Application-Layer Overlay Networks for Communication-Based Train Control Systems

Mei Ya Chan[1], Sami Baroudi[1], Joseph Siu[2], Jörg Liebeherr[1]

[1] Department of Electrical and Computer Engineering
University of Toronto

[2] Thales Canada Transportation Solutions
Toronto

***Abstract*—Communication-Based Train Control (CBTC) is a train control system that uses radio communication to exchange information between trains and a central control station. As an inherently time-critical application, CBTC imposes stringent requirements on communication availability and latency. In this paper, we propose to enhance the communication availability in CBTC systems with Application-layer Service Overlay Network (ASON) technology. We show that ASONs can facilitate the switchover from the CBTC infrastructure network to a backup network with alternative communication paths. We employ Application-layer Service Overlay Network (ASON) technology to facilitate the switchover between the CBTC infrastructure network and a backup network. We conduct measurement experiments to evaluate whether the switchover delay to an ASON over an LTE network can satisfy the delay and bandwidth requirements of a time-critical train control application.**

***Index Terms*—CBTC System, ASON, Hybrid Network.**

I. Introduction

The main objective of a train control system is to ensure safe distances between consecutive trains [6, 11]. A Communication-Based Train Control (CBTC) system is an automated train control system using bidirectional train-to-wayside communication to ensure the safe operation of trains [4, 7, 12, 14]. In a CBTC system, each train continuously communicates its location, direction, and speed to a central control station. Through these communications, the central control station knows the location of all trains moving on the track and assigns each train a permission to move through a designated section of the track at a specified speed. Since train control messages are time-critical, communications in CBTC systems are subject to stringent latency and availability requirement to assure railway safety, e.g., as specified in IEEE 1474 [1].

Fig. 1 presents a sketch of the communication architecture of a CBTC system. A train equipped with wireless radios connects to wayside access points (APs), which have been deployed along the track. For redundancy, trains are typically equipped with two radios, one in the front and one in the rear of the train that each select an AP independently. A wired infrastructure network provides the connectivity between wayside APs and a central control station.

In terms of latency, IEEE 1474 [1] establishes the performance and functional requirements of a CBTC system,

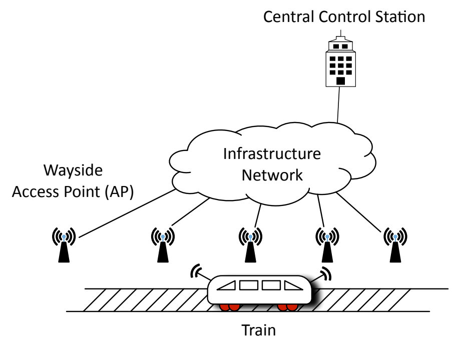


Fig. 1: CBTC system.

including that the typical bidirectional communication delay should range from 0.5 to 2 seconds. If consecutive train control messages are lost or the delay exceeds an acceptable limit, trains in transit are forced to halt to ensure safe operation. To avoid such service interruptions and to increase the line capacity, CBTC systems seek to maximize the availability of communication by a redundant deployment of wayside APs, such that the radios of a train are always within range of two or more wayside APs. More recently, CBTC designs have considered to augment the infrastructure network by additional communication modalities, to ensure availability of the CBTC system in situations when a train and the central control station cannot communicate through the infrastructure network, resulting in a hybrid network design [2]. Examples of alternative modalities are cellular (LTE) networks and ad-hoc networks between trains.

In this paper, we evaluate the use of Application-layer Service Overlay Network (ASON) technology to support a hybrid network design for a CBTC system. An ASON is a virtual overlay network built on top of one or more existing substrate networks that operate at the application layer [3]. An advantage of an application-layer approach is that there is no need for special-purpose hardware and no need for protocol compatibility at lower layers.

The role of the ASON is to switch communication between trains and the central control station from the infrastructure network to an alternate substrate network, akin to vertical handoffs in heterogeneous networks [13]. We explore the open question whether application-layer network protocols are suitable to complete handoffs within the timing constraints of a CBTC system. Enabling time-critical applications over ASON faces several challenges. First, forwarding data by intermediate nodes at the application-layer as opposed to the data-link and network-layer in legacy networks incurs additional delays. Second, an ASON must adapt to nodes joining and leaving the network topology.

In this paper we investigate the ability of a hybrid network running as an ASON to satisfy the latency requirements of a safety-critical train control system. We explore if and to which degree an ASON can ensure continuous communication between the central control station and the trains, and improve the reliability and resilience of the train control system. We present switchover mechanisms for a hybrid network that builds an ASON with an LTE substrate network to enhance the existing infrastructure. We evaluate the switchover delays in measurement experiments that involve an ASON over an LTE substrate network. We show that ASON protocol parameters can be set such that a switchover to an LTE backup network is achieved within a few hundred milliseconds in most cases.

The remainder of the paper is organized as follows. In Sec. II we present the communication model and network environment. In Sec. III we discuss components of an ASON software system. In Sec. IV we illustrate the usage of ASON to facilitate a switchover. In Sec. V we present measurement experiments that evaluate the latency and bandwidth of an ASON solution over an LTE network. We present brief conclusions in Sec. VI.

II. Communication Environment

We consider a CBTC system with a frequency hopping spread spectrum (FHSS) based infrastructure network as shown in Fig. 1. The train and the central control station execute an application-layer protocol for the exchange of control information, which is executed between the vehicle on board controller (VOBC) of a train and an application server at the control station. In Fig. 2 we illustrate the steps of the protocol.

A VOBC initiates communication with the application server by sending a connection request. The server replies to the request with an initial sequence number. Then, the server broadcasts a polling message, which echoes the initial sequence number. The broadcast message identifies the VOBC for which the message is intended. Upon receipt, the VOBC issues a reply, which contains the sequence number of the polling message. The polling is repeated in cycles with a length of about 500 ms. If the server does not receive a response within a polling cycle, it sends another polling message with the same sequence number. Broadcast polling messages are received by both on-board radios of a train. A VOBC sends two replicas for each reply message to the application server. One replica is sent from the front and one replica is sent from the rear of the train.

Fig. 3 illustrates potential failover scenarios of a CBTC system in a hybrid network. The additional network components, compared to Fig. 1, are a cellular network and ad-hoc

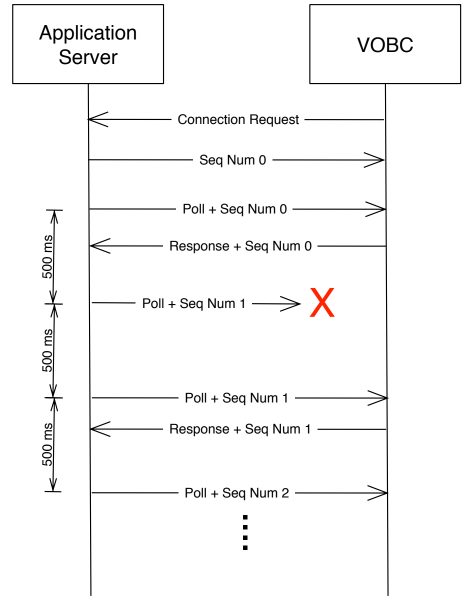


Fig. 2: Polling protocol between application server and VOBC.

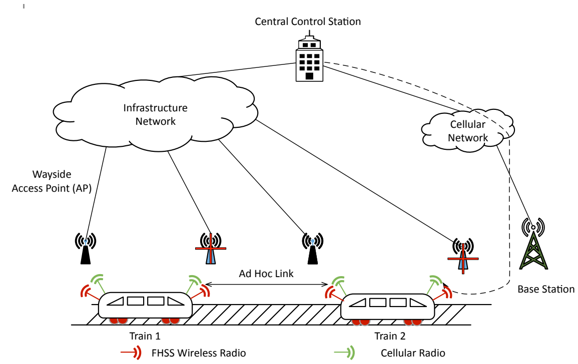


Fig. 3: Communication paths in a hybrid network.

radios deployed on trains. The cellular network and ad-hoc networking capability can provide backup during communication failures. Consider Train 2 in Fig. 3 and suppose that it is unable to communicate with the central control station. If trains are equipped with ad-hoc radios, Train 2 can set up an ad-hoc link to another train (here: Train 1). If Train 1 can access the infrastructure network, it can relay messages between Train 2 and the central control station. In this case, the FHSS wireless radios need to operate simultaneously in infrastructure mode and in ad-hoc mode, or another FHSS wireless radio is required for ad-hoc networking between trains. If Train 2 has an LTE radio interface, which provides access to the Internet, the cellular network can provide another alternative path to the central control station.

III. ASON Software System

ASONs are built and maintained by a software system that enables applications to discover peer nodes, to join and leave an overlay network, and to exchange messages across an overlay network. Each overlay network is viewed as a collection of peer nodes with a unique overlay identifier. The

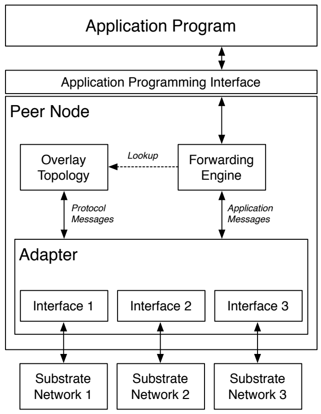


Fig. 4: Components of an ASON peer node.

overlay identifier is associated with configuration attributes that determine the types of protocols, substrate networks, and security properties of the overlay network. An overlay network is created implicitly by the first peer node that joins an overlay with a specific overlay identifier. If an overlay network becomes partitioned, e.g., when nodes are not within transmission range of any other node in the overlay network, each partition operates as an independent overlay with identical overlay identifiers. Hence, when the partition is repaired, e.g., peer nodes moving closer together, the partitions can merge to again form a single overlay network.

An overlay network executes a protocol that establishes a network topology, such as a full mesh, a tree topology, or structured topologies, such as a hypercubes or a triangulation. A peer node in the overlay network only communicates with its neighbors in the overlay topology. Peer nodes perform hop-by-hop forwarding to deliver messages between any two nodes in the topology. When using a structured topology, the logical addresses of peer nodes can be used to determine how to forward a message without the need for a routing protocol.

An application program uses an overlay network by issuing calls to an API that initiate a peer node with given configuration attributes, join and leave an overlay network, or send and receive *application messages*, where send operations can be one-to-one (unicast), one-to-many (multicast), or one-to-all (broadcast).

In Fig. 4 we show the main components of a peer node and its interactions with an application program and substrate networks. While the names and detailed behavior of individual components may vary across different ASON designs, the basic architecture and functionality will be similar as shown in the figure.

A peer node performs node discovery and topology maintenance by exchanging periodic control messages, called *protocol messages*. Peer nodes are running on the application server at the central control station and on the VOBC of each train. Since the central control station is always up and running, we may assume that the application server is permanently connected to the overlay network. On the other hand, VOBCs dynamically join and leave the overlay network. The process of a VOBC joining an overlay network involves

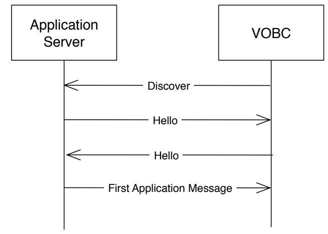


Fig. 5: Operations for joining an ASON.

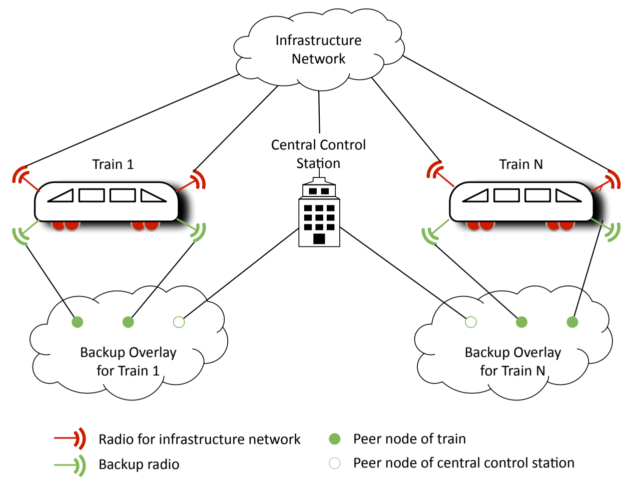


Fig. 6: Backup overlays in a hybrid CBTC system.

an exchange of protocol messages as shown in Fig. 5. First, the VOBC of a train sends a Discover message to the application server. If the application server elects to make the VOBC a neighbor in the overlay topology it sends a Hello message. The VOBC confirms the peer relationship by replying with a Hello message. At this point, the peer relationship is established, and application messages of the CBTC application can be transmitted.

To maintain and confirm the continued neighborhood relationship in an ASON, peer nodes periodically exchange Hello messages with their neighbors. The time interval between the transmissions of successive Hello messages, referred to as the *heartbeat time*, determines the responsiveness and convergence rate of the overlay network after a change of the overlay topology.

IV. Switchover Using ASON

In this section, we discuss how an ASON can facilitate the switchover process between the infrastructure network and a backup network. The backup network, referred to as *backup overlay*, is an ASON that is established in a decentralized peer-to-peer fashion by peer nodes running at the central control station and the VOBC of the trains.

A. ASON Configuration for CBTC

In the hybrid CBTC system with ASON support, a train has separate radio equipment for the CBTC infrastructure network and the backup overlay. We refer to the radios used for communication in the overlay network as *backup radios*. As in the infrastructure network, for redundancy, each train may be equipped with two backup radios, one installed at the front and one at the rear of the train. (If the backup overlay is built over an LTE network, a single backup radio may be sufficient.) Before transmitting on the overlay network, a VOBC of a train creates a peer node for each backup radio. We assume that the central control station has an attachment to the public Internet or a private IP network, so that the application server can join backup overlays. Fig. 6 presents an illustration of an infrastructure network with a separate backup overlay for each train. With one overlay network per train, the number of peer nodes is either two or three. Since the central control station must participate in every backup overlay, the number of peer nodes at the central control station grows large. However, this does not cause issues even with hundreds of trains, since the processing overhead for each peer node is small.

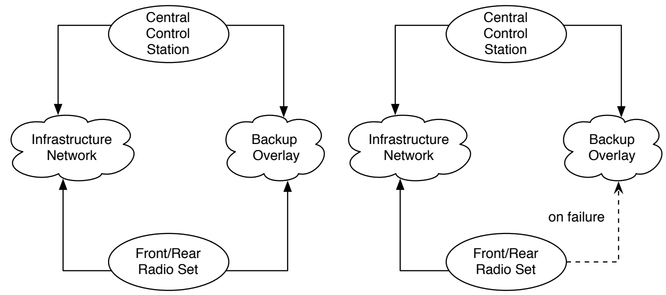
B. Complete Redundancy vs. Join-on-Failure

We discuss two approaches to take advantage of backup overlay networks in a CBTC system with ASONs as shown in Fig. 6, i.e., with one backup overlay for each train. In the first approach, referred to as *complete redundancy*, the central control station transmits every train control message to a train through both the infrastructure network as well as the backup overlay for that train at all times. In the second approach, referred to as *join-on-failure*, the train joins and uses the backup overlay network only in case of a communication failure in the infrastructure network.

For simplicity, we assume that each train has only a single backup radio. We emphasize that a backup overlay does not guarantee the delivery of messages in case of a failure or disruption in the infrastructure network. If a train is not in an area with cellular coverage, it obviously cannot join a backup overlay where a cellular network is the substrate. Likewise, an ad-hoc link to another train can only be successful if that train is within range of the ad-hoc radio.

In the complete redundancy approach illustrated in Fig. 7(a), both the application server at the central control station and the VOBC of a train always connect to the backup overlay for that train. All messages between the server and the VOBC that are sent across the infrastructure are replicated on the backup overlay. If communication across the infrastructure is disrupted, messages sent through the backup overlay are still delivered. Since both trains and servers have joined the backup overlay, there is no need to perform a switchover to the backup overlay.

In the join-on-failure mode shown in Fig. 7(b), the VOBC of a train does not join the backup overlay network during normal operation, and control messages are sent only across the infrastructure network. The central control station always joins the backup overlay and passes a replica of each control message to the local peer node of the backup overlay. Since the VOBC has not joined the overlay network, the peer node at the central control station is the only peer node in the backup overlay. Hence, since there is no neighbor in the



(a) Complete redundancy. (b) Join-on-failure.

Fig. 7: Switchover approaches.

overlay topology, the lookup of the forwarding engine to the overlay topology shown in Fig. 4 will not return a next hop, and the replica of the control message is dropped. Therefore, during normal operation, no messages are transmitted in the substrate network of the backup overlay. If a VOBC does not receive polling messages during several consecutive polling cycles, it triggers a switchover to the backup overlay. Then, the VOBC creates and configures a peer node immediately joins the backup overlay network. Since the (already existing) peer node at the application server is the only other peer node in the backup overlay of the train, the peers at the VOBC and the application server become neighbors in the overlay topology. Once they have established that they are neighbors (following the exchange of messages shown in Fig. 5), the replicate control messages from the control station will be sent across the substrate network to the VOBC as overlay application messages. If the disruption of the infrastructure network is repaired, control messages sent across the infrastructure network begin to arrive at the VOBC. At that time, the VOBC leaves the backup overlay and resumes normal operation. Fig. 7(b) provides a sketch of the switchover in the join-on-failure approach. We emphasize that all actions for the join-on-failure switchover are performed at the VOBC without the need to maintain state information at the central control station.

C. Tradeoffs

The two approaches for engaging the backup overlay, complete redundancy and join-on-failure, present a tradeoff. With complete redundancy, both VOBC and application server are always attached to the overlay network, and there is no switchover delay when the infrastructure fails. On the other hand, the approach incurs the overhead of transmitting and processing redundant replicate messages, in addition to the overhead for maintaining the backup overlays at all times. With join-on-failure, the replicate messages issued by the application server for the backup overlay are transmitted across a substrate network only after the peer node at the VOBC has triggered the need for a switchover. The main concern with join-on-failure is the latency incurred between the time instant when the VOBC triggers the need to a switchover until the time instant when the first control message from the application server arrives at the VOBC. This latency, henceforth

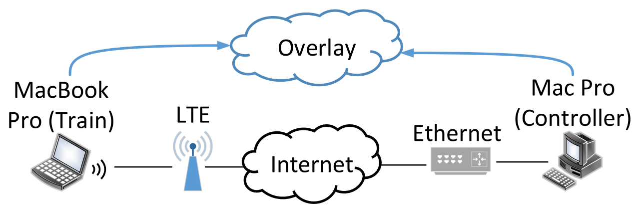


Fig. 8: Experimental setup.

referred to as *switchover delay*, is due to the steps shown in Fig. 5. The switchover delay consists of the elapsed time until the peer nodes at the VOBC and the application server have established each other as neighbors in the backup overlay, and the additional delay that is caused by the transmission of an application message. Since missed polling cycles play a role in detecting infrastructure failures or disruptions, a switchover delay that is less than the length of a polling cycle will not miss more than one polling message due to a switchover.

V. Measurement Experiments

In this section we present measurement experiments that seek to evaluate the viability of ASONs over an LTE network in support of a hybrid CBTC system. We want to evaluate (1) whether an ASON over an LTE network is capable of satisfying the time constraint of a CBTC system, and (2) whether an LTE network can support the communication overhead incurred by a backup overlay. To the best of our knowledge, no measurement experiments exist of the overhead and latencies of running ASONs over an LTE network.

The experiments involve two systems: a MacBook Pro equipped with a 2.7 GHz Intel Core i7 with 8 GB DDR3 RAM and a Mac Pro with two 2.8 GHz Quad-Core Intel Xeon with 4 GB DDR2 RAM. On one of its USB ports, the MacBook Pro has a cellular adapter, a 4G LTE Novatel Wireless U679 Turbo Stick from Bell Canada. The Mac Pro is connected to the campus network of the University of Toronto using 1 Gbps Ethernet. The experimental setup is shown in Fig. 8.

In the experiments, which are all conducted indoors, the Mac Pro represents the central control station and the MacBook Pro plays the role of a VOBC on a train. In the following, we refer to the Mac Pro as the *controller* and to the MacBook Pro as the *train*. The controller has a public IP address, and the LTE adapter on the train has a dynamically assigned private IP address. The campus network and the Internet represent the network between the LTE service provider and the central control station. Since we are only interested in the switchover to the backup overlay, the CBTC infrastructure network is not represented.

ASONs are constructed with the open source software system *HyperCast* [9, 10], which offers a superset of the functionality discussed in Sec. III. As discussed in Subsec. IV-A, the ASONs in the experiments have two peer nodes, one at the controller and the other at the train. The ASONs are configured to send protocol and application messages via unicast UDP.

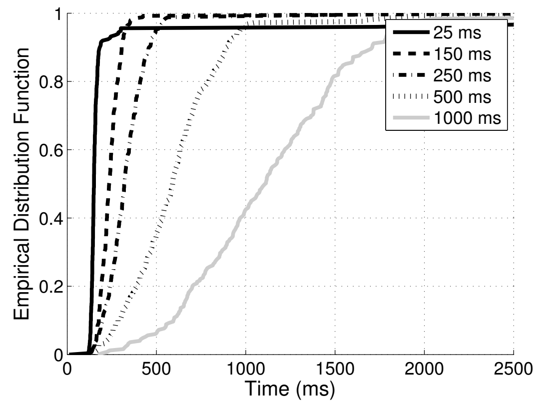


Fig. 9: Switchover delays of ASON over LTE.

The overlay topology uses the DT protocol [8] with a buddy discovery method, where the public IP address and a port number of the controller is provided as the address of the buddy. (Since the ASON has only two nodes, the type of overlay topology is not relevant, as long as a network topology that consists of a single logical link is supported.)

A. Measurement of Switchover Delays

We first evaluate the switchover delay of the join-on-failure approach, as defined in Subsec. IV-B. We measure the earliest time that a peer node can receive an application message after initiating the process of joining an overlay network. In the experiment, the controller has set up an overlay by creating a peer node. An application at the controller sends one application message every 10 ms using the API of the peer node. (Recall that, as long as a peer node has no neighbors, the message is dropped at the local peer node and not sent out). Then the train creates a peer node that joins the overlay, which results in establishing a logical link to the peer node at the controller. Once the link is established, the train can receive the application messages from the controller. We measure the elapsed time at the train between the call to join the overlay until the first application message from the controller is received. We collect measurements of the delay for different values of the heartbeat time of the overlay protocol. Recall that the heartbeat time controls the convergence speed of the overlay, see Sec. III. Each experiment is repeated 250 times.

Fig. 9 depicts the empirical distribution of the switchover delay where the heartbeat time is set to 25, 150, 250, and 1000 ms. Table I shows the median, 95^{th} percentile, and the maximum values. We first observe that the switchover delays increase with larger heartbeat times. This is expected since the convergence time of the overlay topology depends on the heartbeat time. For heartbeats times between 250 and 1000 ms, the 95^{th} percentiles are close to twice the heartbeat time, whereas for a heartbeat time of 25 ms and 150 ms, the 95^{th} percentile does not exhibit a strong correlation to the heartbeat time. This indicates that for a heartbeat time of 150 ms or less, the delays incurred by the LTE network dominate the switchover delays. Thus, by setting the heartbeat

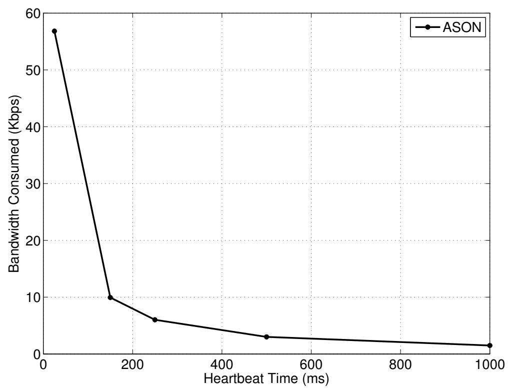


Fig. 10: Bandwidth requirements of protocol messages.

time to 150 ms or less, the overlay protocol does not have a significant impact on the delay performance. It is worth pointing out that the maximum delays listed in Table I, with values exceeding 10 s, by far exceed the delay constraints of a CBTC system. The maximum delays are clearly not correlated with the heartbeat time parameter. Since delays of this order are unlikely to incur in the Internet, the data points to occasional excessive delays occurring in the LTE network.

TABLE I: Statistics of the switchover delays.

Heartbeat Time	Median	95^{th} Percentile	Maximum
25 ms	150 ms	300 ms	13520 ms
150 ms	235 ms	324 ms	3118 ms
250 ms	320 ms	496 ms	1797 ms
500 ms	590 ms	969 ms	2760 ms
1000 ms	1109 ms	1831 ms	5819 ms

B. Overhead of the overlay protocol

We next evaluate the overhead of the overlay protocol. We consider an ASON with two peer nodes. We conduct experiments, where two peer nodes join an ASON at about the same time, remain in the ASON for 20 seconds, and then leave the overlay. No application messages are sent in this experiment. (The bandwidth requirements for application messages depend on the CBTC application. Typically, a train generates data traffic at a rate of 70 kbps, with 50 kbps for the downlink and 20 kbps for the uplink [5].) The protocol messages largely consist of Hello messages that peer nodes periodically sent to their neighbors in the topology. Fig. 10 shows the bandwidth consumption for different values of the heartbeat time. As expected, the bandwidth consumption is determined by the value of the heartbeat time. By setting the heartbeat time to a value of 150 ms or more, the average consumption is less than 10 kbps, which can be considered negligible in the context of the available capacity in an LTE network.

VI. CONCLUSIONS

We have studied a solution to improve the availability of communication paths in a CBTC system using ASON technology, where ASONs create backup networks that use either ad-hoc or an LTE network. We investigated whether ASONs can support the delay constraints of safety-critical communication in a train control system. We conducted measurement experiments that evaluate the latencies when switching to an ASON backup overlay running over LTE. We found that, using suitable protocol parameters, an ASON-over-LTE solution can satisfy sub-second switchover delays in most cases. The occasionally observed excessive delays in the LTE network require additional study, as they jeopardize the use of LTE networks as a backup solution for safety-critical networks.

ACKNOWLEDGEMENTS

This work has been supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and by Thales Canada Transportation Solutions.

REFERENCES

[1] IEEE 1474.1-2004. Standard for Communications-Based Train Control (CBTC) performance and functional requirements, 2004.
[2] B. Ai et al. Future railway services-oriented mobile communications network. *IEEE Commun. Mag.*, 53(10):78–85, 2015.
[3] A. Capone, J. Elias, and F. Martignon. Models and algorithms for the design of service overlay networks. *IEEE Trans. Netw. Service Manag.*, 5(3):143–156, 2008.
[4] M. Y. Chan, S. Baroudi, J. Siu, and J. Liebeherr. Measurement-based handover method for communication-based train control systems. In *Proc. IEEE 86th Vehicular Technology Conference (VTC-Fall)*, Sept 2017.
[5] J. Kim et. al. Automatic train control over LTE: design and performance evaluation. *IEEE Commun. Mag.*, 53(10):102–109, 2015.
[6] J. Farooq and J. Soler. Radio communication for communications-based train control (CBTC): A tutorial and survey. *Commun. Surveys Tuts.*, 19(3):1377–1402, 2017.
[7] H. Jiang, H. Zhao, and B. Zhao. A novel handover scheme in wireless LAN in CBTC system. In *Proc. IEEE International Conference on Service Operations, Logistics, and Informatics*, pages 473–477, July 2011.
[8] J. Liebeherr, M. Nahas, and W. Si. Application-layer multicasting with delaunay triangulation overlays. *IEEE J. Sel. Areas Commun.*, 20(8):1472–1488, October 2002.
[9] J. Liebeherr and M. Valipour. Dissemination of address bindings in multi-substrate overlay networks. In *Proc. International Teletraffic Congress*, pages 270–277, 2011.
[10] J. Liebeherr, J. Wang, and G. Zhang. Programming overlay networks with overlay sockets. In *Proc. 5th COST 264 NGC Workshop, LNCS 2816*, pages 242–253, September 2003.
[11] J. Moreno, J. M. Riera, L. De Haro, and C. Rodriguez. A survey on future railway radio communications services: challenges and opportunities. *IEEE Commun. Mag.*, 53(10):62–68, 2015.
[12] R. D. Pascoe and T. N. Eichorn. What is communication-based train control? *IEEE Veh. Technol. Mag.*, 4(4):16–21, 2009.
[13] E. Stevens-Navarro and V. Wong. Comparison between vertical handoff decision algorithms for heterogeneous wireless networks. In *Proc. IEEE 63rd Spring VTC*, pages 947–951, May 2006.
[14] L. Zhu, , F. R. Yu, B. Ning, and T. Tang. Communication-based train control (CBTC) systems with cooperative relaying: Design and performance analysis. *IEEE Trans. Veh. Technol.*, 63(5):2162–2172, 2014.