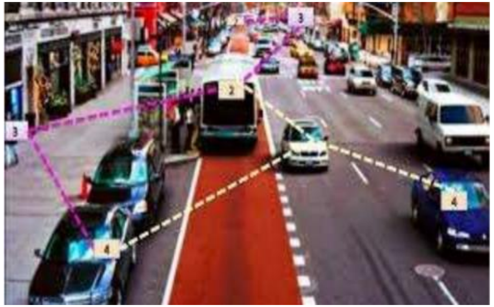
An Autonomous and Dynamic Coordination and Discovery Service for Wide-Area Peer-to-peer Publish/Subscribe

Kyoungho An, Shweta Khare, Aniruddha Gokhale and Akram Hakiri
kyoungo.an@rti.com,{shweta.p.khare,a.gokhale}@vanderbilt.edu,
akram.hakiri@gmail.com

VANDERBILT UNIVERSITY

Data Distribution Needs for IIoT

- IIoT applications are highly distributed and mission-critical
- Requiring:
  - Geographically distributed data dissemination
  - Strict Quality of Service (QoS) guarantees:
    - Reliability
    - Durability
    - Timeliness
    - Security

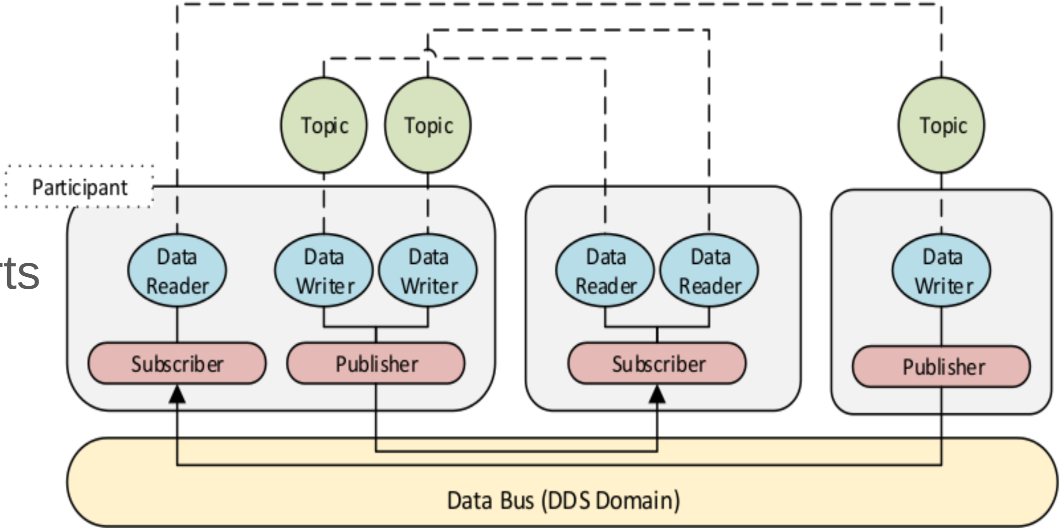


- **Publish/Subscribe** communication paradigm is well suited for IIoT application needs as it provides **scalable and decoupled data delivery** among communicating peers.

VANDERBILT UNIVERSITY

OMG Data Distribution Service (DDS): Publish/Subscribe standard for IIoT

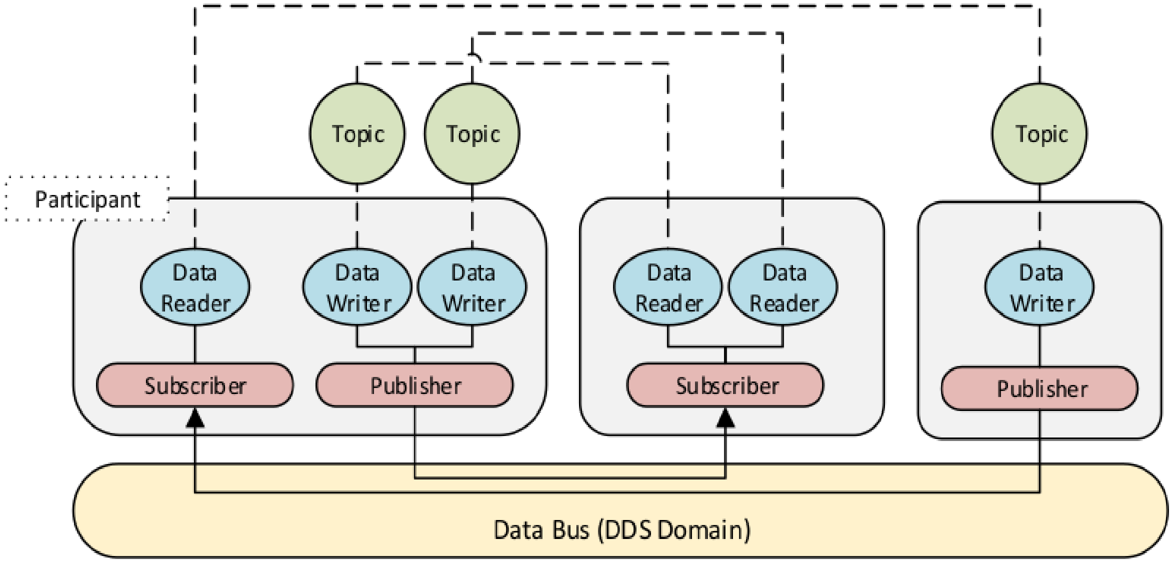
- OMG DDS is a **data-centric, anonymous, topic-based** publish/subscribe standard.
- **Peer-to-Peer** architecture supports **low-latency** and **scalable** data delivery.
- Configurable **QoS policies**:
  - **Reliability, Durability, Deadline, Liveliness, Ownership, Lifespan, History, Resource Limits,** etc.



VANDERBILT UNIVERSITY

OMG DDS: Limitations for WAN-Scale Use

- Current technology limitations restrict the use of DDS to a single LAN:
  - DDS uses **multicast for discovery**
  - **NAT/Firewall** use prevents data distribution across LANs.
  - Existing **broker-based** solutions to bridge DDS LANs:
    - **Not Scalable**: Require **manual configuration**
    - Require **invasive changes** to the application code
    - **Lack autonomous and dynamic discovery and coordination** service to interconnect peers across multiple networks



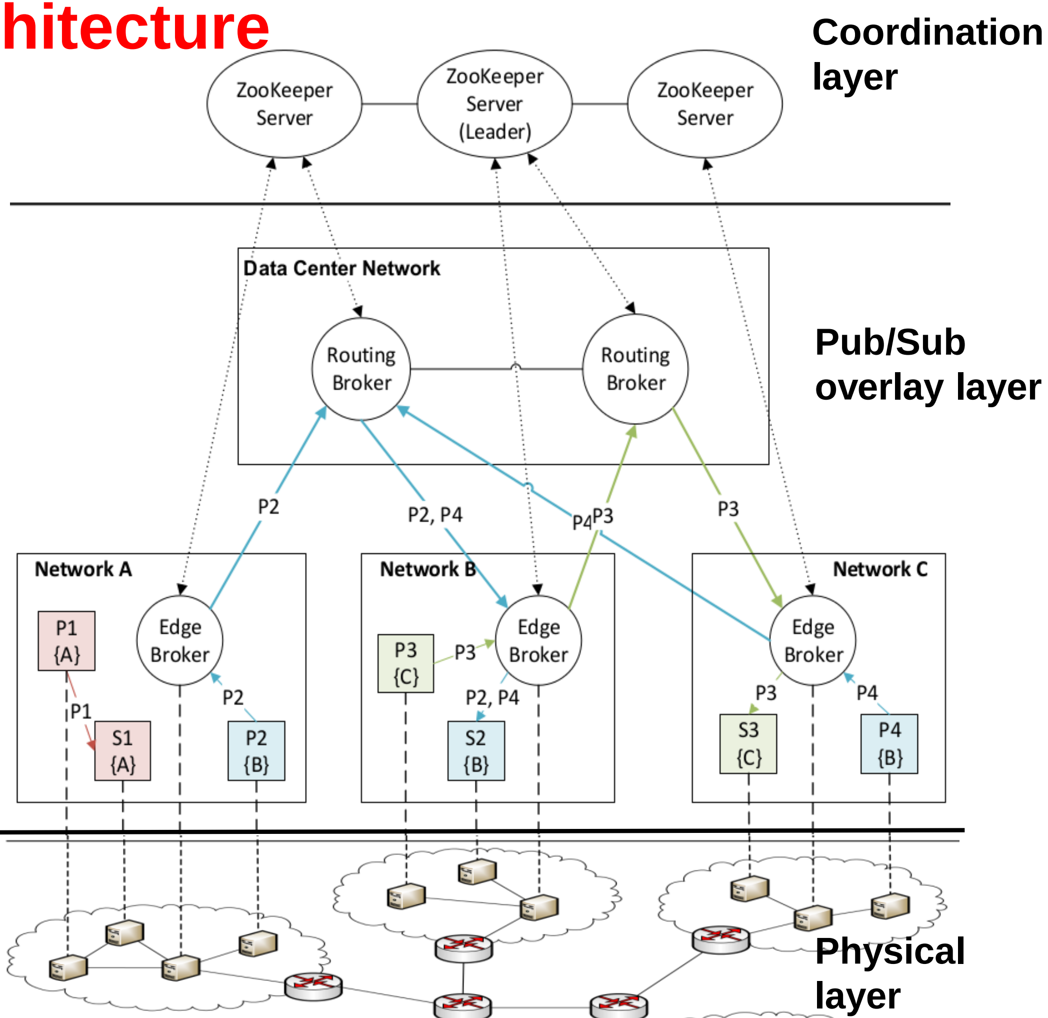
VANDERBILT UNIVERSITY

In Summary

A readily available, rapidly deployable, and non-invasive middleware solution to autonomously discover and interconnect DDS peers at WAN-scale does not exist.

PubSubCoord:Solution Architecture

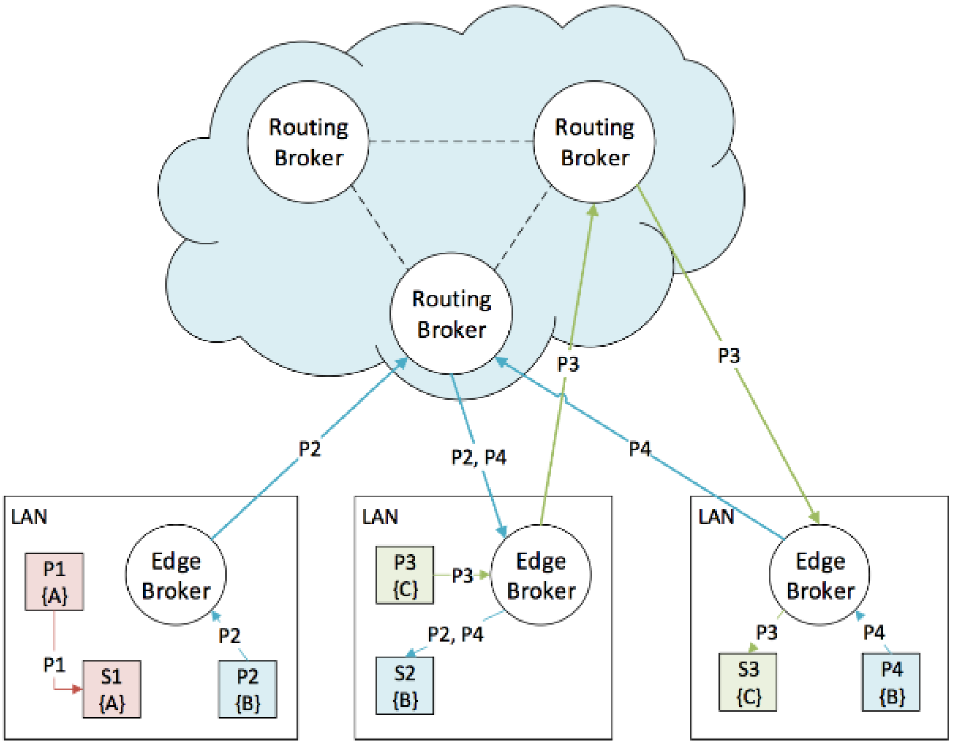
- **2-Level Broker architecture** for low latency (**maximum 2-hop**) dissemination.
  - **Edge Broker Layer:** serves as a **gateway** for locally connected endpoints in a LAN
  - **Routing Broker Layer:** serves as a **mediator** to route data between edge brokers according to assigned and matched topics
- **Coordination layer** is responsible for autonomous discovery and data routing between brokers.



PubSubCoord: Data Dissemination

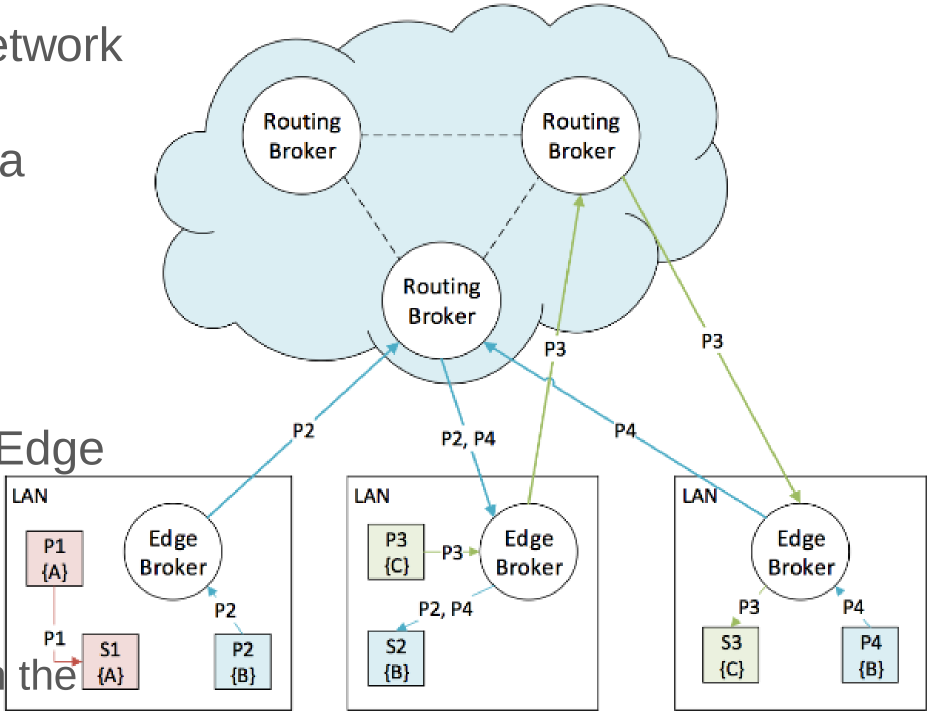
- **Local Communication at the Edge:**
  - ➢ P1 and S1 are interested in topic A. Since they reside in the same network, they communicate via **UDP-based unicast without incurring a hop to the routing broker layer**.

- **Communication across networks via Routing Broker layer:**
  - ➢ P2, P4, and S2 are interested in topic B but are deployed in different networks, so their **communications are routed through a routing broker** that is responsible for topic B.



Benefits of PubSubCoord Design

- ❖ Low-latency (maximum 2-hop) data dissemination over the broker overlay network
- ❖ Load balancing at routing broker layer via elastic autoscaling
- ❖ Easy state maintenance
- ❖ Failed Edge Brokers do not affect other Edge Brokers.
- ❖ Efficient intra-LAN dissemination
  - ➢ traffic that is local is not allowed to reach the routing brokers
  - ➢ local dissemination is handled by the edge brokers themselves thereby avoiding round-trip WAN latencies

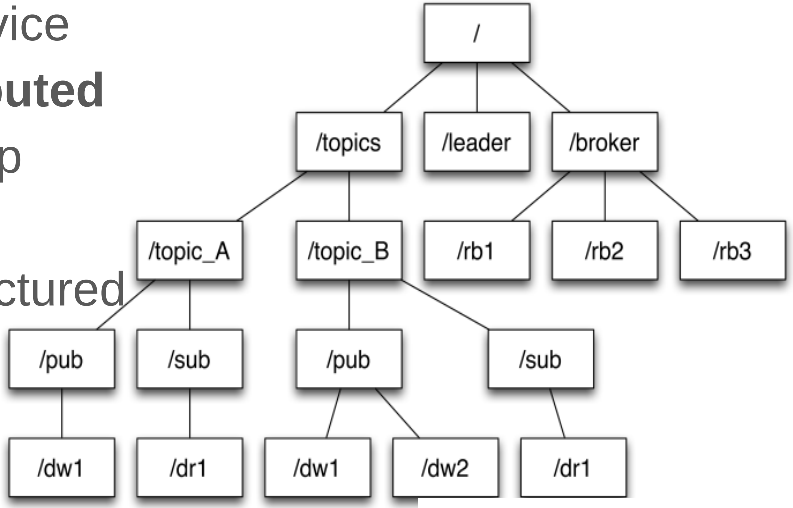


VANDERBILT UNIVERSITY

PubSubCoord Design: Coordination Layer

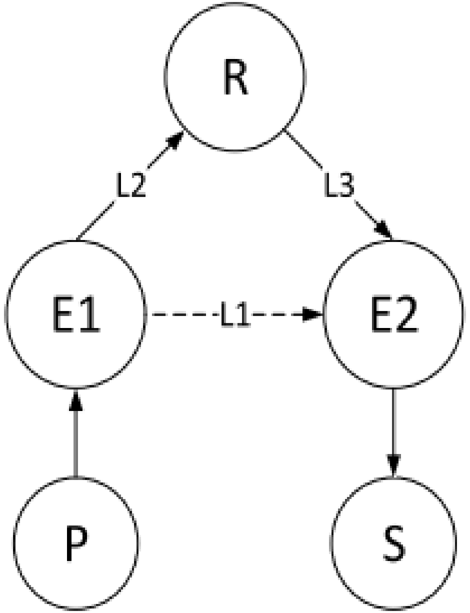
- PubSubCoord uses a coordination layer comprising an **ensemble of ZooKeeper servers,** which help brokers discover each other and build broker overlay networks
- **Zookeeper** is a centralized and replicated service which provides **generic constructs** for **distributed coordination.** Example: Leader election, group membership, locks, etc.
  - **Znodes:** Data Model of Zookeeper is structured like a file system comprising of znodes- **Zookeeper data object** (path and data)
  - **Watch Mechanism:** notifies a client of ZooKeeper of a change to a znode that is being watched by that client.

PubSubCoord Data Model



PubSubCoord Design: QoS Optimization

- In case of **congested, slow or lossy WAN links** over the two-hop route connecting Edge brokers via a Routing broker, PubSubCoord supports **Deadline-Aware overlays**, which **directly interconnect two Edge-Brokers:**
  - **Improves reliability and latency** by providing an additional one hop path, directly interconnecting two edge brokers
  - Leveraged by pub/sub streams that require **stringent assurance and deadline-driven data delivery.**
  - **Uses DDS deadline QoS** that expresses the maximum duration within which a sample has to be updated.

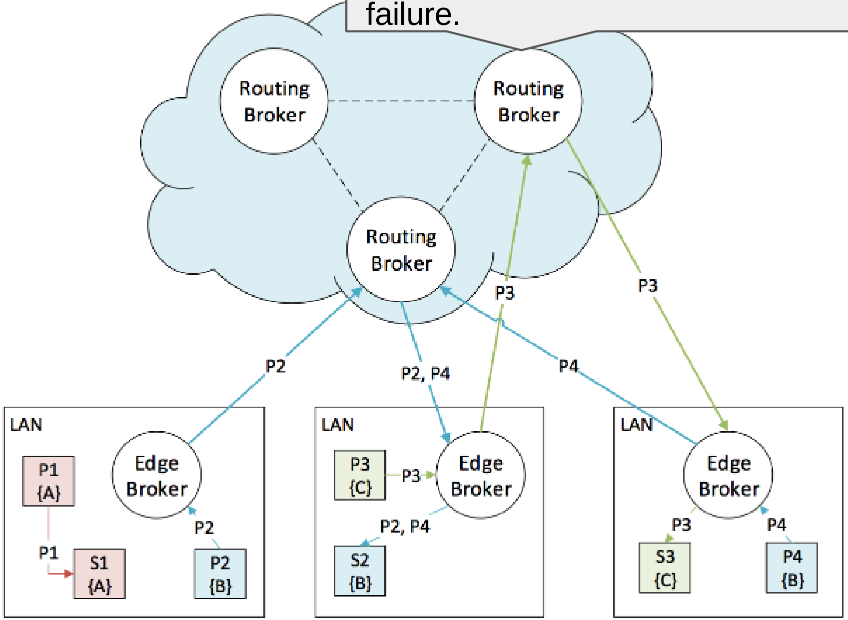


VANDERBILT UNIVERSITY

PubSubCoord Design: Load Balancing & Fault Tolerance

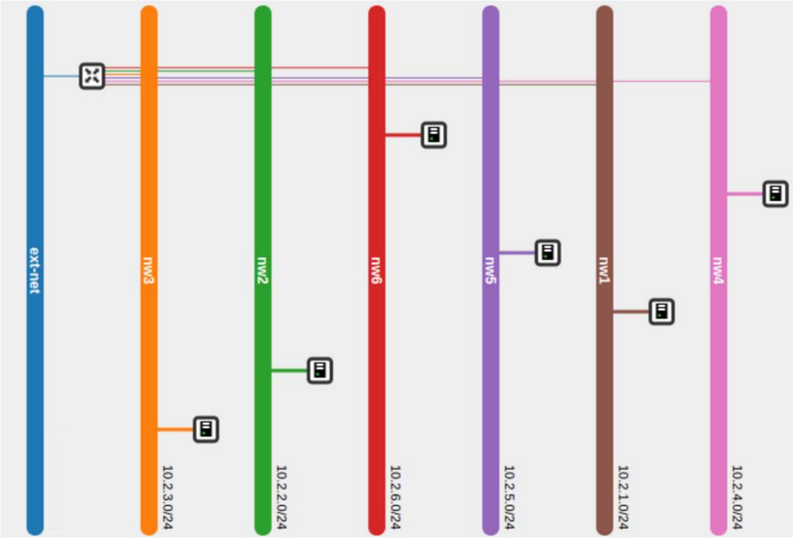
- **Load Balancing:** Leader Routing Broker distributes topics among worker Routing Brokers. Example: **least loaded routing broker** in terms of number of topics, CPU/network utilization, etc.
- **Fault Tolerance:** Leader Routing Broker **reassigns topics handled by a failed broker to another worker** Routing broker to avoid service cessation. ZooKeeper's watch mechanism is used to notify the appropriate edge brokers to update their paths to the right routing broker.

Leader RB performs **load balancing** of topics among worker brokers and **redistribution** of topics on broker failure.



Experiment Setup: Testbed Configuration

- **OpenStack private cloud** comprising **60 physical machines** each with 12 cores and 32 GB of memory.
- **VM configuration:** 1VCPU and 2GB RAM. Edge and Routing broker instances run in their own VM. Multiple publisher and subscriber test applications share a VM.
- Neutron was used to create **120 virtual networks/**LANs.
- **Emulated latencies:** 20 milliseconds roundtrip LAN and 80 milliseconds roundtrip WAN
- **PTP time Synchronization.**
- **RTI Connext 5.1** is used for implementation.



VANDERBILT UNIVERSITY

Experiment Setup: Test application configuration

- ❖ All DDS endpoints are configured with the following **QoS settings:**
  - ➢**RELIABLE reliability QoS:** Reliable data delivery at transport-level
  - ➢**KEEP_ALL history QoS**: Keep all data history in memory
  - ➢**TRANSIENT durability QoS:** Deliver history data for late joiners
  - ➢**LIFESPAN QoS** 60 seconds: Keep data history for 60 seconds

- ❖ Publishers send **64Byte messages every 50 miliseconds**.
- ❖ **5000 messages** are sent per publisher. Only use values only after 1,000 samples since the latency values of initial samples are not consistent due to coordination and discovery overhead
- ❖ **End-to-end latency** was calculated as the time difference between the send timestamp at the publisher and reception timestamp at the subscriber.

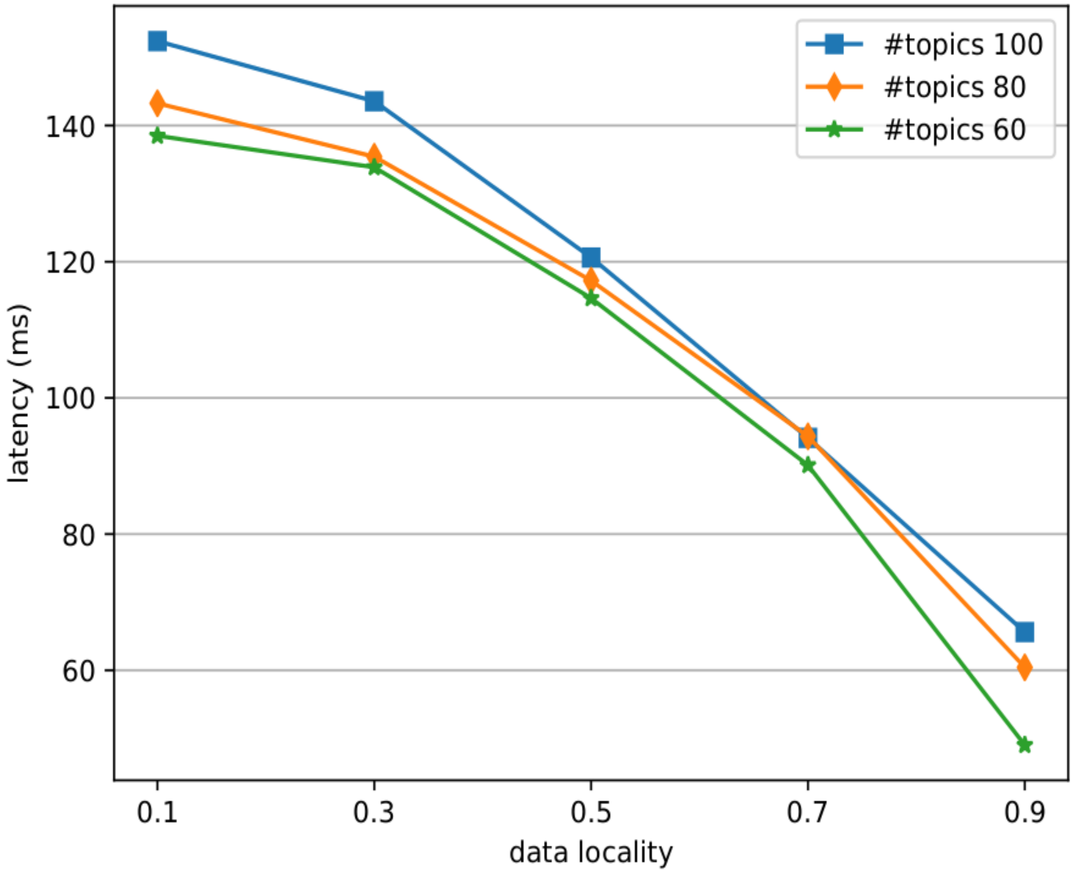
VANDERBILT UNIVERSITY

Data Locality Experiment

Edge Broker layer is responsible for dissemination of local traffic thereby preventing WAN latencies.

- ❖ Measure end-to-end latency for different values of **data locality**:
  - ➢ fraction of topics in an isolated network which are local to the network and do not have interested subscribers in another network

- ❖ As the data locality increases, the end-to-end dissemination latency decreases



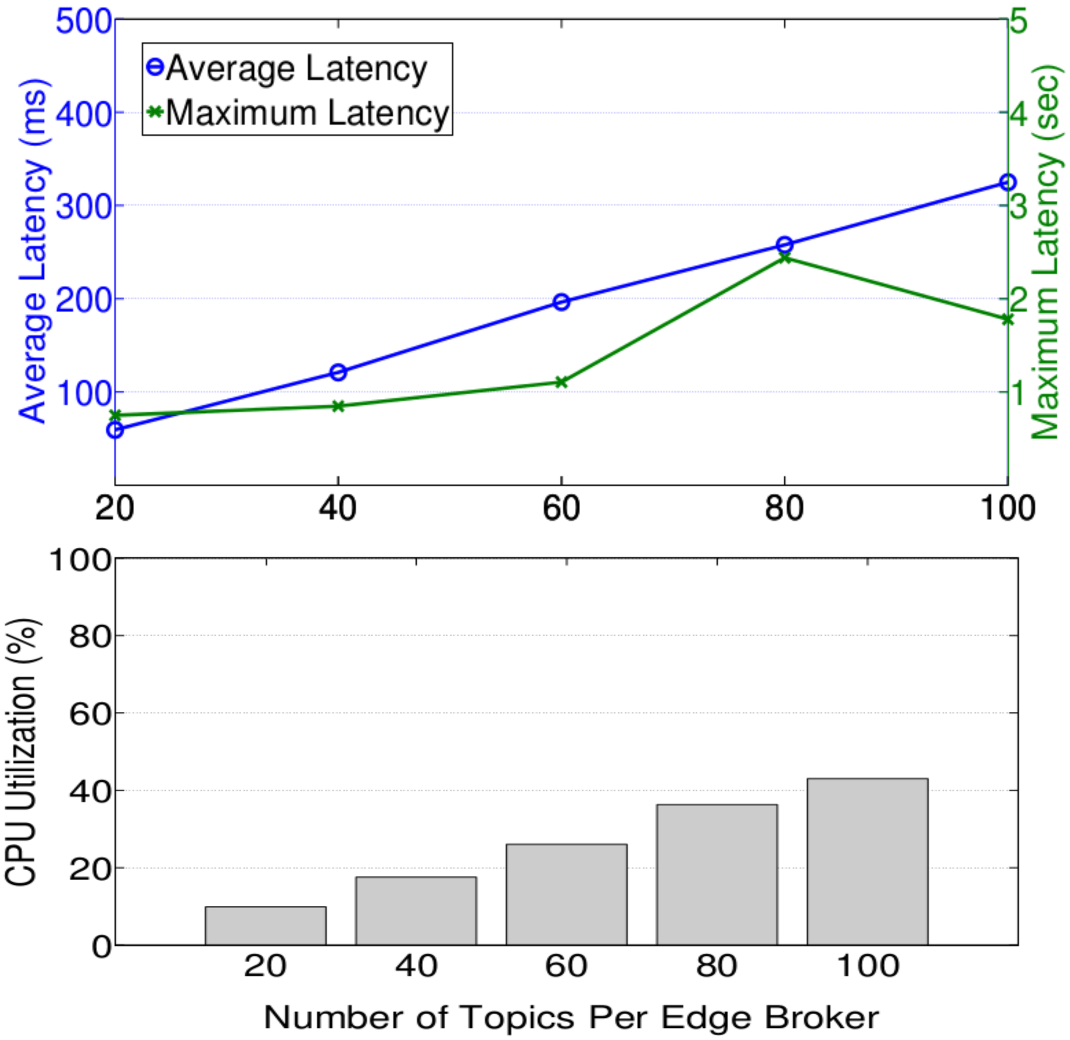
Scalability Experiment Setup

- 400 VMs were used: 120 VMs for edge brokers; 40 VMs for routing brokers; 40 VMs were used for publishers and 200 VMs for subscribers.
- VM for publishers hosts 25 publisher applications. VM for subscribers hosts 50 subscriber test applications. Thereby, creating a total of 1000 publishers and 10,000 subscribers
- Subscribers in each network are interested in 100 topics out of 1000 topics in the system.

VANDERBILT UNIVERSITY

Scalability Experiment: Number of Topics

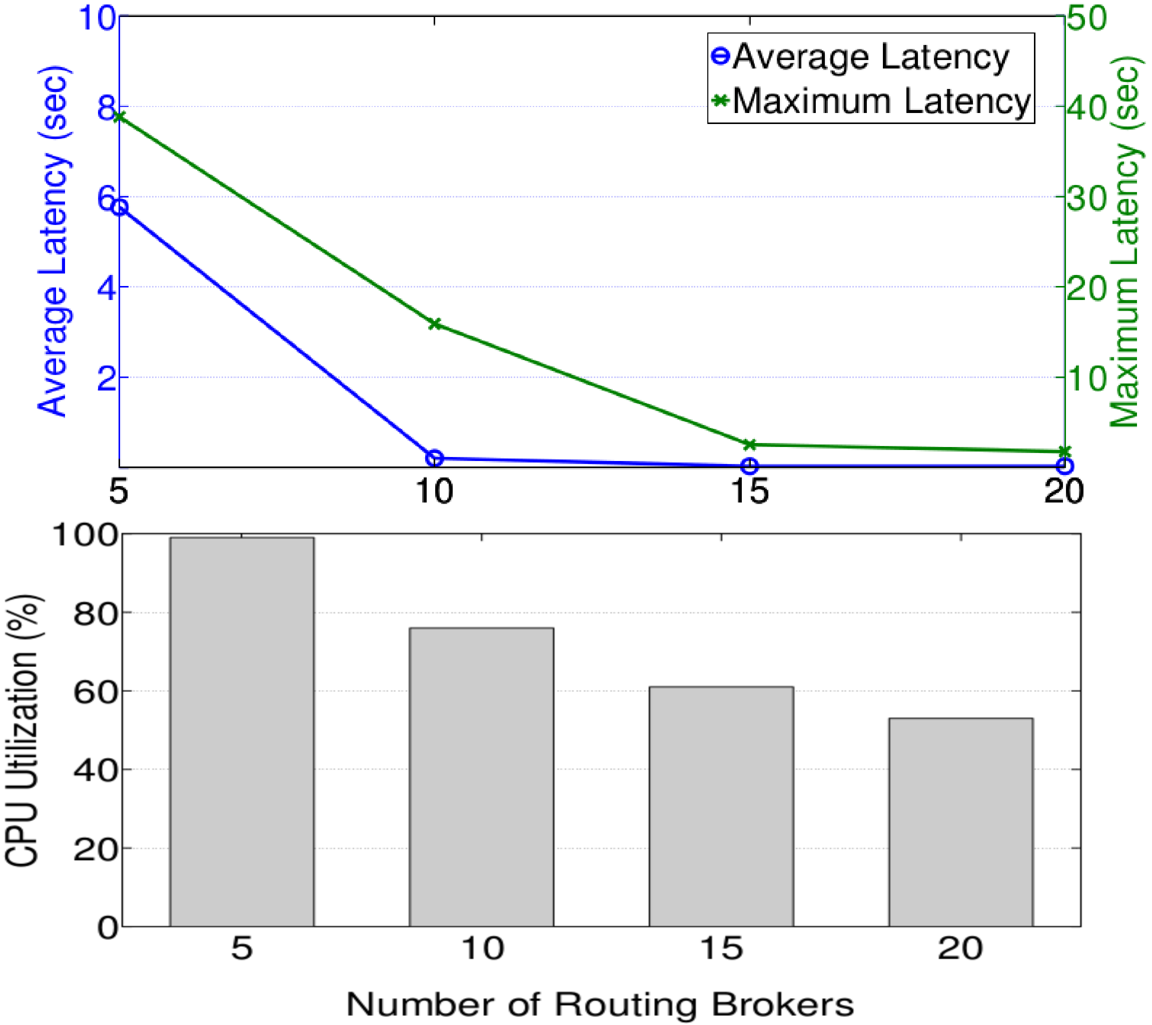
- The computation overhead and end-to-end dissemination latency grows linearly with the number of adopted topics at the Edge Broker.



Scalability Experiment: RB Load Balancing

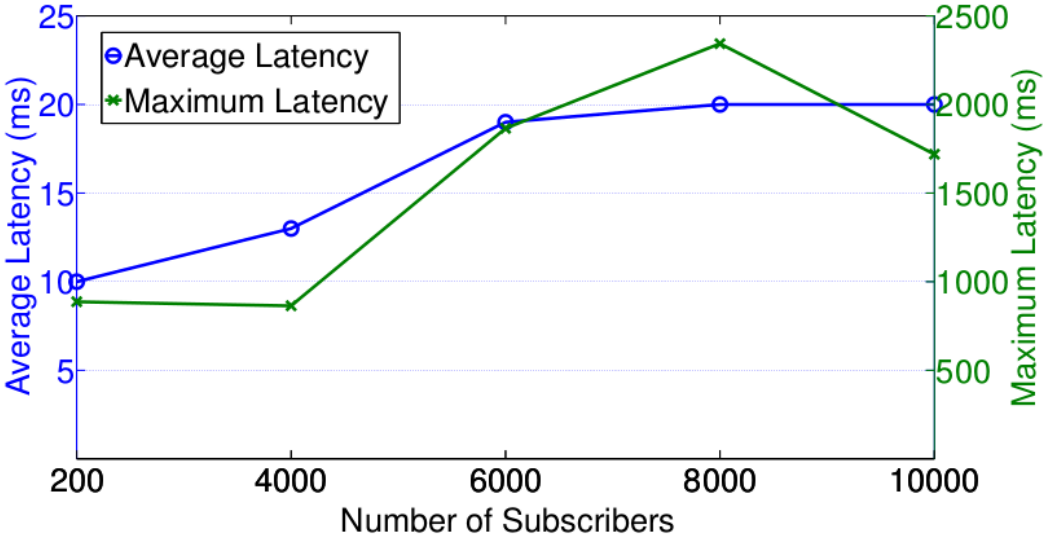
Our solution supports load balancing at the Routing Broker Layer.

- When there are 5 instances of Routing Brokers, the CPU of the routing brokers becomes saturated and latency gets adversely impacted.
- Latency values improve on scaling-up the number of routing brokers to 10.

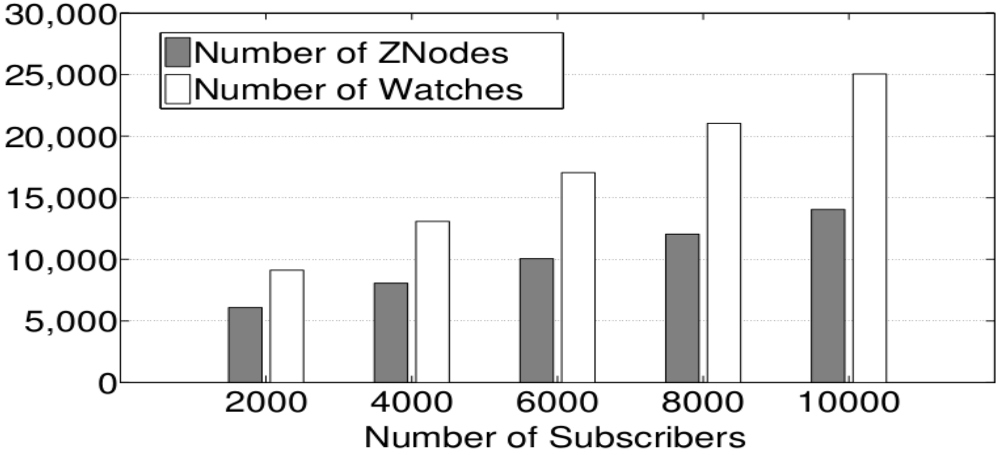


Scalability Experiment: ZK Coordination

To evaluate the scalability of ZK based centralized coordination, the number of simultaneously joining subscribers is increased from 2,000 to 10,000 in steps of 2,000.



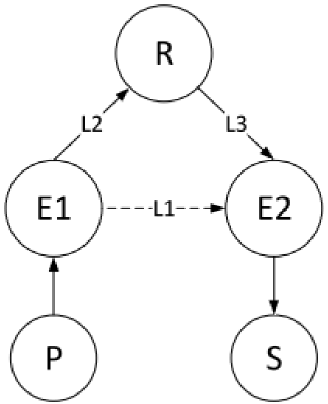
- Time taken by ZK server to respond to a client request, increases from **10ms to 20ms** with increasing number of subscribers.
- Number of znodes and watches increases as the system scales.
- **Overhead of ZooKeeper based centralized coordination service remains acceptable even at scale.**



Deadline Aware Overlay Experiment:

Deadline Aware Overlays are used for topics which have stricter data delivery requirements in case of congested, lossy or slow WAN links.

- ❖ Compare the dissemination latency and broker overhead for deadline-aware multi-path vs single-path overlays under different WAN link configurations:
  - ➢ **A: 30ms delay and no packet loss**
  - ➢ **B: 250 msec delay and 1% packet loss**

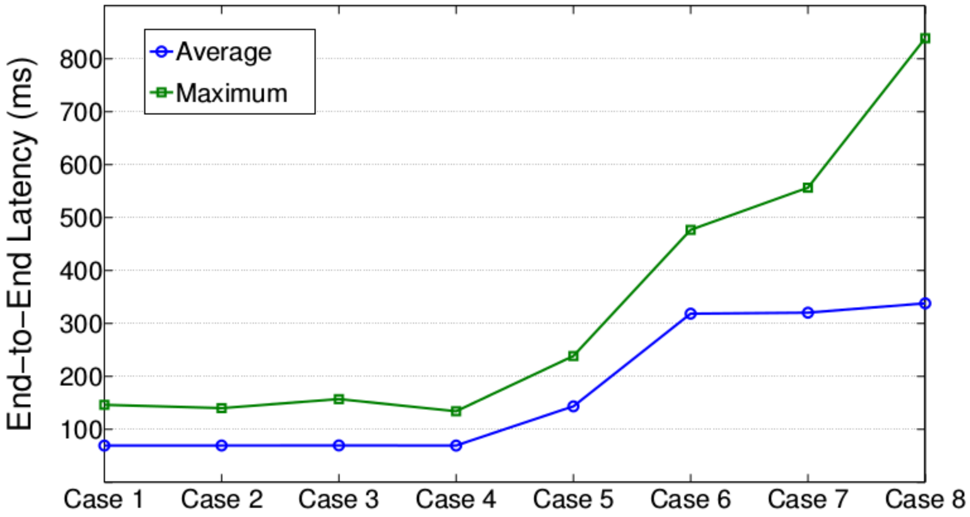


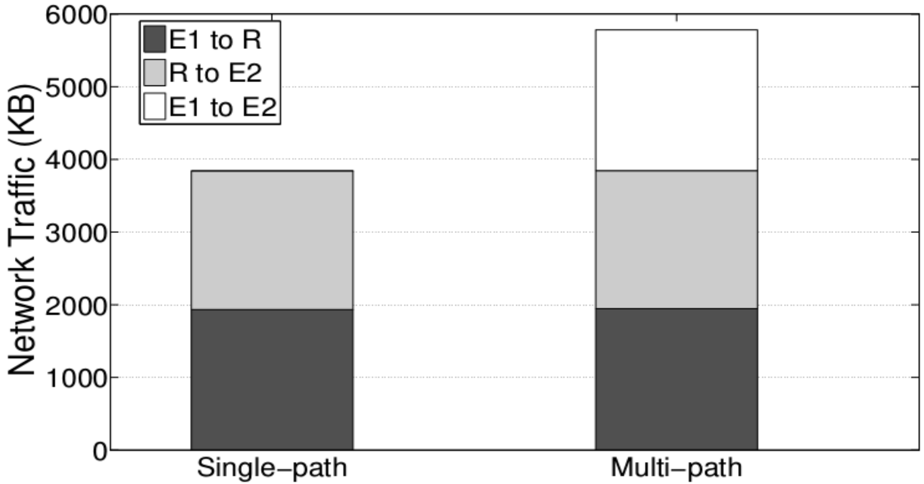
Test Cases	**L1**	**L2**	**L3**
Case 1	A	A	A
Case 2	A	A	B
Case 3	A	B	A
Case 4	A	B	B
Case 5	B	A	A
Case 6	B	A	B
Case 7	B	B	A
Case 8	B	B	B

VANDERBILT UNIVERSITY

Deadline Aware Overlay Experiment:

- Test cases 1 to 5 for multi-path overlays perform better than single-path overlays.
- Topics with strict delivery requirements can benefit from deadline-aware overlays under adverse WAN link conditions.
- Maintaining multi-path overlays impose additional computation and network transfer overhead at the edge broker





Conclusions

- Presented PubSubCoord which is an autonomous and dynamic coordination and discovery service for WAN-scale DDS applications
- PubSubCoord disseminates data in a scalable manner for systems having many pub/sub endpoints and topics across multiple networks.
- Centralized coordination service like ZooKeeper can serve as a pub/sub control plane for large-scale systems
- Configurable QoS supported by DDS can be used for low-latency data delivery in WANs by building multipath overlays
- **Future work**
  - Effective load balancing algorithms at routing broker layer
  - Experiment with IoT systems (smart transportation)
  - Support for other pub/sub technologies
  - Interoperability
  - Integration with SDN and Time Sensitive Networking

VANDERBILT UNIVERSITY

Thank you.

VANDERBILT UNIVERSITY