VizPub: Visualizing the Performance of Overlay-Based Pub/Sub Systems

Nils Peder Korsveien, Vinay Setty, Roman Vitenberg
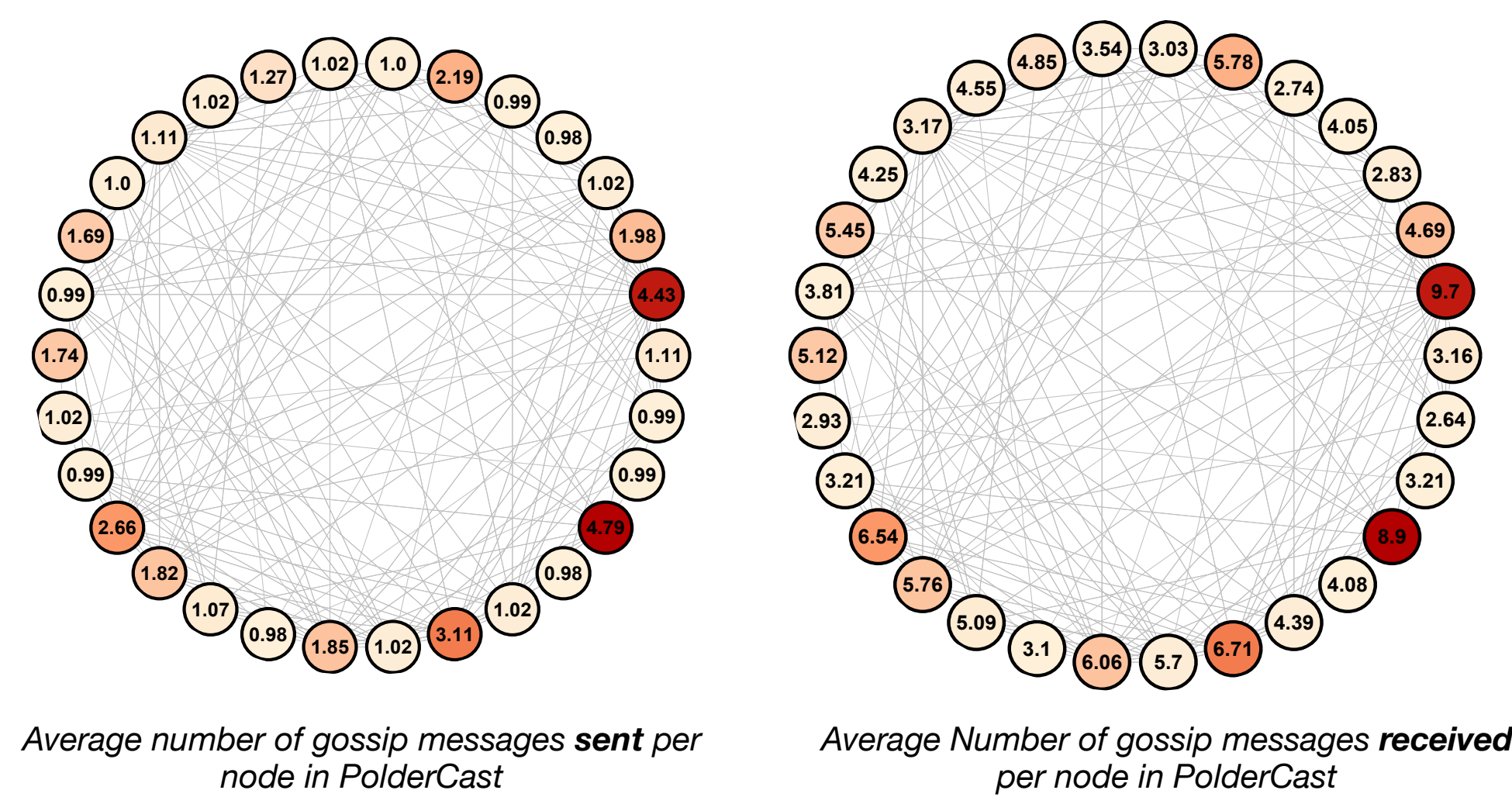
UiO : **Department of Informatics**
University of Oslo

We propose a tool for visualizing a variety of performance metrics in topic-based publish/subscribe systems, ranging from dissemination of publications to overlay properties. The tool can be used for gaining insight into the system performance and for comparing different pub/sub systems.
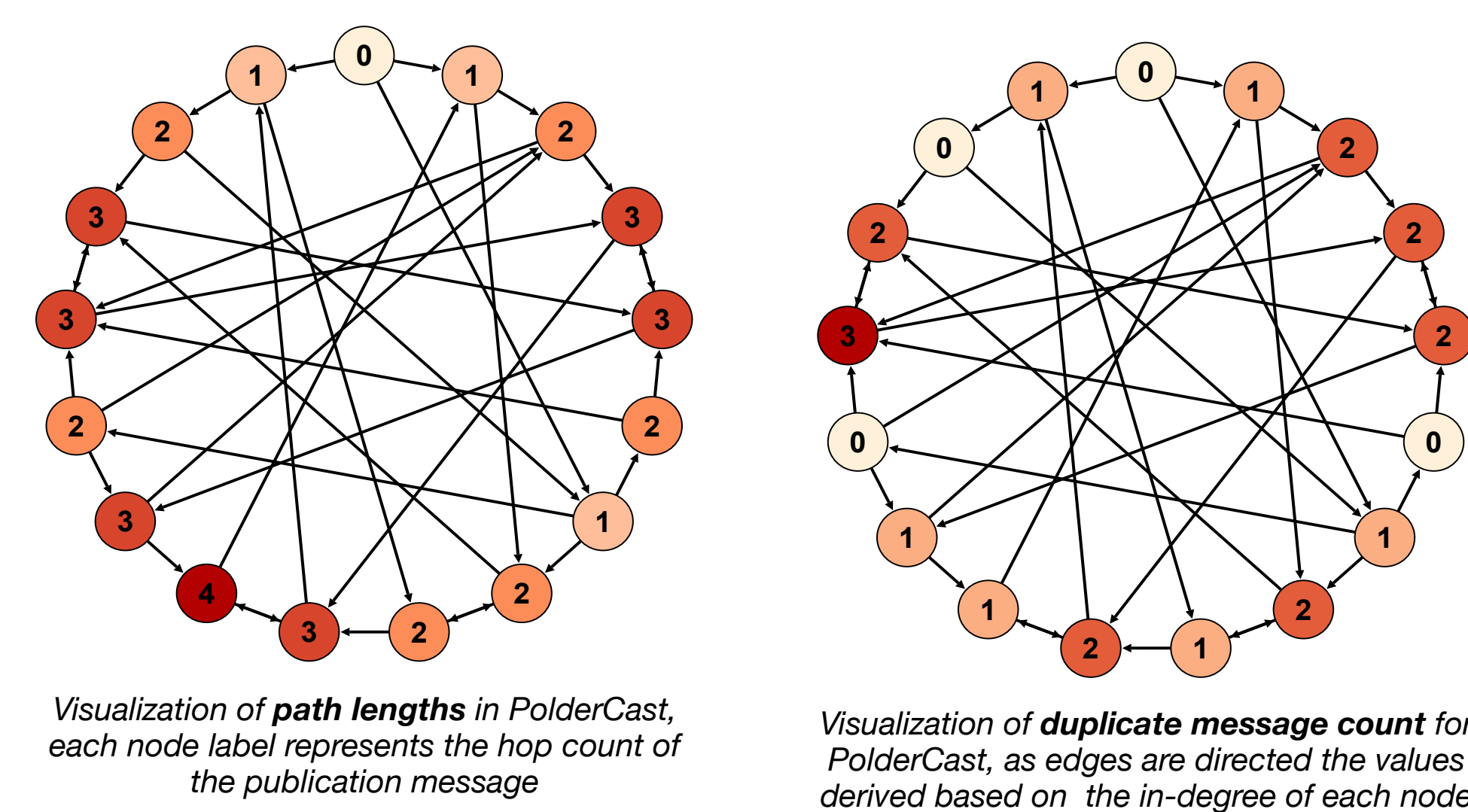
Demonstration

Write demonstration description here.

Architecture

Reporter Interface

Method Name	Description
long reportId()	The Unique Id of this node
long[ ] reportNeighborIds()	The unique ids of this node's neigbors
long[ ] reportTopics()	List of topic ids this node subscribes to
long reportControlMsgsSent()	Number of overlay control messages sent
long reportControlMsgsReceived()	Number of overlay control messages received
long reportControlBytesSent()	Number of overlay control bytes sent
long reportControlBytesReceived()	Number of overlay control bytes received
PubMessage[] reportPubMsgsSent	List of publication messages sent
PubMessage[] reportPubMsgsReceived	List of publication messages received

PubMessage Structure

Data field	Description
long MsgId	The unique id of this message
long TopicId	The unique ids of this node's neigbors
long SourceId	List of topic ids this node subscribes to
long DestinationIds[ ]	Number of overlay control messages sent
long OriginalSenderId	Number of overlay control messages received
long TimeStamp	Number of overlay control bytes sent

- The *Reporter* capture performance metrics of the pub/sub system over a period of time by implementing the *reporter interface*

- The *Collector* is responsible for pulling the information from the Reporters and aggregating it into various visualized metrics

- The *Visualization Unit* is a machine running the *Gephi framework*. The final *.gexf* file generated by the Collector can be opened in Gephi and the user can choose to playback the pub/sub execution

- The structural overlay properties such as the degree, diameter and clustering coefficient can be derived by building a topology of the overlay (nodes and edges) using the information reported via reportId() and reportNeighborIds().

- The dissemination properties including hit ratio, path lengths, and number of duplicate messages received, can be derived by ana- lyzing the list of publication messages

Structural properties

Average number of gossip messages **sent** per node in PolderCast

Average Number of gossip messages **received** per node in PolderCast

- Visualize structural properties of the overlay
- More
- And more..

Event dissemination

Visualization of **path lengths** in PolderCast, each node label represents the hop count of the publication message

Visualization of **duplicate message count** for PolderCast, as edges are directed the values derived based on the in-degree of each node

- Visualize publication message dissemination step-by-step
- More
- And more..

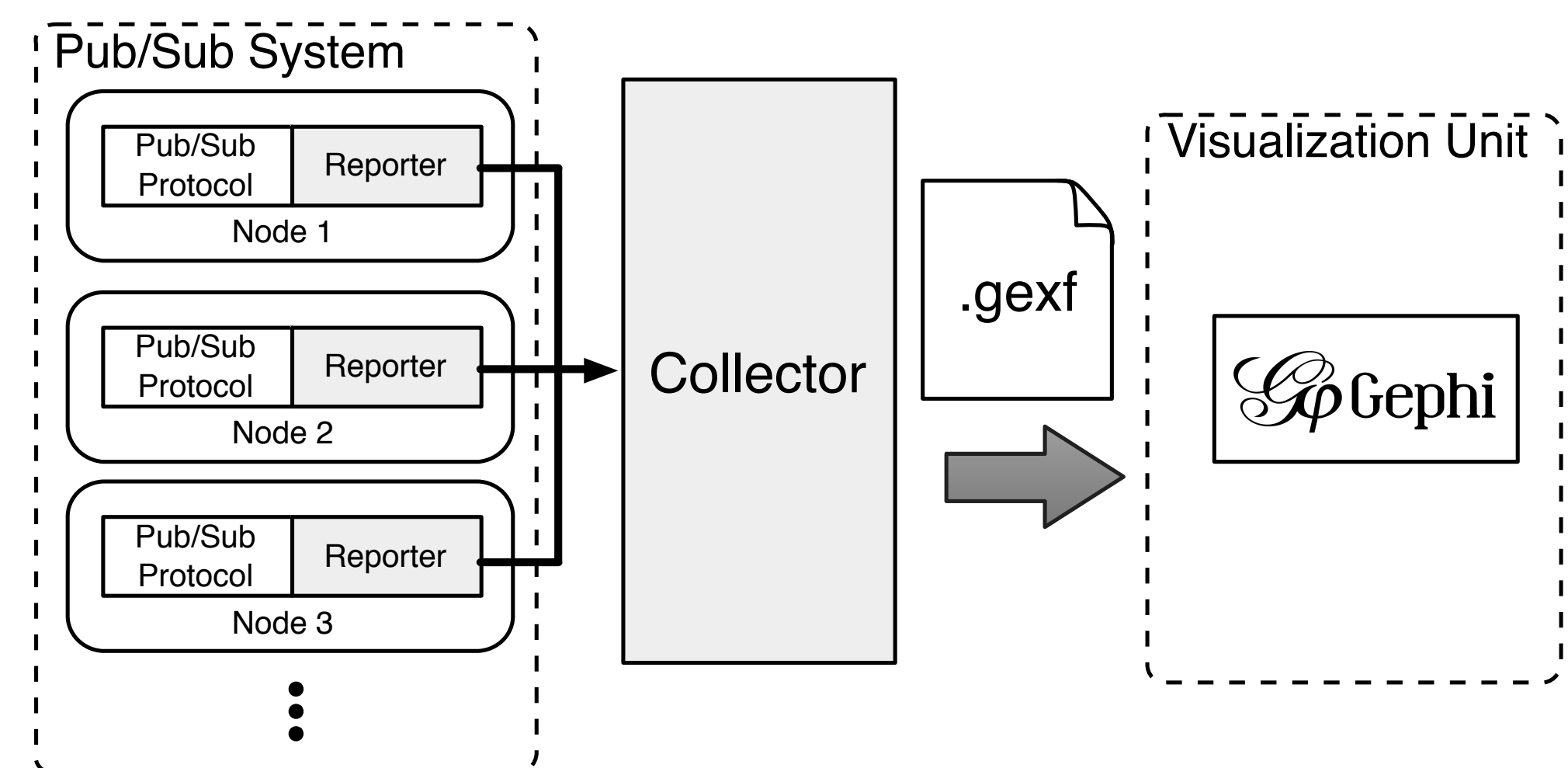## Gephi Framework

- The *Timeline Component* enables the user to play back the simulation, animating both size, color and labels of both nodes and edges

- The *Time Window* can be adjusted to strip the graph of any nodes, edges or attributes that does not exists within the specified interval

- The *Play/Pause Button* initiates playback, where step size and playback speed can be defined by the user

- Using the *Statistics Component*, the user may calculate selected metrics for the chosen time interval

- Information regarding node properties and attributes can be displayed in the *Node Query Panel* by clicking on a particular node with the *Node Query Tool*