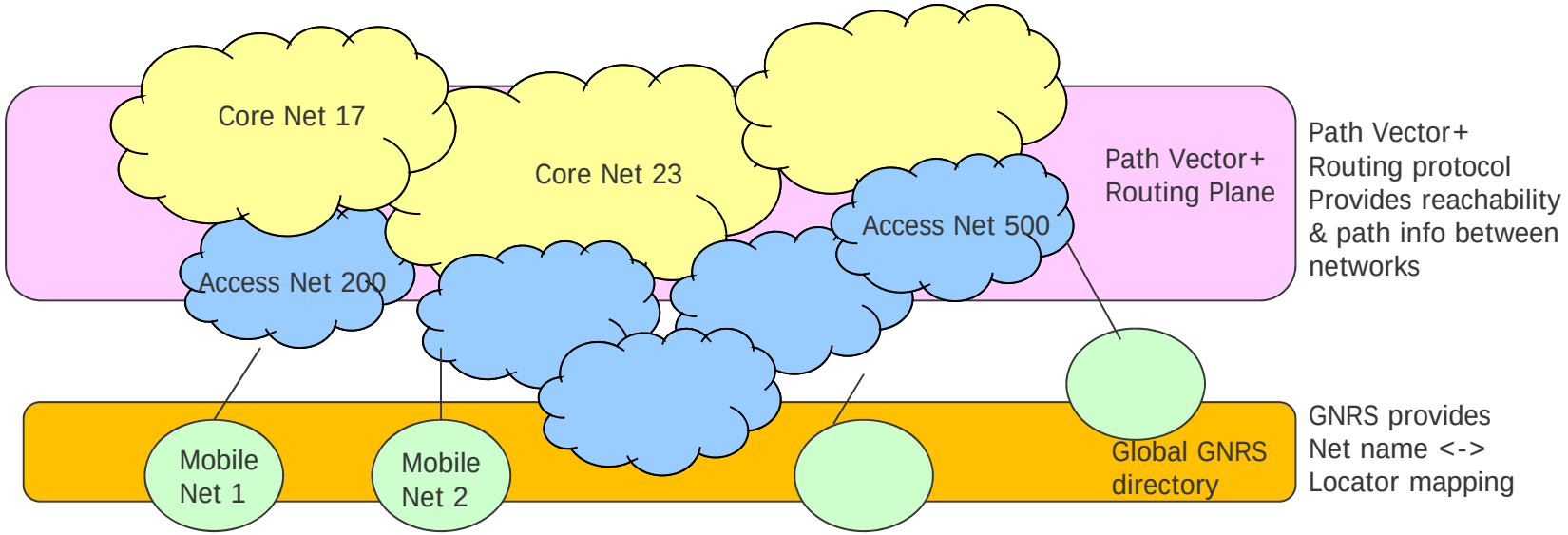
- Virtual network domains can be created by combining Vnodes and/or networks into logical aggregates
  - Vnodes and networks can form VN's without being physically contiguous – GNRS provides membership list & trust relationships
  - Virtual networks share fine-grain intra-domain routing information & expose multiple ingress and egress points to the inter-domain protocol
  - Can be used to aggregate disjoint wireless access networks (e.g. "NJfreeWiFi") or set up a "mobile cloud service" with improved routing visibility; many other uses ....



Design overview

- MobilityFirst interdomain approach uses

  GNRS service + enhanced path vector routing to achieve design goals – still evaluating multiple design options....

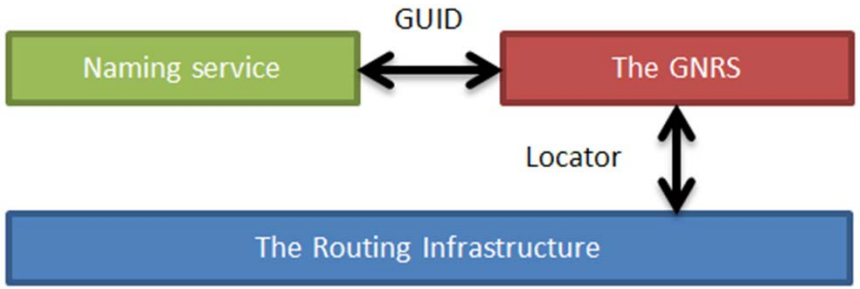


A typical usage scenario

- Alice sends data to Bob
  - Alice queries the naming service to obtain Bob's identifier, which is its public key
  - Alice queries the GNRS to get Bob's locator, which contains the routing information
  - Using Bob's locator, Alice knows how to reach Bob

Key components: naming services, GNRS, and the routing infrastructure

- The naming service maps a human-readable name to a GUID
  - Each network or end-host has a GUID as its identity.
  - GUID is the **public key**.
- The GNRS stores the location of a GUID
  - The GNRS service is essentially a distributed database.
  - Every network can have its local GNRS node.
- The routing mechanism determines how packets are routed from the source locator to the destination locator



The design of locators

- Used by both data and control plane routing
- Contains k levels of hierarchical network addresses (k is not fixed) and one host address.
- NID1:NID2:...:NIDk:HID
  - NID is the GUID of the network
  - HID is the identifier of the network interface of the host
- Locators may change. A device can have multiple locators at the same time.
- The minimum locator length determined based on empirical data
- Loose hierarchical routing
  - Sender locator: path reaching the core (tier-1 ISPs)
  - Receiver locator: path reaching from the core to the receiver network

Routing: Sender-driven vs. Receiver-driven

- A routing path usually consists of three phases: **up** (to the core) → **core** → **down** (to the edge)
- Traffic engineering: both **sender** or **receiver** can influence the routing path

TE / Used in which phase	Up	Core	Down
Sender-based TE (Bloom filter)	Yes	No	Yes
Receiver-based TE (Locator-based)	No	No	Yes

- Reconcile sender- and receiver-based approaches
  - By default, sender-based TE overrides receiver-based TE.
  - The destination network can set a “path-follow” bit in the locator to indicate that receiver-based TE should be enforced (i.e., the actual routing path should strictly follow the path indicated in the locator).

Basic receiver-driven routing

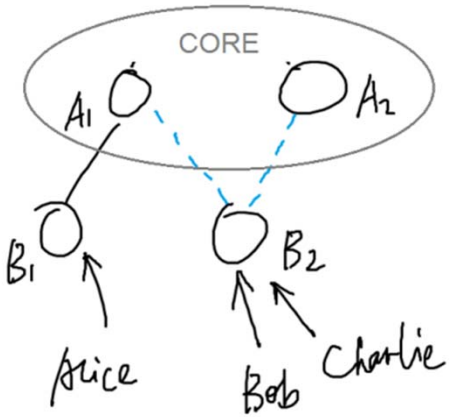
- Assume the topology to be a **DAG**
- Receiver-driven routing guarantees reachability
  - Additional optimizations using the sender-driven routing
- The locators of Alice and Bob contain

  loose source routes (LSR).
- Intra-network routing transparently handled by the network.
- Routing at the core and routing at the edge are handled differently and independently.

Query the Locator: Caching

- GNRS-level Caching
  - A network publishes entries about its provider. So the querier can just lookup GNRS instead of asking the actual network.
  - For example, B2 can publish

Source	Destination	Provider
:B1:	*	A1
A2:*	*	A2



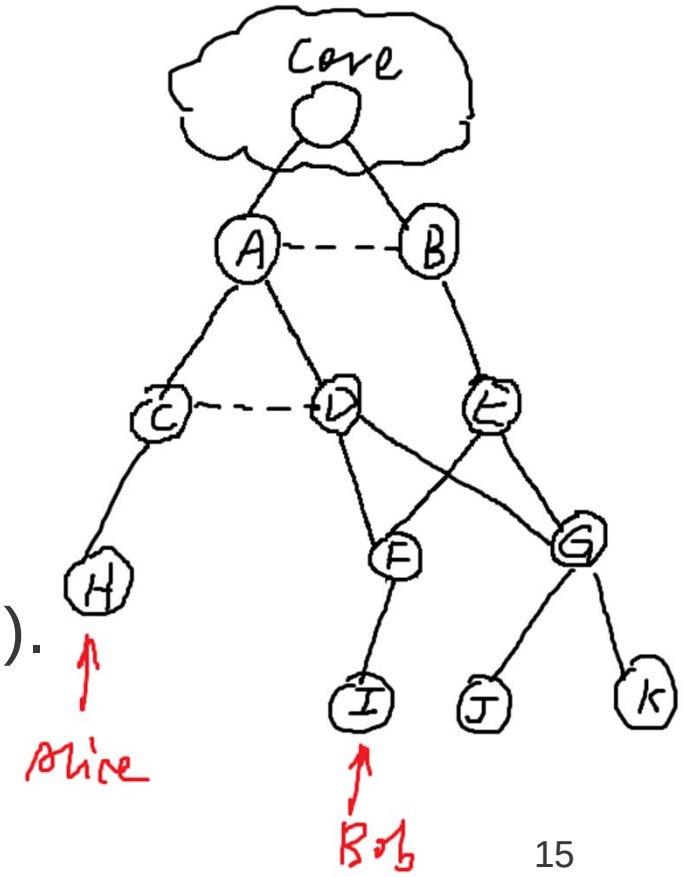
  - If no published GNRS entry matches, then the querier needs to ask the B2.
- Network-level Caching
  - A network can cache its (indirect) provider information so it does not need to ask its direct provider.

Sender-driven routing: motivation

- Motivation 1: **Peering links** are not in the DAG structure. Receiver-based routing cannot handle them.
  - When going up: how to decide when to use peering links?
- Motivation 2: The actual routing path can be different from the path embedded in the locator.
  - When going down: when multiple customers are available, how do we choose a customer?
- BGP addresses both issues by globally propagating reachability information
  - Not scalable. How to do it in a scalable manner?

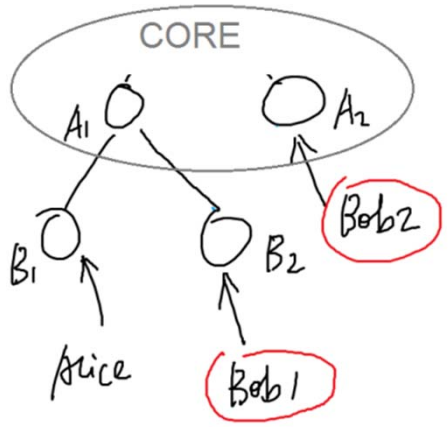
Sender-driven Routing

- A network maintains the (direct or indirect) **customer set of each of its neighbors**.
- Example: find a path from Alice to Bob.
  Consider decisions made by C.
  C knows the customer set of D:
  CS(D) = {F, G, I, J, K}.
  C knows the customer set of A:
  CS(A) = {D, F, G, I, J, K}.
  C wants to reach I.
  C first tries to use the peering link
  C→D, which can be used since $I \in CS(D)$.



Multiple Locators

- An end host can bind to multiple locators, each associating with a particular interface.
- Both Bob's locators are returned to Alice.
  The network helps Alice decide which locator to use by providing hints.



	Bob1	Bob2
Connection type	UMTS HSPA	802.11b
Last-hop RTT	200 ms	2 ms
Last-hop Bandwidth	1000 kbps	11 Mbps
Up time	2 hours	5 mins
Signal strength	Good	Poor
...	...	...

- Such information can be maintained by either network or GNRS.

Multicast

- A host address can represent multiple end-hosts. Such multicast addresses are transparent to external networks.
- For example, in the locator A : B1 : X, the multicast address X represents all taxies in NYC.
  - Then a packet from A : B2 : Y goes through the path Y→B2→B1→X.
  - Only B1→X is multicast and all other hops are unicast
- Multicast across network domains supported by virtual network node mappings
  - Identify the common network closest to the edge to fork the traffic

Security: Source Accountability

- Source accountability: detecting and preventing source spoofing
  - GUID is the hash of the **public key** of each end host.
  - 1st hop router verifies that sender does not spoof its GUID by sending a verification packet to the sender
    - $V = HMAC_{RS}(hash(P), src\ locator, dst\ locator)$
    - P: the packet, RS: the secret kept by the 1st hop router. It changes over time to prevent replay attack.
  - The sender then sends back the signed verification packet.
  - Such a verification procedure is performed only at the beginning of each flow, then the 1st hop router can use a “accept cache” to whitelist the flow.

Securing the Control-plane Message Exchange

- Securing the control-plane message exchange
  - S-BGP is difficult to deploy today partly due to the need for a global PKI.
  - In our design, routing updates can be signed and verified using the self-certifying addresses.
  - Secure the routing system against hijacking and route forgery.

A Simulation Testbed (In progress)

- **Input**: the Internet topology data (using the UCLA data)
- **Phase 0: Building the topology**
  Separate the topology to the core (tier-1 ASes) and the edge, visualize the topology
- **Phase I: Basic end-to-end simulation**
  Randomly pick two end hosts (sender and receiver) at two stub ASes, and consider the routing between the sender and the receiver. Key evaluation metrics include
  - Routing table size
  - Path inflation
  - Locator size
  - How is our design resilient to the topology change?

A Simulation Testbed (In progress)

- **Phase II: Mobility simulation**
  - Consider the mobility pattern of users (either using an analytical model or real traces). Quantify the “hand-over” overhead of users.
  - How fast is the up-to-date network location propagated?
  - What is the probability of experiencing a connection failure due to cached locator (depending on the moving speed of the peer)?
- **Phase III: Other features**
  - Consider two important features:
  - Multicasting addresses (intra-network multicast and inter-network multicast)
  - Multiple interfaces

Key Components: data plane vs. control plane

- Data plane
  - User application traffic
  - Naming service
- Control plane
  - Querying GNRS nodes
  - Information exchange among GNRS nodes
- Control plane service should not use the GNRS service again, otherwise we have the loop dependency.
- Both data and control planes share the same routing infrastructure.

Basic Receiver-driven Routing (cont.)

- Example 1:
  - The locator of Alice is A : B1 : C1 : X
  - The locator of Bob is A : B2 : C2: D : Y
  - Find out if they are in the same network by performing longest prefix matching. In this example they are in the same network A. Therefore the routing path is: X→C1→B1→A→B2→C2→D→Y
- Example 2:
  - The locator of Alice is A1: B1: C1: X
  - The locator of Bob is A2: B2: C2: Y
  - Alice wants to talk to B. In this case they are not in the same network. The routing path is: X→C1→B1→A1→CORE→A2→B2→C2→Y

Query the Locator: The Iterative Approach

- Example: Alice queries the GNRS about Bob's locator A:B:C:X
  1. GNRS looks up the database and finds out the mapping

  Bob's GUID → C:X

  2. GNRS asks C to expand the locator.
  3. C knows its provider is B. C asks B about its provider.
  4. B knows its provider is A. B asks A about its provider.
  5. A returns A to B (A is in the core).
  6. B returns A:B to C
  7. C returns A:B:C:X to GNRS
  8. GNRS returns A:B:C:X to Alice
- Optimizing performance: GNRS-level caching and Network-level caching

GNRS Services

- Our routing design does not specify the information exchange protocol among GNRS nodes.
- GNRS nodes themselves just use locators to communicate with each other (to avoid the dependency).
- Different organizations publish their own GNRS service.

Sender-driven Routing: how to efficiently store the customer set?

- Use **bloom filter,** essentially *k* different hash functions mapping from network ID to {0, 1}.
- A network maintains *m* bloom filters where m is the number of neighbors.
  - Pros: constant space overhead, no false negatives
  - Cons: false positives
- Dealing with FPs:
  - When a network discovers a non-reachable destination, it reports such a false positive back to the sender.
  - A network maintains a “false positive cache” to avoid future FP.