Transport layer multicast via logical tree networks for deterministic packet delivery in event-driven systems

Martin Landsiedel, Architecture and Network Solutions, Continental Automotive, Wetzlar, Germany | 0009-0003-9577-8244

Introduction

To improve the efficiency of event-driven publish/subscribe systems, a transport-layer multicast using logical tree networks is examined. Instead of sending parallel unicasts to all subscribed clients, packets are routed across the branches of the tree. That results in logarithmic scaling and deterministic packet delivery.

Today's vehicle architecture consists of high performance computers (HPCs) implementing event-driven software. Ethernet continues to gain popularity for in-vehicle communication [1], along with service-oriented protocols like SOME/IP [2]. Efficient communication is crucial to cope with the increasing amount of data.

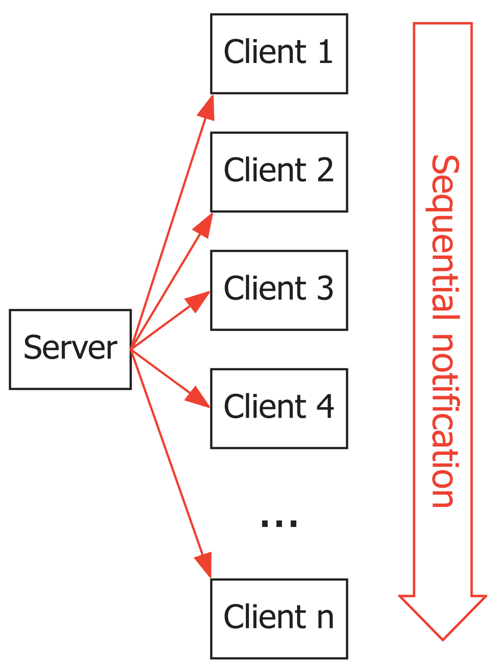

Figure 1. Traditional, linear notification of clients. The server sends a packet to every client sequentially. This results in linear scaling of packet delivery time as well as network and processor load.

As Figure 1 shows, traditional notification of clients via parallel unicasts results in linear scaling of delivery time and network load. With the tree network however, the network and processor load is distributed from the server to the individual clients.

The proposed network is implemented logically on the transport layer to operate transparently to applications and to avoid overhead from underlying transport layer protocols. Required features e.g. of TCP can be reimplemented in an efficient way.

The structure of the tree network is inspired by the TreeP network [3] targeted to peer-to-peer networks for distributed computing. It is modified to be centrally managed by the server, to use a different routing scheme and to strictly form a binary tree. It can be evaluated whether it is sensible to integrate some of the defined management algorithms into the proposed network.

Tree network construction and routing

The tree network is constructed logically on the transport layer by the server, which is placed at the root. Multiple trees can exist in parallel and ideally one tree is created for each multicast group.
The tree network can operate transparently to the application layer and can span multiple internet layer networks. The overhead of underlying transport layer protocols is avoided by optimizing their characteristics for the tree structure.
The constructed network forms a complete binary tree resulting in the minimum tree height for a given number of nodes as displayed in Figure 2 [4]. Each client is assigned an ID used for addressing. They are numbered like a binary search tree, allowing for decentral routing without routing tables.

A packet from the server to all or a subset of all connected clients is transmitted across the branches of the tree. At each inner node, the packet is duplicated and forwarded to one or both child nodes as needed. It results in parallel operation of the branches and each edge carrying a packet only once. A deterministic packet delivery order within each branch is evident, enabling client prioritization.

The total packet delivery time T in a tree of n clients can be calculated as $T(n) = \mathcal{O}[\log_2(n+1) - 1] = \mathcal{O}[\log n]$.

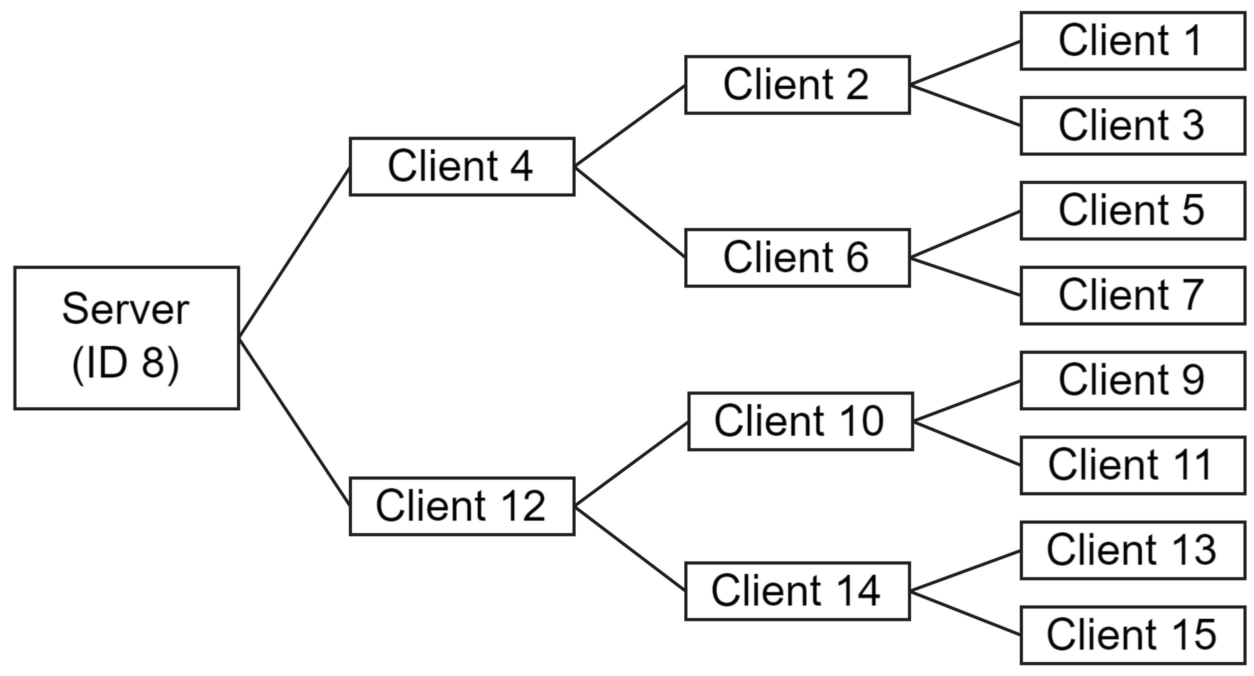

Figure 2. The tree network forms a complete binary search tree. The complete binary tree results in a minimum tree height for a given number of nodes. The search tree numbering allows for decentral routing without the need for routing tables.

Simulation of a star topology

A physical network in star topology is simulated using the cnet network simulator [5]. The network uses a central router to which each client is directly connected via an exclusive 100 Mbps, 1 ms latency Ethernet link. Packets are sent in traditional linear order as well as via a statically defined tree network for comparison.
The packet delivery time to the first and last client depending on the number of addressed clients is shown in Figure 3. It can be seen that the tree network scales logarithmically opposed to the linear communication. Packets to the first and last client are delivered within a short time difference. The linear network can deliver the first packets within constant time, but scales linearly after that.

In summary, according to this simulation, the tree network is suitable for larger networks and/or for networks that need packet delivery to all clients within a short time difference or a deterministic packet delivery order.
It is planned to verify these results outside of a simulation.

The simulation files are uploaded to [6].

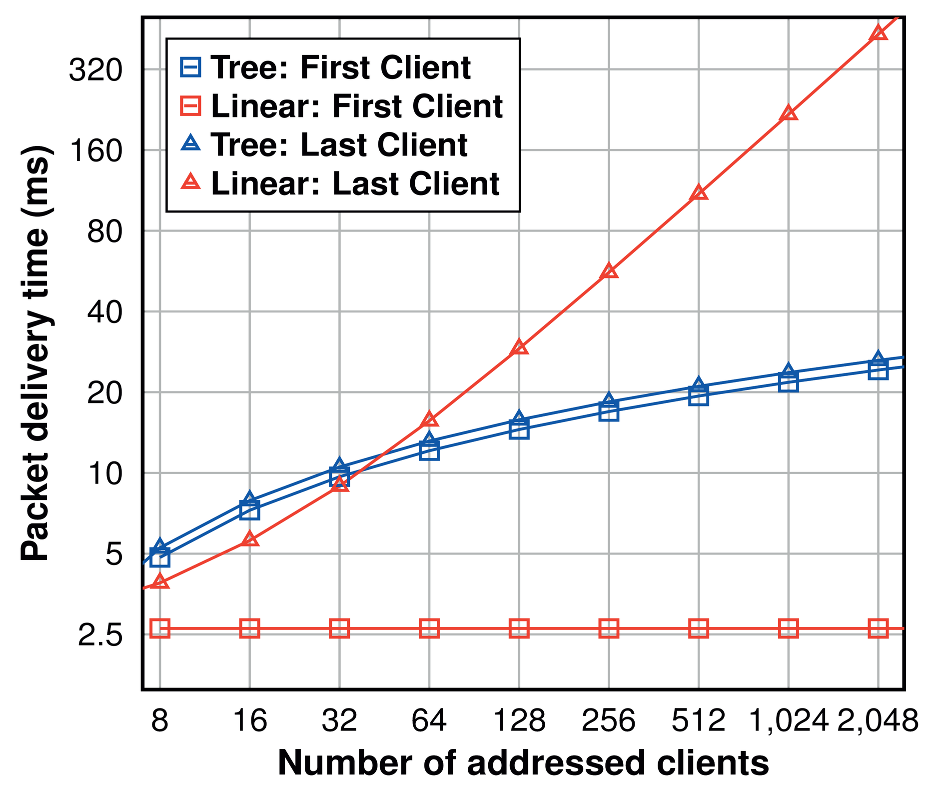

Figure 3. The packet delivery time to a specific client depending on the number of addressed clients in a star topology shows the logarithmic versus linear scaling of the tree network versus linear communication.
The tree network deliveres all packets within a short time difference. The linear communication deliveres the first packets in constant time but scales linearly after that.

Implementation of a bus topology

An experimental implementation of the tree network is built for Node.js as application on top of WebSocket. All client instances are executed on the same Ubuntu machine and communicate via the loopback interface. For comparison, the uWebSockets.js server is used. The measurement is repeated 25 times, sending 20 packets each time.
The packet delivery time to the eleventh and last client depending on the number of addressed clients is shown in Figure 4. The results of the measurements are averaged.

Compared to the simulation results in Figure 3, the results seem to be reversed. Now, the linear communication scales linearly as expected and delivers packets to the first and last clients within a short time difference. The tree network on the other hand delivers packets to the first clients in constant time and scales linearly after that. In general, the tree network takes almost twice the time to deliver the last packet.
The reason for the short time difference of the linear communication seems to be a varying notification order. The average of every node is therefore roughly the same.
The reason for the linear scaling of the tree network and its higher packet delivery time probably is a bottleneck in the parallel execution of the client instances. It is limited by the single executing machine and the Node.js event loop.

A simulation and/or implementation of the bus topology on multiple machines is planned to verify the results.

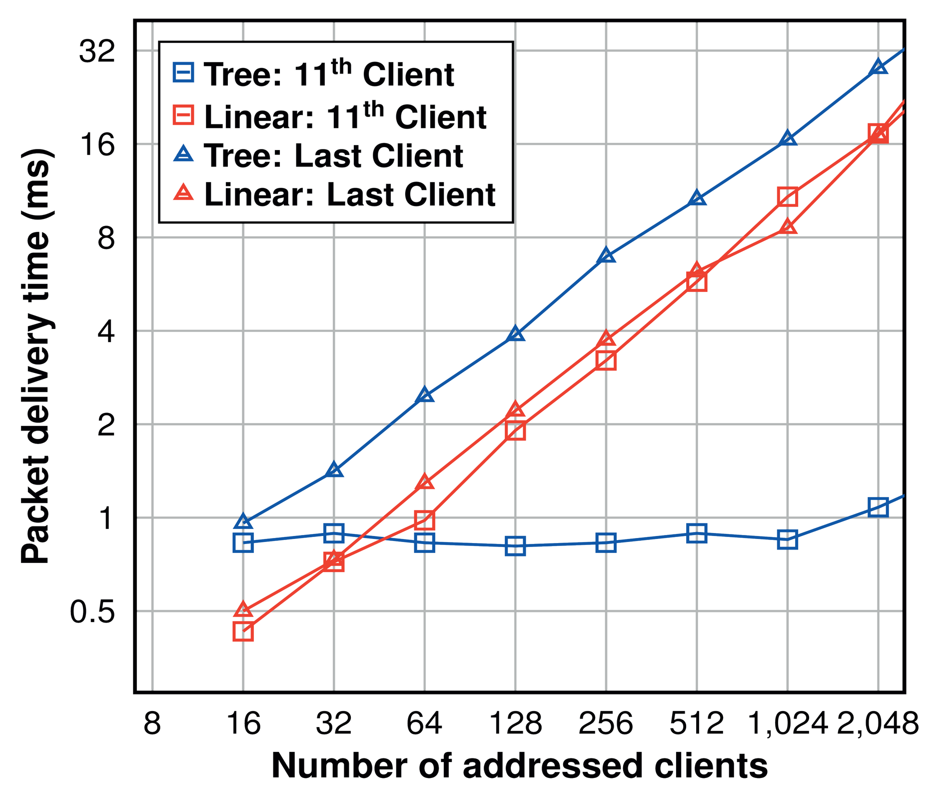

Figure 4. The packet delivery time to a specific client depending on the number of addressed clients in a bus topology shows different results. Here, the tree network can deliver the first packets in constant time, but scales linearly like the traditional linear communication. This deviation is probably caused by a bottleneck in the implementation of the tree network.

Conclusion

The experimental results as well as the theoretical aspects show that the proposed tree networks can provide benefits for multicast communication in event-driven automotive systems.
They can provide a deterministic packet delivery order within each branch of the tree, giving the ability to prioritize clients.
According to the simulation, they deliver packets to the first and last clients within a smaller time difference and they scale logarithmically. However, the experimental implementation in bus topology shows different results and therefore requires further investigation.
In both cases, the tree network proves to be disadvantageous for small networks when considering only the packet delivery time, as overhead causes an initial delay.

Outlook

Further research on this topic is planned to provide additional measurement results, especially for mixed network topologies as in real vehicular networks. It also includes measuring the actual network and processor load of the server and all clients.

Another focus is on the integration of the tree network into the network stack. For that, a TCP- and UDP-compatible transport layer protocol has been developed already to provide measurement results without the overhead of additional underlying protocols. It should be tested together with service-oriented application protocols like SOME/IP.

Literature cited

[1] L. Lo Bello, G. Patti and L. Leonardi, "A Perspective on Ethernet in Automotive Communications—Current Status and Future Trends" in *Applied Sciences 13, no 3*, 2023, article 1278. DOI: 10.3390/app13031278

[2] L. Völker, J. Villanueva, J. Schäferling, "Is SOME/IP the right solution for the next 10 years of vehicles?" in *10th Automotive Ethernet Congress*, Mar. 2024. [Online]. Available: https://some-ip.com/papers/2024-AEC_Ampere_Technica Engineering_-_SOMEIP-next-10-years_final_v1.1a.pdf (visited on 2025-04-30).

[3] B. Hudzia, M.-T. Kechadi and A. Ottewill, "TreeP: A Tree Based P2P Network Architecture" in *2005 IEEE International Conference on Cluster Computing*, Sep. 205, pp. 1–15. DOI:10.1109/CLUSTR.2005.347022

[4] P. E. Black "Complete binary tree" in *Dictionary of Algorithms and Data Structures*, P. E. Black, Nov. 27, 2019. [Online]. Available: https://xlinux.nist.gov/dads/HTML/completeBinaryTree.html (visited on 2025-04-02).

[5] C. McDonald, *The cnet network simulator (v4.0.4)*. [Online]. Available: https://teaching.csse.uwa.edu.au/units/CITS3002/cnet/index.php (visited on 2025-04-30).

[6] M. Landsiedel, *Simulation of logical transport layer tree networks*, Github Repository: https://github.com/marland4a/tree-network-simulation

Acknowledgments

I thank Lukas Trümper and Thomas Stetz for their support during and after my bachelor's thesis, through which I began researching this topic.
A big part of this research was conducted as employee of Continental Automotive Technologies GmbH, Wetzlar (Future Aumovio).

Email: marland.arbeit@mail.de Phone: +49 163 5190128