# A Secure Multicast Tree for the Global Data Plane

Arun Sundaresan, Mikkel Svartveit, Tony Hong

## Background

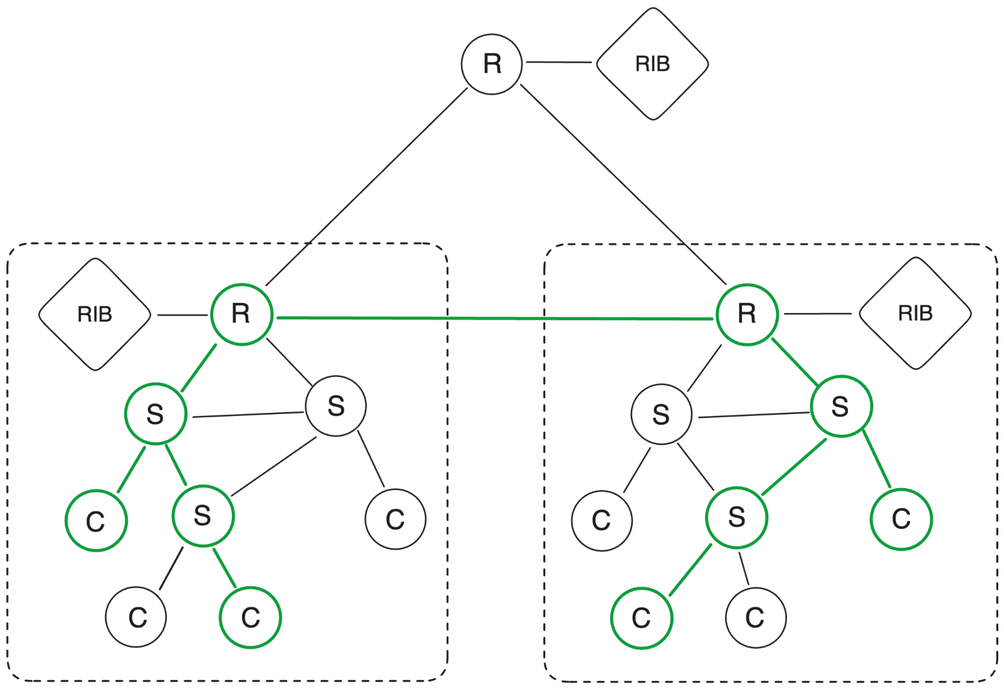
The Global Data Plane (GDP) provides a unified and secure method to manipulate resources on the edge. GDP partitions resources by ownership into a series of trust domains, which can be organized hierarchically. Additionally, clients in the GDP may be members of several multicast groups, which can be organized without regard to the trust domain hierarchy. In 2021, Plutowski et al. proposed a multicast tree building protocol for PSL where trust domains took the form of a k-ary tree. Additionally, they did not provide any security guarantees over their messages or address trust domain hierarchies.

## Goals

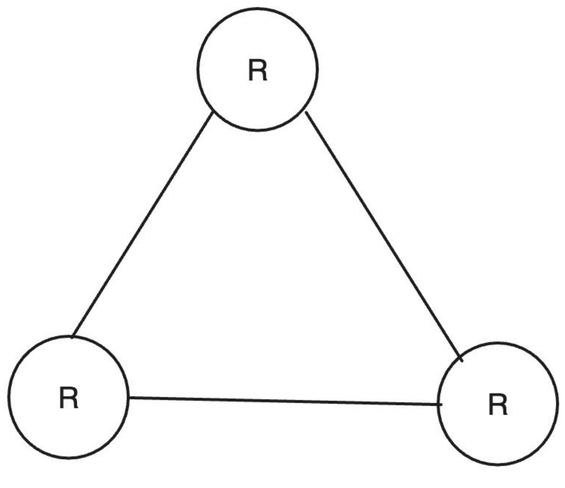
1) Online multicast tree building protocol supporting clients at any level of the trust domain hierarchy
2) End-to-End Symmetric Encryption over messages
3) Reduce time needed for sending messages
4) Avoid overloading routers high up in the hierarchy to improve scalability

## Proposed Multicast Protocol

When new routers or links between trust domains are added to the network, a message is propagated up the tree to notifying ancestor trust domains to update their stored topologies. Similarly, when multicast groups are created, a message propagates up the tree signaling that a descendant has information about the multicast group. In order for a client to join a multicast group, a message is routed up the tree until a RIB aware of the multicast group is encountered. This "lowest common ancestor" RIB runs the Dijkstra algorithm on its stored topology to find a path from the joining client to the rest of the multicast tree. Our approach also leaves trust domain routers to find paths within their trust domains. To send messages, we simply flood the appropriate multicast tree.

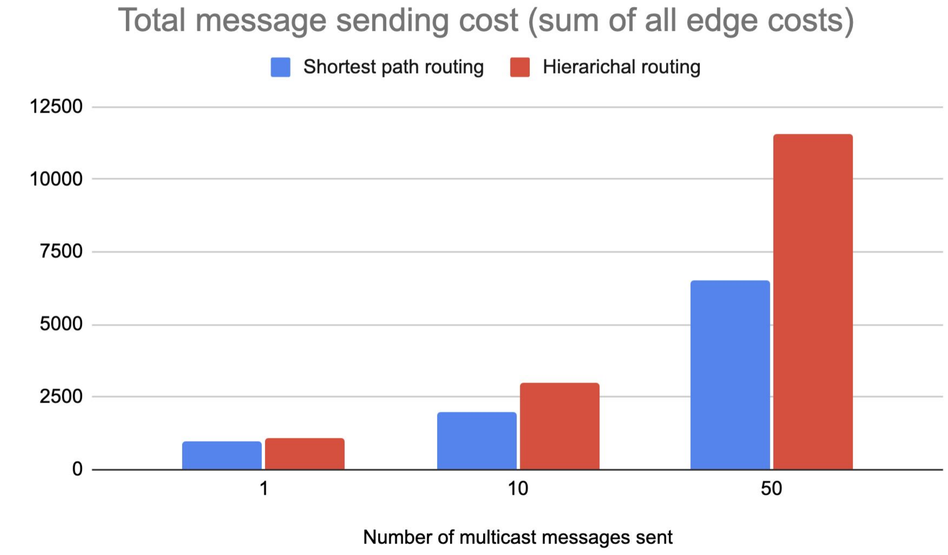



The figure above shows an example of a network topology. A multicast tree with four participating clients is displayed in green. Notice that the trust domain routers do not need to go through their common parent router in order to multicast to each other. The figure below on the left shows the topology the top-level router runs a path finding algorithm on, and the right figure shows how other routers can be used to find a shorter path. Red edges show the route under a strictly hierarchical protocol.

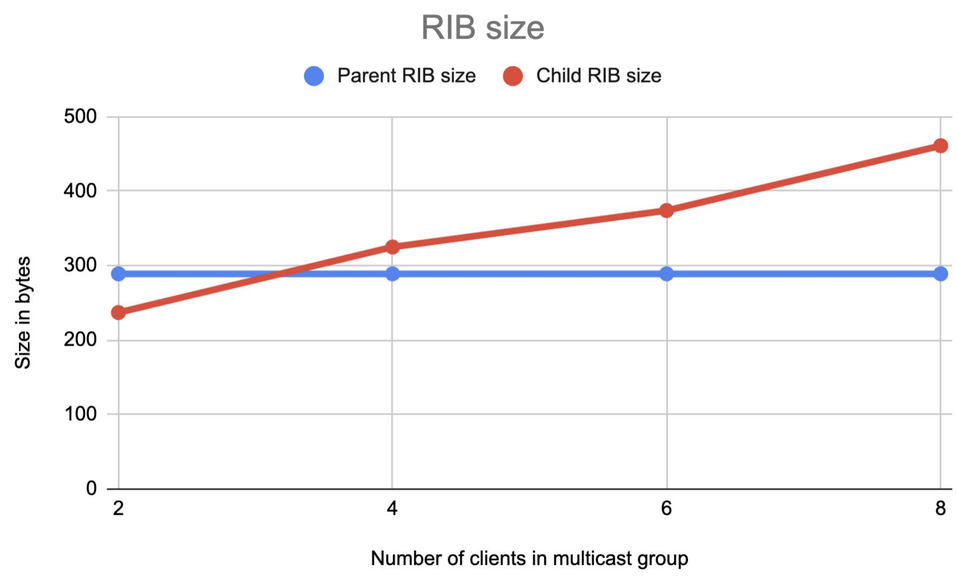



## Evaluation

We have created a Python simulator that allows for experimenting with different GDP network topologies. For these benchmarks, we used the topology and multicast group illustrated in the previous section. All links within the trust domains have cost 1, while all links outside have cost 100. Shortest path routing is measured using the multicast tree built by our protocol, while "hierarchical routing" uses an approach where a multicast message has to go through a common ancestor (like the earlier PSL multicast paper).




The following figure shows that when adding clients within a trust domain to an existing multicast group, only the size of the local RIB is affected by number of clients. The ancestor RIBs stay constant in size.




## Security

We assume clients' identities are verified by a globally visible certificate authority. When multicast groups are created, the first client to join is designated the "owner" and can send messages to the certificate authority authorizing other clients for the multicast group. Trust domain routers ask the certificate authority for such a message when clients ask to join a multicast group. Additionally, the first client to join a multicast group generates symmetric keys used to encrypt and HMAC messages for that group. Hence, confidentiality and integrity are ensured in sent messages. We also support key rotation and revocation.