# Hierarchical Routing for the Global Data Plane

CS262A Fall 2023 Group 8: Mira Sharma, Nia Nasiri, Eric Nguyen, Tianyun Yuan

## Problem

The scalability of GDP hinges on the scalability of its routing system. Previous work [1] implements secure delegated routing, with one global RIB. Routing with a hierarchy of RIBs is better suited for massive scale and nested trust domains. Our goal is to implement hierarchical delegated routing while minimizing advertising traffic, optimizing for very large scale of Data Capsule servers, and increasing data capsule availability as servers constantly go offline and online again. Challenges arise due to the data-centric, location-agnostic guarantees of the GDP, as well as its flat namespace and security focus.

## Background

The **Global Data Plane (GDP)** enables secure distributed computing that harnesses untrusted and diverse resources and service providers. [2]

**Data Capsules (DCs)** — the building blocks of GDP — carry standardized, secure, and certified bundles of data.

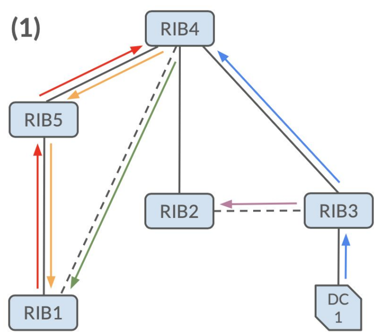
**Routing Information Base (RIB)** serves as a critical data structure within a router, storing and managing routing information that includes the optimal paths to reach various destinations in a network.

A **GDPName** is a universally unique hash that identifies DataCapsules and other components of the GDP.

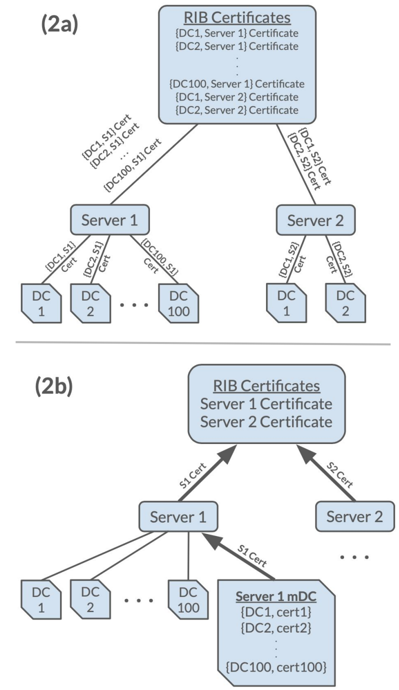
GDP routing uses **secure delegation** — in order to advertise ownership of a GDPName, a server has to have a certificate signed by the public key corresponding to the GDPName, proving that it is authorized to do so.

## Solution

- We choose the "best" route: one with the **least number of hops between trust domains**.
- For simplification, we have a single central RIB corresponding to a single trust domain.
- We focus on **source-based routing** between trust domains: within trust domains, we can use OSPF. A successful run of our routing algorithm is detailed in Figure (1)
- On each RIB, we store entries representing replicas of DCs and their corresponding servers in distinct caches, providing efficiency by decoupling server and DC storage.
- We also introduce a per-server **meta-DataCapsule**, or mDC, that contains the names and certificates of each server DC. Figure (2a) and Figure (2b) shows how the mDC can drastically cut advertising traffic, improving latency when servers come back up after disruption.
- A **checksum** on the mDC verifies that certificates are up-to-date in **heartbeat messages**.

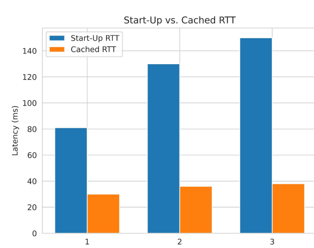



- A replica of DC1 is stored under RIB3. A request is sent to update RIB3, which also forwards up to the global RIB 4. An ACK is sent backwards along the same route.
- RIB3 gossips with RIB2, so if RIB2 GETs DC1 later, it can directly go to RIB3 instead of going up to RIB4.
- RIB1 GETs DC1, defaulting to querying its parent, RIB5, which queries the global RIB4.
- RIB4 sends a request (containing best route from RIB1 to DC1) to RIB1 backwards along the GET's route.
- RIB4 sends a request to update caches backwards along GET's route. RIB1's cache is updated with RIB4 as the next hop for DC1. If RIB5 had a direct connection to RIB3, its cache would be updated with its next hop as RIB3.

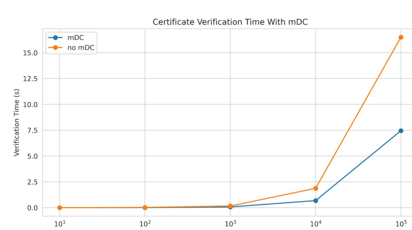



## Results

RTT latency increases as levels are added to the hierarchy, but caching mechanisms mitigate this overhead and improve locality.




DC AdCert verification with no mDC occurs at the RIB; in contrast, the mDC separates server and DC verification, reducing server start-up latency by passing a single certificate to the RIB.




## Future Work

- While the simulated benchmarks serve as a proof of concept, the algorithms are to be extended to realistic scenarios, such as FOGROS2.
- We want clients to have the ability to choose what Trust Domains its packets route through (TrustNet)
- We could implement the RIB key-value store as a distributed hash table (DHT) for faster lookups

## References

[1] *Secure, Fast, Location-Independent Routing for the Global Data Plane*, Arya et al, https://people.eecs.berkeley.edu/~kubitron/courses/cs262a-F21/projects/reports/project17_report_ver4.pdf

[2] *Global Data Plane: A Federated Vision for Secure Data in Edge Computing*, Mor et al, https://people.eecs.berkeley.edu/~kubitron/courses/cs262a-F23/projects/Suggestion_Docs/icdcs_19.pdf