Addressing: Is HID scoping necessary?

- No in theory
    - HIDs are globally unique, so each HID refers to only one host and there is no confusion
- Yes in practice
    - Too many hosts globally, so it is impossible to keep routing information for all of them at each router
    - Current Internet doesn't need explicit scoping since CIDR provides implicit scoping by prefix
    - To address some HID in a remote AD, it must be scoped by the AD in the routing DAG
        - HIDs without AD scoping are assumed to be in the local AD
- Multiple levels of scoping
    - With one level of scoping, the number of hosts inside a domain may be too large to keep individual entries in the route table
    - Larger ADs may have several sub-ADs (similar to AIP)
    - Hiding sub-ADs
        - Possible security etc. issues with exposing sub-AD configuration?
        - May be useful, e.g. to support migration
        - Can use NATs that expand HID nodes to sub-AD → HID DAG

Control plane: Which elements of the architecture manage route calculation?

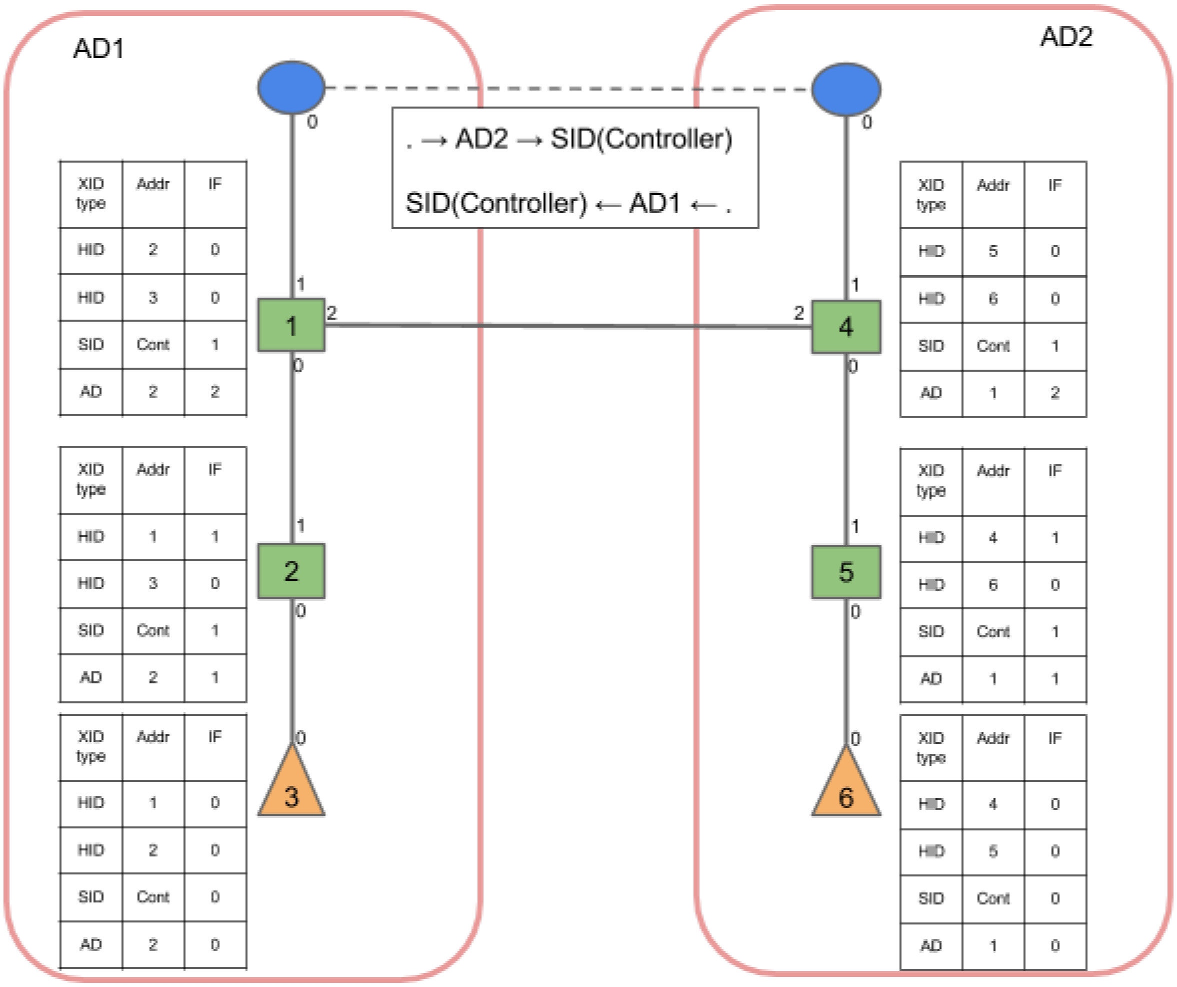
- In current Internet, this is distributed for both intra and inter domain
    - Individual routers exchange info then calculate routes using same algorithm
- In XIA, we can centralize this in the controllers within an AD
    - Controller gathers info, calculates routes, then distributes routes within AD
    - ADs and controllers have many-to-many relationship
        - Large ADs may have distributed controllers
        - One controller can manage several ADs

Data plane: What information should routing tables contain?

- Current Internet
    - (dst IP, netmask) → (gateway, interface)
    - Other info for path selection, e.g. cost
- XIA
    - (dst XID type, dst XID) → (gateway, interface)
    - How to do flow-based routing?
        - Use fixed-length flow ID, which is the hash of source and destination DAGs concatenated and other relevant information (from David Naylor's whiteboard on Basecamp)
        - i.e. (flow type ID, hash) → (gateway, interface)
    - Note on XION
        - After processing SCION header, routing is done by (AD, egress AD ID)
        - Maybe after initial processing, can cache entry
            - (XION, dst XION ID) → (gateway, interface)

        - Maybe even allow references in case routes change
            - (XION, dst XION ID) → (AD, egress AD ID)

<u>Example topology</u>
This demonstrates the routing tables after intra domain routing is running.



. → AD2 → SID(Controller)

SID(Controller) ← AD1 ← .

XID type	Addr	IF
HID	2	0
HID	3	0
SID	Cont	1
AD	2	2

XID type	Addr	IF
HID	1	1
HID	3	0
SID	Cont	1
AD	2	1

XID type	Addr	IF
HID	1	0
HID	2	0
SID	Cont	0
AD	2	0

XID type	Addr	IF
HID	5	0
HID	6	0
SID	Cont	1
AD	1	2

XID type	Addr	IF
HID	4	1
HID	6	0
SID	Cont	1
AD	1	1

XID type	Addr	IF
HID	4	0
HID	5	0
SID	Cont	0
AD	1	0

Legend:

Shapes	Blue circle = controller Green box = router Orange triangle = host
Lines	Solid black = physical links Dotted black = logical links Solid pink = AD grouping
Numbers	Interfaces

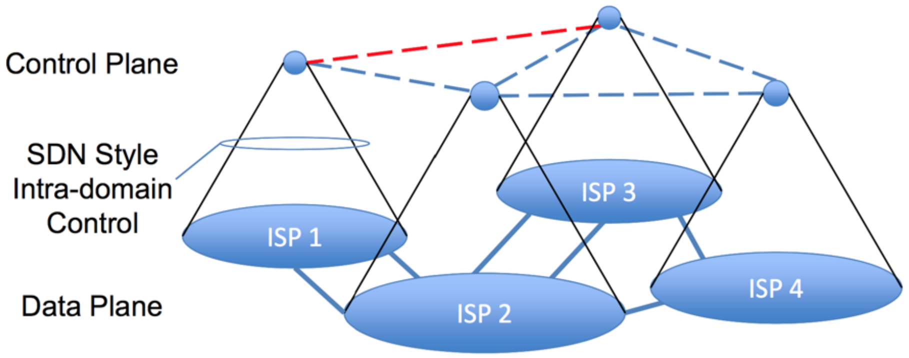
Interdomain routing: How to identify boundary routers?

- In the current Internet, boundary routers are statically configured

    - BGP set up as TCP sessions across point-to-point links
    - Susceptible to misconfiguration.
    - Not secure: prefix hijacking, man-in-the-middle attacks
        - http://www.nanog.org/meetings/nanog49/presentations/Tuesday/HowSecure_NANOG_print.pdf
- For XIA, boundary routers can be dynamically discovered
    - Self-identify through HELLO messages received from neighboring ADs
    - Intrinsic security prevents spoofing

Interdomain routing: Where does interdomain protocol run?

- Current Internet
    - eBGP between boundary routers
    - iBGP required to distribute routes internally
        - usually fully meshed; special rules/configuration to avoid looping
- In XIA, routing handled by controller
    - “xBGP” between controllers instead of routers
        - Done over logical links
            - → AD(neighbor) → SID(controller) or SID(BGP)
        - Essentially, back to the one “router” per AD scenario (see diagram below)
            - Similar to Routing Control Platform.
            - This removes issues of configuring a full mesh of iBGP sessions, and using route reflectors.
        - Once we have support for intradomain AD and SID routing, a naive version of xBGP that uses OSPF can be implemented first.
    - No need for iBGP
        - Controller distributes routes within AD
    - Only exchange AD reachability info (not HIDs)



Interdomain routing: How to express preference across multiple exit points for outgoing traffic?

- In current Internet, ASes use local preference
    - Higher local preference value = higher preference
    - Exchanged between routers within the same AS

- For XIA, local preference can be configured/decided at controllers

Interdomain routing: How to express preference across multiple exit points for incoming traffic?

- In current Internet, ASes use MED to indicate preference to neighboring AS
    - Lower MED value = higher preference
    - MED is usually ignored if no financial settlement, like peering
- For XIA, if we only route by highest-level AD, then there will not be fine-grained control of exits used
    - Consider an example: AS C is customer of AS P, and both have their own networks spanning the US
    - Host 1 in AS C, located in Boston, wants to send a packet to host 2 in AS C, located in San Fran
    - Ideally, C wants the packet to be transmitted across the U.S. in P's network since it is the provider
        - Current Internet supports this, as C can assign the lowest MED value to the ingress router in San Fran for host 2's prefix.
    - For XIA, we use the DAG → AD(P) → AD(C) → HID(2)
        - However, the partial DAG processed at P will be → AD(C) → HID(2)
        - This means P will likely put the packet back into C's network before it gets near HID(2), depending on metric used
            - If fewest hops used, then P will return C's packet immediately
- Possible solutions
    - Route by lower-level ADs (sub-ADs),
      e.g. → AD(P) → AD(C-California) → HID(2)
        - ASes in current Internet do use different AS numbers to express different routing policies
        - For large ADs that want certain hosts to be scoped using sub-ADs, they can advertize sub-AD reachability information and publish DAG that scopes these hosts by sub-AD to the name resolution service (assuming there's one)
          Related questions:
            - Does the AS just send out reachability info for relevant ADs that it wants to scope HIDs with?
                - This seems to work
            - Will there be a case where a router only knows how to route to a higher level AD?
                - If so, can use fallback to the higher level AD,
                  e.g. AD C publishes HID(2)'s DAG as:
                  → AD(C-California) → HID(2),
                  fallback → AD(C) → HID(2)
            - Is it a good idea to create ADs just to take advantage of multiple exit points?
                - Hosts can belong to multiple ADs, so should be fine
        - xBGP details

- Each controller knows if an AD it manages is used for scoping and will thus advertize it to neighbors
- Dynamic MED configuration
  - Increase by certain amount for each sub-AD traversed
  - Cache MED values and advertize highest one
  - e.g. MED=0 for AD(C-Boston),
    MED=100 for AD(C-NY),
    MED=200 for AD(C-Philly)
- ASes will need to be able to identify ADs that belong to it
  - Is AS info implicit in XIP header?
  - If not, each controller can keep a statically configured list of ADs belonging to same AS

- New principal type for exit points or geography,
  e.g. → AD(P) → EXIT(SanFran) → AD(C) → HID(2)
  - Used for route selection and doesn’t affect route correctness
    - Can be ignored if router does not have route entry for it
    - Name resolution can include this type without requiring additional fallbacks
      - e.g. → EXIT(SanFran) → AD(C) → HID(2)
  - Interdomain routing
    - Seems like it will simplify interdomain routing protocols
    - Can just announce AD reachability as before, and internal routers can just use shortest egress route to send packets to the next hop AD
  - Intradomain routing
    - Need additional routing info for exit points, which can be calculated at controller
    - Just need to add geographic information to LSAs
- Middleboxes that temporarily modify the DAG to control exit points
  - Seems like most complicated and inflexible solution
  - When boundary routers exchange HELLO messages across AD boundaries, keep track of HID of other party
  - Select which egress point to use by scoping with HID of boundary router in the other AD
    - e.g. → AD(P) → HID(Boundary router of P near SanFran) → AD(C) → HID(2)
    - Source DAG has to do something similar
  - MB near HID(1) needs to know about boundary routers near HID(2)