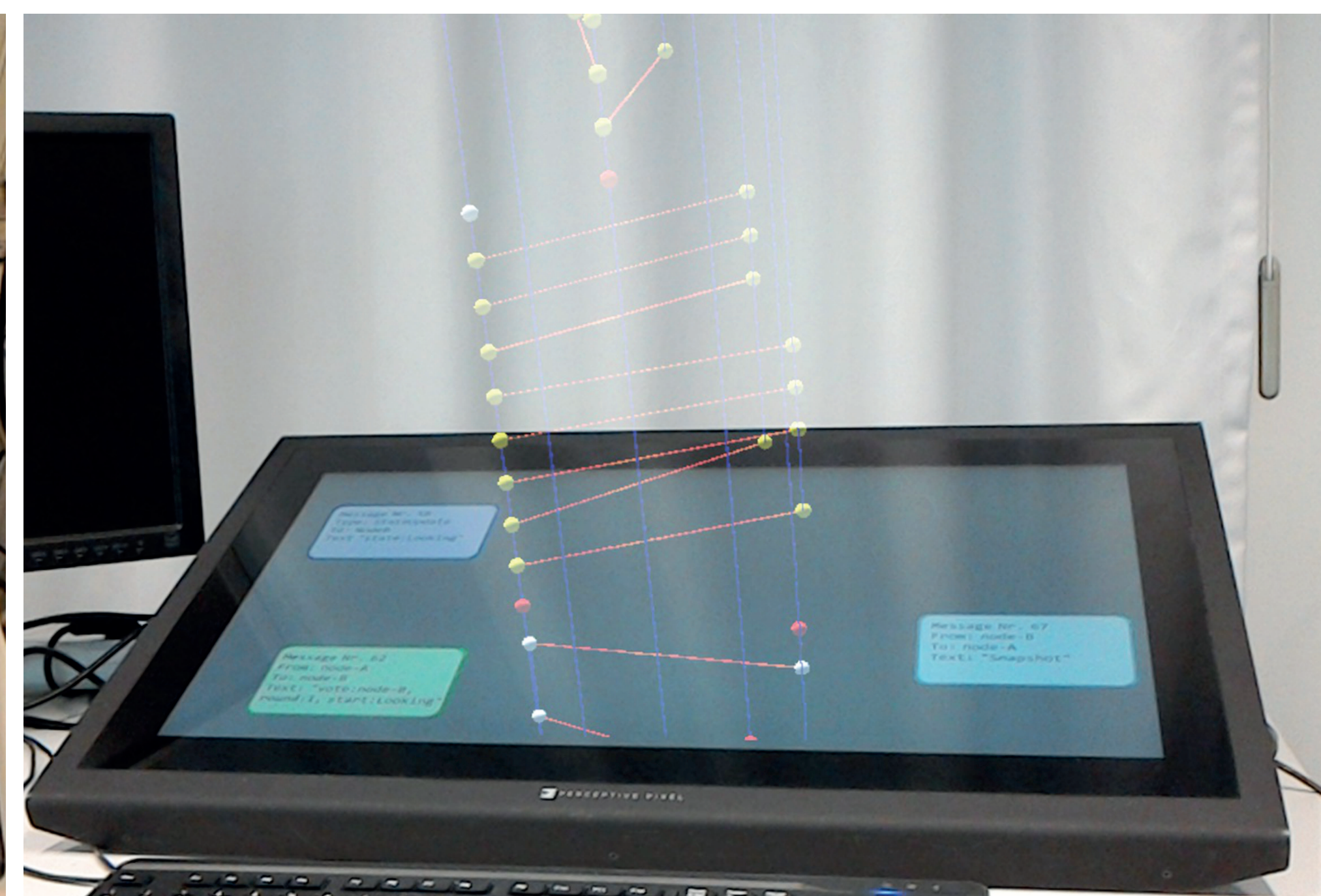
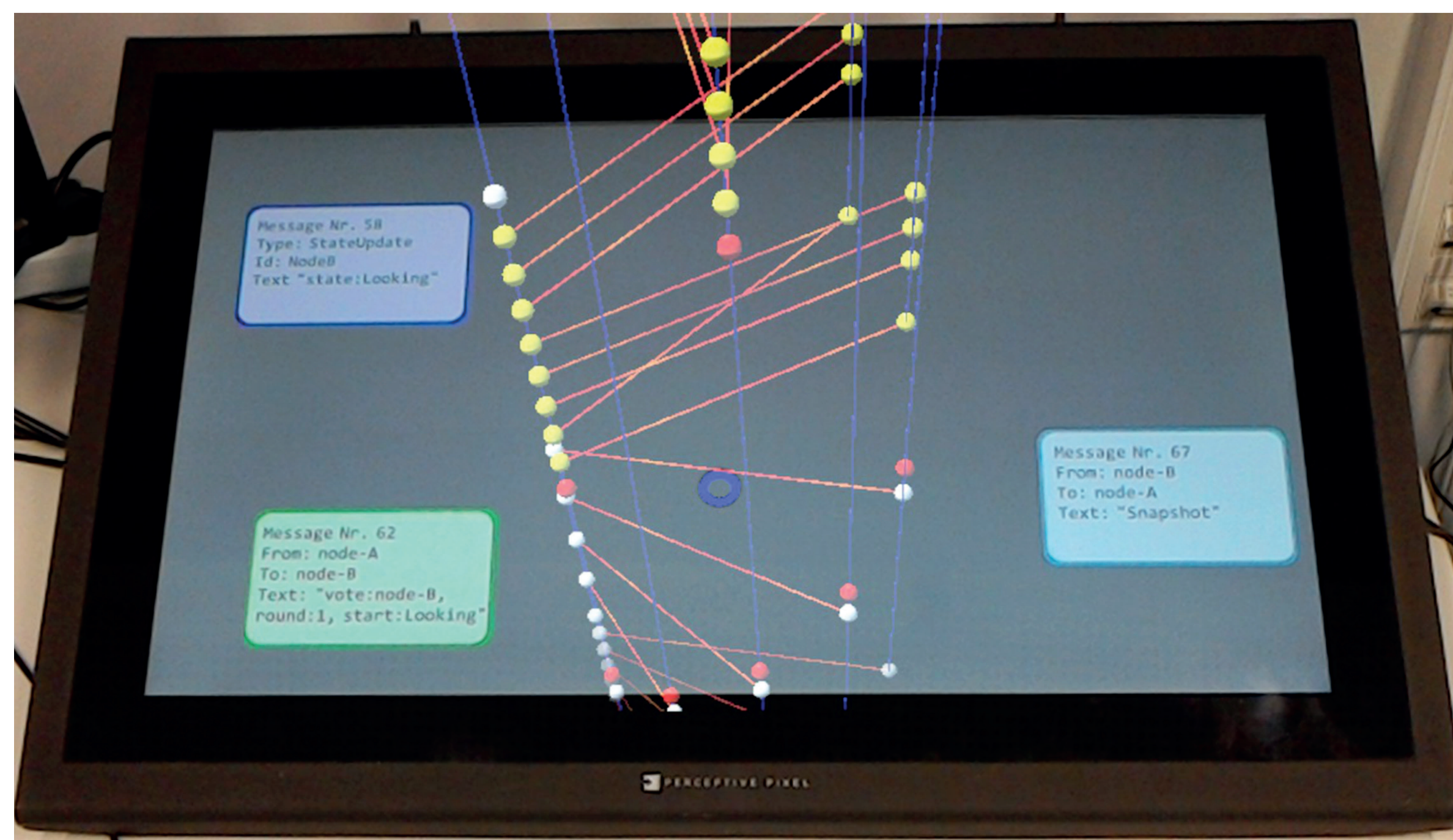
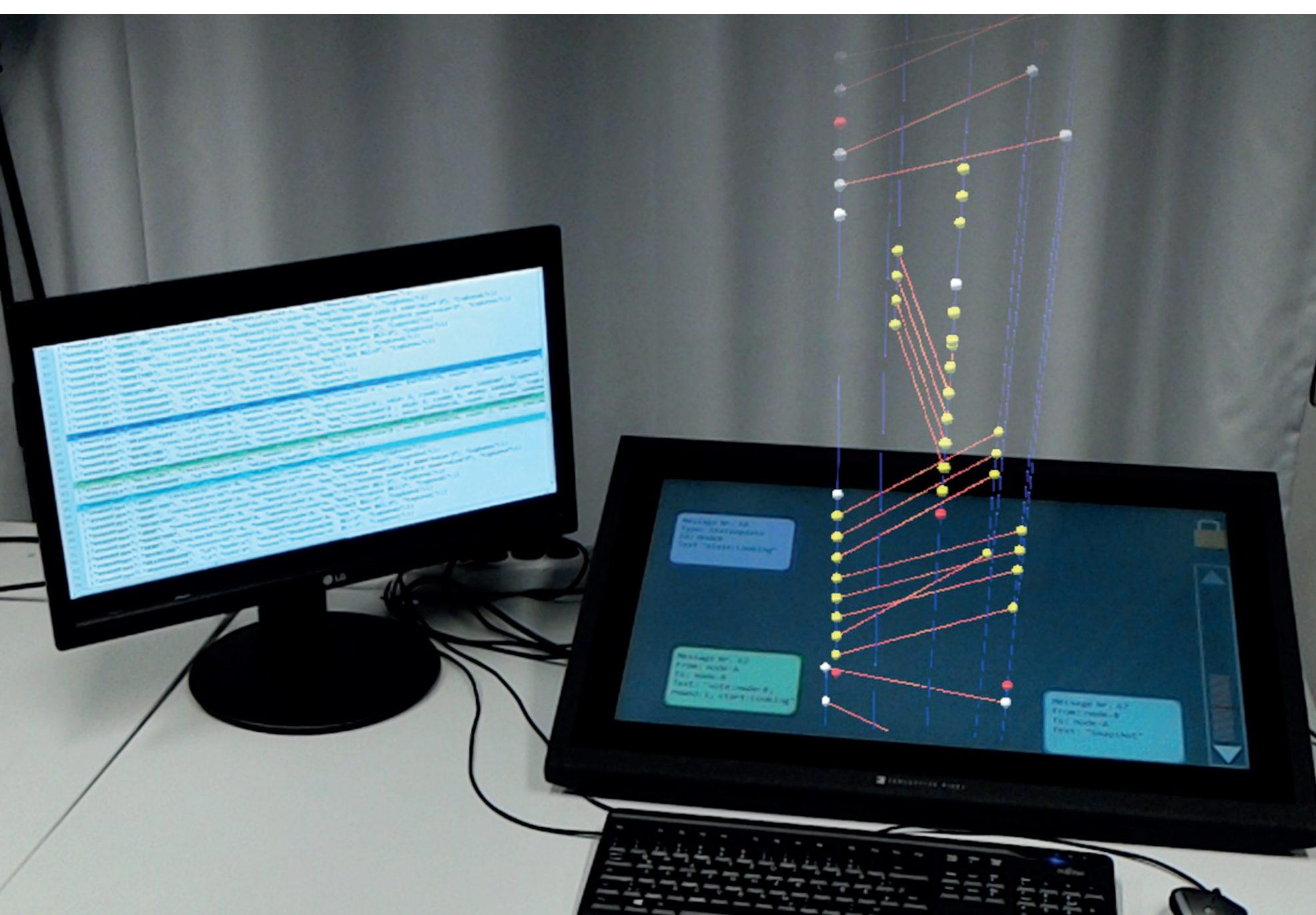


DebugAR: Mixed Dimensional Displays for Immersive Debugging of Distributed Systems

Patrick Reipschläger, Burcu Kulahcioglu Ozkan, Aman Shankar Mathur, Stefan Gumhold, Rupak Majumdar, Raimund Dachzelt

CHI 2018
Engage with CHI



Debugging Distributed Systems

The high number of messages with non deterministic delivery timings between nodes, message losses, and data corruption makes distributed system very hard to debug. Since they often cannot be efficiently analyzed using traditional GUI tools, we propose to use immersive technologies in a multi-display environment to tackle these shortcomings.

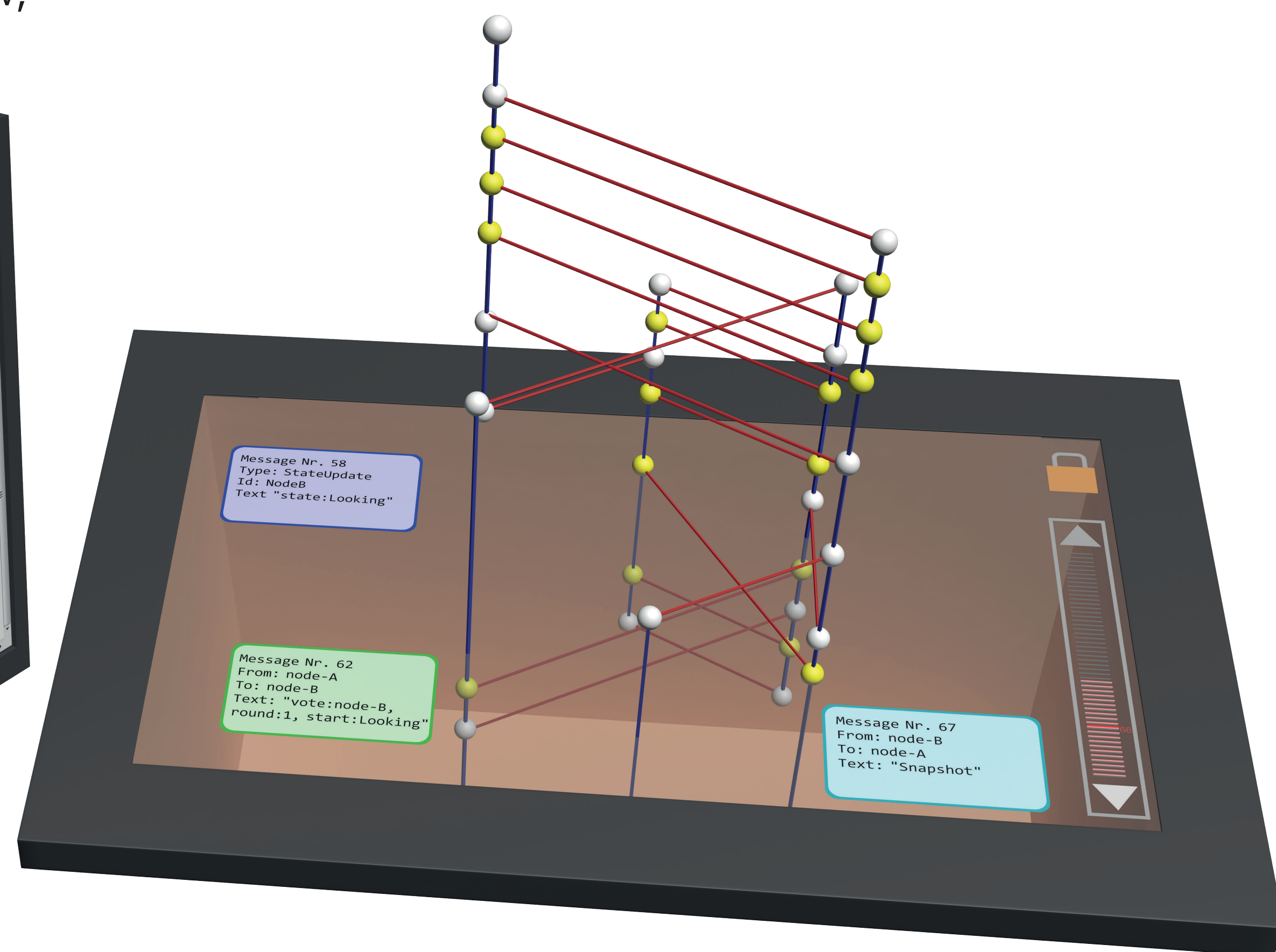
DebugAR Concept

- ▶ Combine message log on conventional display with tilted multi-touch surface
- ▶ Interactive surface shows distributed system execution traces in conjunction with AR head-mounted display
- ▶ Conjunction of familiar tools and new, immersive approach to debugging



Immersive 3D Visualization

- ▶ Shows development of distributed system over time
- ▶ Visualizes actors as vertical lifelines and their state as color coded spheres at discret time steps
- ▶ Current time step is located directly on screen of interactive surface, space above and below represents past and future
- ▶ Number of total time steps visible can be configured, messages above limit will fade out to not obstruct view
- ▶ Colored sphere either placed at each discret time step or as aggregation only on state changes
- ▶ All parts of 3D visualization are sole AR objects only visible through HMD
- ▶ Multi-touch Interaction with visualization: Dedicated slider widget for changing current time frame, pinch gesture for changing total number of visualized messages



Layouting

- ▶ Provides several layouting mechanics to organize actors based on users needs
- ▶ Default circle layout aims to maximize readability, because messages do not intersect with uninvolved actors
- ▶ Proximity of actors can be based on magnitude of message flow
- ▶ Real-life location of nodes can also be used if available, because it can offer useful hints when debugging
- ▶ User can rearrange nodes with drag gestures on the interactive surface

Message and Node Pinning

- ▶ Select specific messages in the 3D visualization with tap gestures to pin them on the display
- ▶ Detailed description of messages and the state of nodes
- ▶ Messages stay pinned even if the time frame of the visualization is moved
- ▶ Freely organizable collection of information relevant for debugging specific problems

Requirements for Visualizing Distributed Systems

- ▶ **Representation of the current system state:** A programmer needs to be immediately aware of which nodes are alive, have crashed, etc.
- ▶ **Configurable layout:** The location of the nodes are of significant concern for the programmer's mental map of the system.
- ▶ **Message provenance:** The system helps identifying problematic communication patterns, and track the causality of events within the system.
- ▶ **Aggregated views:** The current state may not provide sufficient information for deep bugs, requiring aggregated views for nodes and their corresponding messages.

Inter-Display Connection

- ▶ Text log and 3D visualization are used in conjunction with each other
- ▶ Brushing and Linking for highlighting selected items on both displays
- ▶ Linking of currently visible time frames for keeping both screens synchronized

