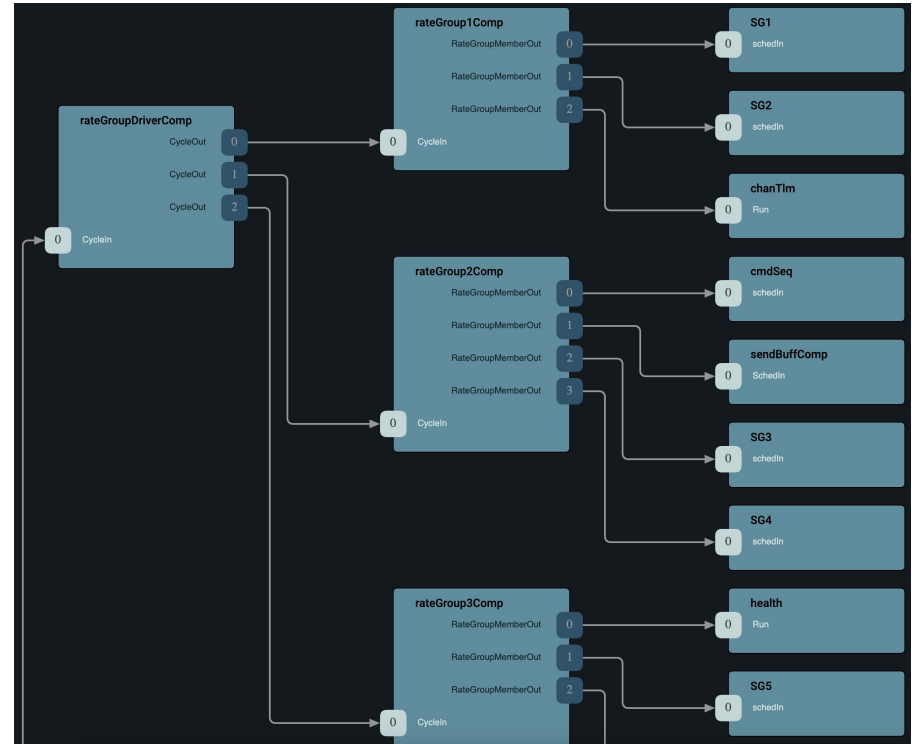


F Prime Topology Visualization Updates

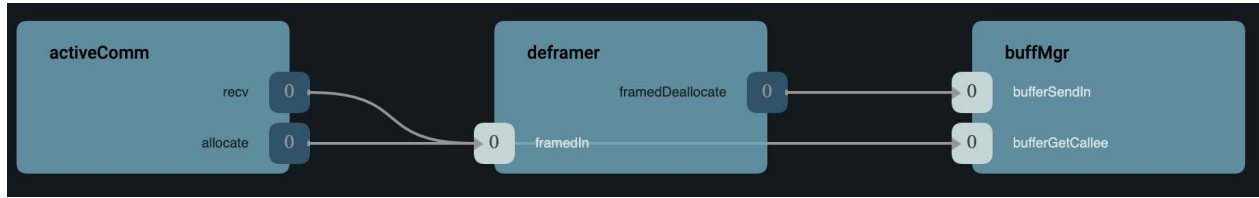
Dan Delany (397M), 10/12/23

Why is this work important?

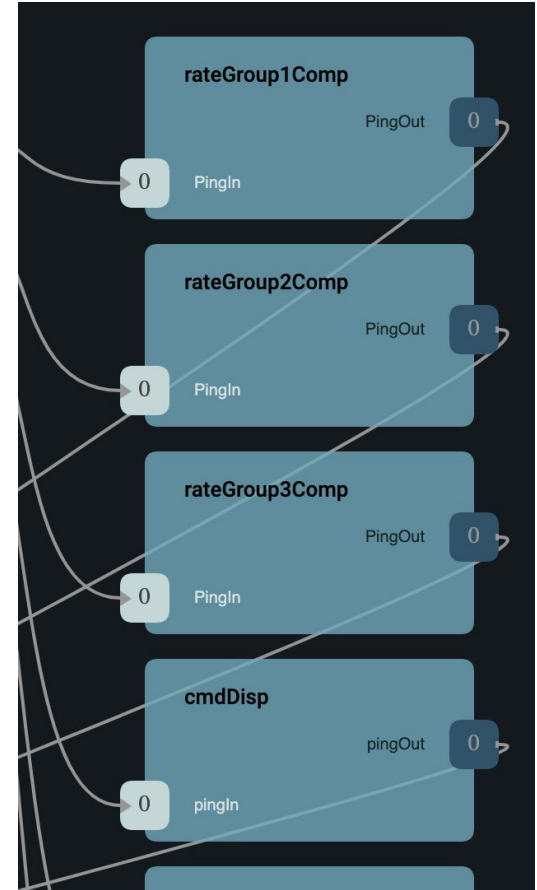
- Visualization improves understanding of complex F Prime topologies
- Naive graph visualization methods have problems which make them confusing or ambiguous



Problems with Naive Methods



- Ambiguity when edges go through ports or overlap other edges
- Edges on top of/colliding with nodes
- Edge crossings/intersections, especially acute angles



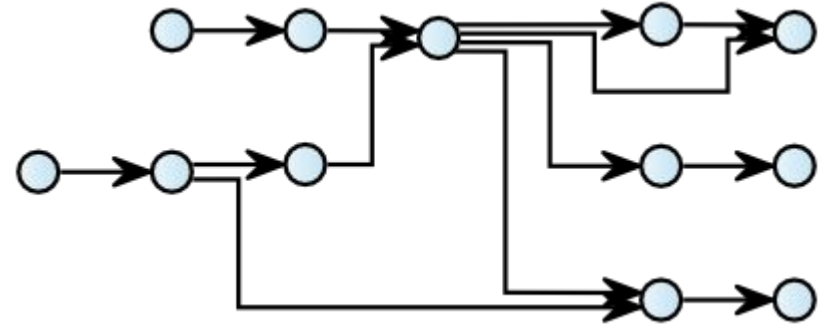
Why is this hard?

- Math/heuristics to detect bad cases eg. collisions
- Line-routing strategy so lines avoid each other
- Multivariate optimization problem
 - Multiple valid ways to layout a graph
 - Degrees of freedom: node placement, edge routes, port order
 - Need a fitness function - what are we optimizing for exactly?

- Lots of existing literature on different ways to solve this problem

Open Source Software - ELKJS library

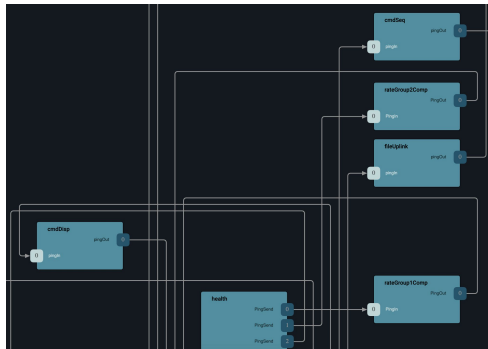
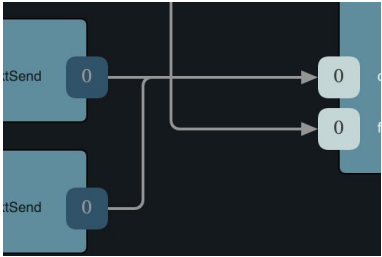
- **ELK** is [Eclipse Layout Kernel](#), open source Java visualization framework from Eclipse Foundation
- **ELKJS** provides Javascript wrapper library for ELK layout algorithms, so they can be used in a web browser
- Both released under permissive open source EPL license



- [ELK Layered algorithm](#) performs layout (but not rendering)
- Based on:
Sugiyama, Tagawa, Toda, “Methods for Visual Understanding of Hierarchical System Structures”, 1981

Observations & Future work

- Most topologies are much more readable with ELK layout
- Rounded corners help disambiguate intersections/crossings (eg. Downlink)
- BUT some “degenerate” cases exist which are worse than the naive solution (eg. Health)



FUTURE WORK

- Expose ELK configuration settings to users via settings panel
- Try other layout algorithms provided by ELK
- Interactive layout (users can move boxes after initial layout, edges get re-routed automatically)