Problem

Our current routing infrastructure assumes that we have a single router per domain, which is a weakness of our prototype. There is some urgency to fix this because this our assumptions about routing are adding complexity to porting SCION to XIA, which is a high priority project.

Inter-Domain Routing

Currently uses a simplified implementation of the OSPF link-state routing protocol.

Relevant files:

- 1 router daemon per router instance
  - https://github.com/XIA-Project/xia-core/blob/master/daemons/xrouted/xrouted.hh
  - https://github.com/XIA-Project/xia-core/blob/master/daemons/xrouted/xrouted.cc
- Modify and print routing table info
  - https://github.com/XIA-Project/xia-core/blob/master/bin/xroute
- Router
  - https://github.com/XIA-Project/xia-core/blob/master/daemons/common/XIARouter.hh
  - https://github.com/XIA-Project/xia-core/blob/master/daemons/common/XIARouter.cc
- Routing table
  - https://github.com/XIA-Project/xia-core/blob/master/click-2.0.1/elements/xia/xiaxidroutetable.hh
  - https://github.com/XIA-Project/xia-core/blob/master/click-2.0.1/elements/xia/xiaxidroutetable.cc

BGP + XIA Control Plane: https://xia.basecamphq.com/projects/5562171/file/161615977/XIA-Ctl.ppt

Intra-Domain Routing

Design-level questions

- Use a decentralized link-state protocol or a centralized SDN design (e.g., as done by 4D)?
    - Either will work for integrating SCION
        - For edge routers, we're always going to have a centralized controller that picks between different inter-domain routing policies, so in some sense, both options are SDN-style
    - We should be able to steal at least some of the existing inter-domain link-state routing (OSPF) code (in xia/daemons/xrouted/)
    - **We are going to use a centralized SDN design**
        - This gives us the advantage of an optimized multi-step bootstrapping process so that only router--and not host--information needs to be flooded
        - We are going to have a centralized controller regardless to interface the inter- and intra-domain routing
            - Add more complexity to the already necessarily smart controller and make the routers dumb
            - This is still a single point of failure
        - Controller can act as a nameserver for SIDs of well known services
- What (e.g. HIDs, SIDs, 4IDs, ports) do routers need to broadcast to ensure reachability?
    - Current inter-domain routing protocol seems to use HIDs and port numbers
        - SIDs currently routed by using HID in DAG, but only because SID forwarding tables are currently just not populated
    - Note that a host can only be connected to a single router
        - Each router uses XHCP to tell its hosts what AD they are in and the location of the nameserver
- Should edge routers be identified as such beforehand and connected to the centralized controller, or dynamically announce themselves to the controller?
    - **We are going to statically configure edge routers as such**
        - This will be much easier to work with
        - It’s not clear that dynamically discovered edge routers even have any advantages
- How will SCION integrate with our intra-domain routing solution?

Where does control lie for constructing SCION paths?

Implementation-level questions

- How to differentiate a domain's edge routers so that we don't flood outside of the domain and the centralized controller knows who to talk to?
    - Current inter-domain routing always floods all ports with no notion of internal and external facing ports
    - Possible zone option: one HID per router but associate an AD with each router port
        - Each router can only belong to one AD in current XIA implementation
    - In the real world, how do edge routers configure themselves? Who owns an edge router? Links between edge routers use private addresses? What is the reason?
    - Does SCION need to do this, and if so, how is it implemented?
- What metric(s) to use for path cost?
    - Initial implementation probably based purely on hop count
- How to structure forwarding tables for XIDs?
    - Refer to current inter-domain routing table structure
        - Seems to be based only on HIDs and port numbers
        - Eventually will want different routing tables for different principal types
- We need a testbed

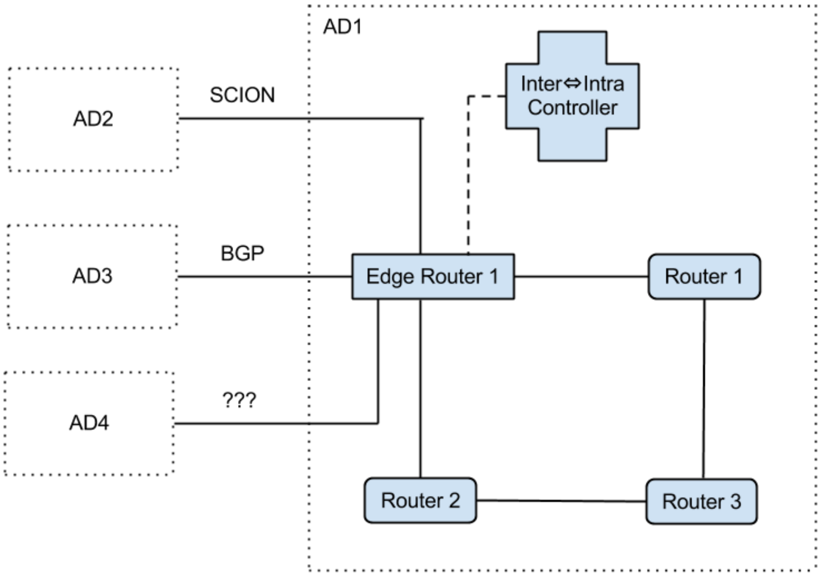

Figure 1: Example routing architecture diagram

Protocol Description

1. Edge routers for the AD are configured beforehand as such
2. Each router blindly sends HELLO packets containing its HID one hop to directly connected neighbors
    a. Edge routers do not send HELLO packets outside the domain
3. Each router waits a certain amount of time to receive HELLOs from neighbors, then replies to each of the received HELLOs with another HELLO, which now includes the neighbor's HID, as a sort of ACK
4. After meeting the neighbors, each router now floods the AD with an LSA that contains its **router** neighbor information
    a. Edge routers do not send LSA packets outside the domain
5. The central controller collects all LSAs for a certain amount of time, and then combines them into a graph of the AD's router topology
6. Using hop count as the distance metric, the controller calculates the shortest paths from each router to every other router
7. The controller then constructs the routing table for each router, and sends each router its routing table
8. Each router adopts the routing table and communication between routers and the central controller is established
9. Each router then sends link-state information about its host HIDs to the central controller
10. Central controller collects this information just as before, and calculates routing table updates so that all hosts can now communicate as well
11. Routing table updates are sent to each router
12. Each router applies the routing table updates and communication between routers, hosts, and the central controller is established
13. On a regular interval, each router regularly repeats the HELLO exchanges to check link health and sends out LSAs on any changes
    a. Upon receiving a subsequent LSA, the controller will wait a certain amount of time to collect other LSAs originating from the same link-state change event, and then recalculate routing tables

These routing table updates are then disseminated to the appropriate routers

SCION Integration

Previous work: https://github.com/XIA-Project/xia-core/tree/scion_integration

- Is there a better way to do addresses than the multipart XID addresses? Variable size XIDs?
    - This is due to the entire path being encoded into the packet?
- SCION implemented as a new XIA principal type

SCION header sizes are much bigger than XID header sizes.
SCION requires three major elements inside each AD, including beacon server, certificate server, and path server.

SCION + XION: https://xia.basecamphq.com/projects/5562171/file/161615978/XIA-SCION-Implementation.pptx

Papers

- SCION: available on the XIA web site (Oakland conference)
- "Design and IMplementation of a Routing Control Platform", ACM NSDI 05.
- "STRIDE: Sanctuary Trail -- Refuge from Internet DDoS Entrapment"
  In *Proceedings of ACM Symposium on Information, Computer and Communications Security (ASIACCS),* Hangzhou, China, May 2013.
- Gude, Natasha, et al. "NOX: towards an operating system for networks."
  ACM SIGCOMM Computer Communication Review 38.3 (2008): 105-110.
    - http://www.cs.stonybrook.edu/~vyas/teaching/CSE_592/Fall12/papers/nox.pdf
    - http://www.noxrepo.org
    - https://github.com/noxrepo/nox-classic/wiki
- Koponen, Teemu, et al. "Onix: A Distributed Control Platform for Large-scale Production Networks." OSDI. Vol. 10. 2010.
    - http://www.usenix.org/event/osdi10/tech/full_papers/Koponen.pdf
- Tesseract 4D
    - http://www.cs.cmu.edu/~4D/papers/tesseract-nsdi07.pdf
- 4D
    - http://www.cs.cmu.edu/~4D/papers/CMU-CS-05-117.pdf

People

- Soo Bum Lee<soobum@cmu.edu> and Yue-Hsun Lin <tenma@andrew.cmu.edu>
- Hyeontaek Lim <hl@cs.cmu.edu> - Early involvement with XIA prototype implementation