
This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Varis, Nuutti; Manner, Jukka

Evaluation of link layer mobility in Ethernet networks

Published in:
Computer Communications

DOI:
[10.1016/j.comcom.2016.07.005](https://doi.org/10.1016/j.comcom.2016.07.005)
[10.1016/j.comcom.2016.07.005](https://doi.org/10.1016/j.comcom.2016.07.005)

Published: 01/05/2017

Document Version
Publisher's PDF, also known as Version of record

Please cite the original version:
Varis, N., & Manner, J. (2017). Evaluation of link layer mobility in Ethernet networks. *Computer Communications*, 103, 193-209. DOI: [10.1016/j.comcom.2016.07.005](https://doi.org/10.1016/j.comcom.2016.07.005), [10.1016/j.comcom.2016.07.005](https://doi.org/10.1016/j.comcom.2016.07.005)

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.



Evaluation of link layer mobility in Ethernet networks



Nuutti Varis*, Jukka Manner

Aalto University School of Electrical Engineering, Department of Communications and Networking, P.O. Box 13000, 00076 Aalto, Finland

ARTICLE INFO

Article history:

Received 3 March 2015

Revised 6 June 2016

Accepted 9 July 2016

Available online 11 July 2016

Keywords:

Ethernet

Distributed Hash Table

Mobility

Experimental evaluation

System design

ABSTRACT

Ethernet is the prevalent link layer mechanism in data communication networks. Today, it is used in enterprise and home networks, data center networks, telecommunication networks, and in various industrial deployments. In addition, more and more hosts are becoming mobile, connecting to Ethernet networks via Wi-Fi, or through virtualization solutions in data centers. Unfortunately, the Ethernet protocol suite itself has no generic support for host mobility.

In this paper, we evaluate the effects of host mobility in Ethernet networks using real-time emulation. We compare routing bridges, an IETF-driven Ethernet frame forwarding protocol, with our DBridges design. DBridges is an evolution of the routing bridges standard, integrating a one-hop Distributed Hash Tables (DHT) scheme into the protocol. Our solution offers improved scalability characteristics in Ethernet networks, as well as enhanced support for host mobility.

Our evaluation shows that while host mobility without any explicit signaling will remain a best-effort service in Ethernet networks, we can significantly improve its efficiency and reliability in certain use cases, while providing improved scalability and safety properties. We also show that while the host mobility support in DBridges suffers from transient forwarding loops, the problem is exceedingly rare in real networks and is mitigated by the base functionality in routing bridges.

© 2016 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY license. (<http://creativecommons.org/licenses/by/4.0/>)

1. Introduction

Ethernet has found its way into most network deployments in the world. It is the de facto link layer protocol for large-scale data centers, in home networks to connect various devices to the Internet through a router, and increasingly in access and aggregation segments of telecommunication networks. Arguably, the most prominent feature of Ethernet that has driven its adoption is simplicity. Each Ethernet device has a unique address that is pre-configured by the manufacturer of the device. This allows the basic Ethernet network to function without configuration; each device can receive messages via the unique Ethernet address, and host locations are discovered from frames arriving on an interface (e.g., a switch port).

Data center networking has emerged as a significant driver for wired Ethernet development in the recent years. Inside the data center, virtualization is commonly used to provide flexible provision of services and applications to customers. Among other benefits, virtualization as a mechanism allows data center operators to

dynamically adjust the resource allocation inside the data center, leading to potential energy savings and resource efficiency [1,2]. A large part of the flexibility of virtualization stems from the possibility to dynamically migrate virtual machines to different parts (e.g., racks) inside the data center. From the perspective of the network, the *live migration* [3] of virtual machines often presents itself as a host mobility event.

Conventional 802.3-based Ethernet does not mandate any explicit signaling when a host changes its location; i.e., it allows hosts to be silent. Without an explicit handover procedure, the network will not be able to deliver frames to the mobile host until it has been active in the new location. While standard Ethernet does not have any explicit signaling when a host attaches or detaches from a network, supplemental protocols, such as 802.1X [4] or in many cases the 802.11 wireless Ethernet protocol family can be used to monitor the connection state of Ethernet devices. In the case of virtualization, the live migration process typically ends with an Address Resolution Protocol (ARP) announcement sent to the network to update the location information (i.e., MAC learning tables) in the Ethernet switches.

This paper evaluates and compares two different methods for supporting host mobility in Ethernet networks directly on the link layer. Our efforts concentrate on evaluating link-layer behavior that

* Corresponding author.

E-mail addresses: nuutti.varis@aalto.fi (N. Varis), jukka.manner@aalto.fi (J. Manner).

in the Ethernet family of protocols as a whole, scoping out the mobility management protocols typically in use with wireless devices (e.g., MobileIP [5,6], Inter-Access Point Protocol (IAPP) [7], and many IEEE ratified wireless standards [8,9]), or protocols designed for media-independent handovers [10]. Consequently, host mobility support in wired Ethernet networks can be described on a high level as a method for updating the location of the mobile host (e.g., host MAC address) on switches in the network. To emulate host link-layer behavior consistent with real-world scenarios, we use data collected from wireless devices to implement a mobility model for the evaluation.

Conventional Ethernet switching uses passive location information updates to support host mobility in the network, i.e., each Ethernet frame emitted in the network updates the location of the host in switches without any explicit signaling. The passive updating scheme is directly integrated into the base protocol operation of all Ethernet switches and offers flexible updates of location information in the network. However, as pointed out in the literature [11–14] and practical use cases, conventional Ethernet networks have significant scaling issues that limit the number of active hosts and the overall throughput of the network.

Active location information update mechanisms have been proposed in the past by various research papers and industry efforts. Active location updates, signaled through an external mechanism, are generally used to either eliminate or minimize the flooding behavior in Ethernet networks used by frame forwarding and higher layer address resolution (e.g., ARP). The active location information update mechanism can also be used to support mobility in the network by signaling location updates to switches in the network instead of learning the location of hosts passively from forwarded frames.

Our previous work [15] introduced DBridges, a design that merges the one-hop Distributed Hash Table (DHT) scheme from SEATTLE [16] with the IETF routing bridges standard [17], and evaluated our design with static hosts. In this paper, we identify problem areas in our original design when the network contains a significant amount of host mobility. We also present an evolution of the original DBridges design that significantly improves the behavior of the system in networks with high degree of host mobility. Finally, we also evaluate our new system design in relation to the signaling characteristics of host mobility on the Ethernet link-layer by analyzing network behavior from several perspectives:

- The continuous signaling load in the network during mobility events,
- the effect and overhead of host mobility on the signaling,
- the effect of host mobility on overall path length of frames,
- the length of convergence periods in the network, and
- the connection interruptions on the host application layer.

The evaluation is performed on a network topology that resembles a wireless access network. Our evaluation results show that in many cases, DBridges significantly improve the efficiency and reliability of the signaling in Ethernet networks with mobile hosts. Furthermore, we show that in worst case situations, our solution performs similarly with conventional Ethernet networking while simultaneously reducing the signaling traffic in the network. We also describe a forwarding loop problem in the SEATTLE DHT scheme and show how the problem is exceedingly rare in real networks and mitigated by our choice of routing bridges as the base system.

The rest of the paper is structured as follows. Section 2 covers some of the work done on Ethernet mobility and gives a brief overview of the various designs that support Ethernet mobility. Section 3 covers the functionality of Ethernet networks in relation to host mobility and introduces the passive and active location information update mechanisms in more detail. Next, Section 4 describes routing bridges, the Ethernet frame forwarding protocol

used as a foundation for our design. We will also discuss the functionality and issues with host mobility in terms of routing bridges. Section 5 introduces host mobility support in DBridges and gives a high-level overview of the DHT scheme. We also discuss some of the inherent problems with the active location information update mechanism. Sections 6 and 7 include our evaluation, starting with an overview of our evaluation setup and environment, followed by a discussion of our results. Finally, Section 8 concludes the paper.

2. Related work

There have been several proposals for solving scalability in Ethernet networks where a part of the problem set is host mobility. Most of the other work includes host mobility as one of the requirements for the system. The primary motivator for the requirement is the prevalent use of virtualization, and as a consequence, the live migration of virtual machines. However, the majority of the work concentrates on evaluating other issues with Ethernet, such as bandwidth, fault tolerance, security, cost-efficiency, and segment size (both in terms of number of active hosts, and number of switches). The following section discusses different domains, where Ethernet link-layer mechanisms are typically involved in the host mobility solution of the network.

In addition, the increase in cloud computing has brought Software Defined Networking (SDN)-based network architectures to the academic forefront, and shifted focus away from a traditional distributed network architectures such as various link-state protocol-based approaches (including DBridges). Recently, Network Function Virtualization (NFV) [18] has gained traction due to its promise of increased elasticity in the network. As a consequence a significant body of recent work has been published on the research problems surrounding NFV. In this context, mobility is often treated as a higher level construct (e.g., migration of virtual network functions) than what is discussed in this paper (e.g., the link-layer signaling effects of host mobility events).

In contrast, we concentrate on evaluating the effects of host mobility on our solution and compare it to a conventional Ethernet solution (i.e., routing bridges). To our knowledge, our evaluation represents one of the first in-depth studies of the network effect of mobility in Ethernet networks. Consequently, our related work goes over proposals that deal with host mobility in Ethernet networks, however relatively few actually evaluate the effect of mobility. The majority of the mobility management research that considers experimental evaluation is done in the domain of wireless Ethernet, and as such is not directly comparable to our work.

Using Ethernet as part of the mobility solution in the context of telecommunications networks has been proposed in the past. Mobile Ethernet [19] offers either a centralized or a fully distributed host location information storage in a hierarchical ring-like topology. The hierarchical constraint is used to reduce the amount of conventional MAC learning done in the network and to segment the network around a ring-like core network. Mobile Ethernet evaluates the overhead of two different frame forwarding schemes in terms of generated traffic concentrated on the core network switches. However no evaluation of other scaling characteristics is done.

Recently, there has been significant academic activity surrounding the forthcoming 5G family of cellular data communication standards. One of the key research topics in 5G standardization is to improve the elasticity and efficiency of the telecommunications network architecture. The presented works typically offer benefits through NFV-based solutions, or by simplifying the user data and control planes. Ameigeiras et al. [20] propose a two-layer approach to solve the mobility-related inefficiencies in the evolved packet

core. They use link-layer mobility (through encapsulation or frame rewriting on the edges of the network) on the access network, and switch to Mobile IP-based solutions on the regional level. The management is driven by a logically centralized, SDN-based controller. Banik et al. [21] expand on our prior work [22] by replacing our DBridges-based proposal for mobility management with an SDN-orchestrated system using MPLS encapsulation.

Mobility management for fast-moving vehicles [23] in Spanning Tree Protocol (STP) based Ethernet networks has also been evaluated in the past. The paper proposes to build VLAN-based tunnels in the network to perform a seamless handover between access routers on the edges. Two separate schemes are proposed. First, management-driven tunnel creation, and second, a signaling-based solution where the mobile access points initiate tunnel creation upon attachment. The evaluation concentrates on a hybrid solution where the management-based solution is used for the “top half” of the STP network and the signaling-based solution is used in the lower half. The paper evaluates tunnel setup times for different-sized topologies and packet losses in the network with different numbers of access (edge) switches.

There are several proposed solutions that use a form of distributed storage to offer host mobility support in the network. The SEATTLE project [16] uses one-hop DHT to eliminate flooding from the network and to support host location changes efficiently in the network. The solution is distributed and there is no logically centralized components, which also is our main inspiration on DBridges. Next, SPAIN [24] uses “chirping” to disseminate host location information in the network when a host changes location. Chirping, in combination with host location changes, broadcasts the new location of the host throughout the network, creating an immediate up-to-date state in the whole network at the cost of increased bandwidth. Finally, Smartbridges [25] use a form of diffusing computations when a host changes location to update the whole network with the new host location. All of the forementioned solutions in the worst case store all host location information on all switches in the network. However, SEATTLE only requires the information in an all-to-all communication pattern, while the other two solutions spread the information as a part of the location updates.

There are also a number of centralized solutions proposed to improve host mobility support in Ethernet networks. Both SANE [26] and its successor, Ethane [27], offer strict security-driven network architecture for Ethernets that also offers host mobility support. VL2 [28] implements an overlay network on top of IP that assigns location-independent addresses to hosts. As a part of the system, it also provides a centralized hierarchical information storage that is used to resolve location-independent addresses to the underlying IP network addresses. This allows VL2 to support host mobility, as the location independent addressing allows hosts to retain their IP address while they move in the network. PAST [13] uses a logically centralized controller (e.g., OpenFlow [29]) to orchestrate efficient routing of packets in the network, based on the destination Ethernet address, and a VLAN tag. Host mobility in PAST networks triggers a recomputation of the forwarding tree for that destination address, VLAN tuple that is then installed into the network, and the old path is torn down. Fang et al. [30] propose a hierarchical label-based forwarding architecture, where seamless host mobility is supported through signaling mechanisms similar to DBridges (e.g., inform old location of the new attachment point of a mobile host). Shadow MACs [31] uses per-flow MAC address rewriting on the edges of the network to provide efficient frame forwarding directly on the Ethernet link-layer. The MAC address rewriting can also be used to provide seamless host mobility that achieves per-packet consistency, however the processing overhead may be prohibitively high if the mobile hosts exhibit a large

number of concurrent flows or their mobility behavior is highly dynamic.

3. Host mobility in Ethernet

Host mobility support in Ethernet networks is incorporated into the base protocol operation, i.e., MAC learning tables and port-based forwarding of frames. In this section, we go over some of the details of Ethernet networks relating to host mobility, describe some of its challenges, and present the high-level details of an active location information update scheme.

3.1. Ethernet frame forwarding

In the Ethernet network, switches are responsible for forwarding Ethernet frames throughout the network. Each switch contains a lookup table that records the source Ethernet address and switch port number from each incoming Ethernet frame. Frame forwarding is based on this information; if the Ethernet address of a previously recorded entry matches the destination address in the frame, it is forwarded to the port indicated in the lookup table. If there is no entry for the address, the frame is broadcast from all ports in the switch, except the incoming port. This latter provision guarantees that even silent hosts can receive frames in an Ethernet network, since the flooding will deliver the frame to every link. It is also the basis for the “plug and play” behavior of Ethernet devices.

3.2. Host mobility support challenges

The basic frame forwarding method in Ethernet switches is poorly suited for networks where the hosts are mobile. There are two reasons for this. First, after a host has attached to another switch in the network, the host must emit a frame to update the MAC learning tables in the network. Until the host emits something on the link, all Ethernet frames destined to the mobile host are still forwarded to the old switch, where they are blackholed. This issue is partly mitigated by upper layer protocols (e.g., DHCP) that are activated when the link on the host comes up, or by an explicit operating system configuration that emits an ARP announcement when it detects that the link has become active. Unless external protocols or host modifications are used, this issue cannot be fully resolved by network design changes.

The second issue is that the location information of the mobile host is *passively updated* by emitted Ethernet frames rather than the mobility event. Consequently, the location updates are fully dependent on the frames traversing the network, as the frame types dictate the set of updated switches. Unicast frames will update the switches on the path taken by the frame. Multicast and broadcast frames typically update a large subset (or all switches in the network) as they are delivered to multiple recipients. In the former case, only a subset of the switches have an up-to-date information for the host and can correctly forward frames from attached hosts to the mobile host. In the latter case, the whole network is updated as the frame is flooded throughout the network, and all hosts can continue communicating with the mobile host normally, but at the cost of increased traffic and processing overhead.

In the end, host mobility in Ethernet networks that have no explicit signaling for mobile hosts is a best-effort service. Even in perfect conditions, as there is no handover process for hosts, frames that are currently destined towards the mobile host, but arrive before the host has emitted a frame that updates the paths in the network, are blackholed at the old switch. We can, however, examine and evaluate the mechanisms by which different Ethernet frame forwarding protocols cope with host mobility in the network.

3.3. Active location information updates

In contrast, location information in switches can also be updated actively. More generally, in an active system there are typically two sources that can update location information. First, location information is explicitly signaled to switches that contain stale information. Stale information is automatically detected by other switches in the network if they receive frames for a host that is no longer attached to one of their ports. Secondly, passive MAC learning can also be used to learn location information from frames traversing the network. The active signaling method allows the host service to continue regardless of the frame types the mobile host emits because switches are responsible for actively propagating the current location of the host to switches that contain stale information.

One of the direct problems with active location updates is the amount of overhead it generates. Active location updates require a separate signaling scheme on top of the Ethernet network, which adds traffic to the network and processing overhead on the switches. The signaling overhead is typically heavily mitigated by the use of MAC learning tables where location information can be cached after it has been learned through the active signaling. A more significant issue in certain use cases may be the change in how up-to-date location information is propagated in networks. Active location updates shift the responsibility of timely location updates to the external signaling system.

4. Routing bridges

To solve some of the inherent issues in conventional Ethernet switches, IETF set out to design a new Ethernet frame forwarding protocol. The result of the effort is a new protocol called Transparent Interconnection of Lots of Links (TRILL), and a new type of Ethernet switching device, called a Routing Bridge (RBridge). RBridges are fully autoconfigurable and retain the Ethernet plug-and-play nature of hosts. They are also incrementally deployable with STP-based Ethernet networks.

In contrast to STP-based solutions, RBridges forward Ethernet frames using Shortest Path Forwarding (SPF) which is computed using a modified version of the IS-IS link state protocol [32]. SPF-based forwarding improves the use of links in the network, and in many deployments offers shorter average paths in the network. RBridges use a small additional header, added on the first hop and removed on the last hop of the network, to convey forwarding information in the frame as it traverses the network. Additionally, the header also contains a hop count field to protect the Ethernet network from broadcast storms.

Routing bridges primarily use three different sources of information to learn the host locations in the network: (1) Native Ethernet frames, i.e., locally attached hosts, (2) TRILL encapsulated Ethernet frames, i.e., hosts behind remote routing bridges, and (3) the End-Station Address Distribution Information (ESADI) protocol. Two mechanisms are inherent to the Ethernet network, and function identically with conventional STP-like switches. The third source introduces an optional active signaling mechanism to disseminate host location information in the network.

Routing bridges associate a “confidence level” for each source of information. The MAC learning process, and by extension the location information management only occurs, when the confidence level of the learned source is equal or higher than the confidence level of the current information. By default native Ethernet and TRILL encapsulated frames use an identical confidence level, while ESADI-based information is given a significantly higher value.

RBridges preserve the same functionality relating to host mobility as conventional Ethernet frame forwarding protocols. RBridges employ a similar passive location update method for end hosts.

However, instead of each switch recording the location information in the MAC learning tables, only the edges of the network (i.e., the switches involved in encapsulating and decapsulating frames) must do it. In addition, flooding is still used as the mechanism to deliver multi- or unknown destination Ethernet frames in the network. In summary, even if RBridges use a completely different forwarding method inside the Ethernet network, the same benefits and issues that apply to more conventional frame forwarding protocols apply to routing bridges.

4.1. End Station Address Distribution Information Protocol

RBridges have a separate optional protocol to disseminate host location information in the network. The ESADI protocol [33] uses multi-destination signaling messages to deliver host location information on a virtual LAN basis to other routing bridges in the Ethernet network.

ESADI can be seen as an active way to propagate host location information in the network. However, there is a subtle difference from the earlier discussion of active location updates. ESADI uses a *proactive* method to update the location information in switches, regardless of whether it is stale or not. Updating information proactively simplifies the protocol design. However, it also introduces two problems. First, the protocol may generate a high amount of wasted signaling in the network if the majority of the switches participating in the ESADI instance do not need the propagated information. Secondly, in networks with highly mobile hosts, the proactive method must compromise between increased multicasting and update frequency.

5. DBridges

DHT Routing Bridges (DBridges) extends the base routing bridges specification with a one-hop Distributed Hash Table (DHT) mechanism, introduced by the SEATTLE project. The main goal of the mechanism is to eliminate major sources of flooding from the network. At the same time, the underlying functionality can also be used to implement active location information updates in the network, thus improving host mobility support in Ethernet networks without requiring external signaling protocols or host modifications. To guarantee incremental deployability with routing bridges and situations where the DHT is unusable, the mechanism can be complemented with conventional passive location updates.

5.1. Architectural overview

DBridges build on the foundation of routing bridges to offer active location information updates in the network between DBridges and support passive location updates simultaneously with routing bridges. The link state protocol is used to advertise DHT participation across all switches in the network, which routing bridges ignore and DBridges use to build a map of participants. As with routing bridges, DBridges only need location information updates on switches that have access ports, i.e., ports that have directly attached hosts. The “core” of the network can then consist of either RBridges or DBridges responsible for storing the information in the DHT.

The DHT has three functions in the network, two relating to scalability in Ethernet networks and one relating to host mobility support. First, the host location information eliminates unknown destination flooding from the network by forwarding the frames through the switches that store the location information. Secondly, the DHT is used to store addressing information for hosts. The addressing information is used to implement a proxy ARP-like system, where normally broadcast requests are instead redirected through the DHT to the correct switch for resolution.

Finally, the location information is also used to implement an active location information update mechanism to improve the reliability of host mobility support between DBridges. Furthermore, the DHT functionality allows switches to update location information in a reactive way, i.e., only when it is necessary. This reduces the signaling overhead that occurs with a more proactive system (such as ESADI) or flooding with passive location updates.

5.2. Distributed Hash Tables

Distributed Hash Tables (DHT) are the most well-known application of Consistent Hashing [34]. Consistent Hashing was first introduced as an efficient method to load balance resource usage in a distributed system. Since then, the design has been adopted in a wide variety of systems, ranging from peer-to-peer applications to distributed storage systems.

DBridges implement a system where switches in the network act as nodes in the DHT. As mentioned earlier, all switches receive participation information through the link state protocol that routing bridges implement. Thus, when the network is in a converged state, every switch has a complete view of the DHT. This is in contrast to the typical algorithm use [35], where each node sees only a subset of the complete view and uses the DHT both as an overlay network to deliver information and as a mechanism to select the node where information is stored. Currently, DBridges store two types of information: (1) Location information, containing the MAC address of the host, and the RBridge nickname of the switch it is attached to, and (2) layer 3 addressing information, containing the IPv4 address of the host, and the MAC address of the host.

This specialization of the algorithm is called a one-hop DHT [36]. The complete view of the switches in a key space allows participants to directly (i.e., by using “one-hop”) index the correct switch for a host MAC or IP address. Typically, one-hop DHT systems have lower lookup latency and lookup failure rates than multi-hop systems, while the lookup table size grows linearly with the number of nodes in the system. The scalability requirements are not an issue for DBridges, as there are always a practically finite number of switches in the network. A more extensive description of the basic one-hop DHT functionality and its use to eliminate flooding in the Ethernet network can be found in our previous work [15].

5.3. Location information update mechanism

The confidence level-based MAC learning table update mechanism is insufficient in Ethernet networks with host mobility. Since ESADI information in default configuration is more trustworthy than information derived from incoming native Ethernet frames, switches will only update the information when it is missing from MAC learning tables. In addition, ESADI in practice only periodically updates the location information, resulting in extended periods of service disruption for mobile hosts. DBridges treat all three location information sources to be of equal confidence. This allows DBridges to support host mobility in the network, as all three sources can update existing entries in the MAC learning table.

Our design replaces ESADI with a one-hop DHT mechanism that updates location information in the network using unicast signaling. We use “DHT signaling” in subsequent discussion to refer to the unicast frames. However, since the update mechanism is reactive instead of proactive, switches only get updated when they contain stale location information for a host in their MAC learning table. To avoid excessive host service disruption, our reactive scheme requires that switches use *redirection* to temporarily forward frames destined to the mobile host through intermediary switches.

The use of redirection, in combination with the SPF-based forwarding algorithm introduces a consistency problem in the MAC learning table of switches. When information from three different sources can rearrange itself in the network (e.g., due to frame re-ordering or processing delays), it is possible that a switch ends up in a state where the location information for a given host is invalid. The effect of the invalid information varies in severity, depending on the interplay of the order of received frames, and the traffic and mobility patterns of the hosts. Typically, two different outcomes can be observed. More commonly, the redirection temporarily forwards frames between two hosts using non-optimal paths, or in rare cases it may create a transient forwarding loop.

The effect of the first outcome is a temporarily increased latency for forwarded frames between the source and destination hosts. DHT signaling automatically repairs the path in seconds. Additionally, when the redirected path contains several intermediate switches or in very large networks, it may lead to service disruptions if the redirected path exceeds the maximum hop count. The second case is more severe, causing a localized forwarding loop on the SPF path between the source and destination DBridges. However, the hop count in the frames ensure that they are dropped after a finite number of hops. The loop automatically corrects itself when the location information in one of the switches is updated, either to another invalid state, or the correct state.

5.4. Design changes

To minimize the probability of invalid state in MAC learning tables, we have modified our initial design. First, we have adjusted the MAC learning process in DBridges. DBridges always learn location information from native Ethernet frames and DHT signaling, and conditionally from TRILL encapsulated frames. Location information from TRILL encapsulated frames is only learned if the originating switch was a RBridge (i.e., the Ethernet network uses a mix of DBridges and RBridges). This change directly reduces the probability of invalid state in MAC learning tables, because location information for the same (mobile) host is no longer learned from incoming TRILL encapsulated frames and DHT signaling.

The rationale for dropping TRILL encapsulated frames as a source is based on the observation that a fully deployed DBridge network requires only a native Ethernet frame to trigger the location information management in the one-hop DHT. Subsequently, the network-wide location information management can be achieved using the reactive DHT signaling. In mixed networks, location information relevant to RBridges has to be learned through TRILL encapsulated frames because there is no DHT signaling to learn it from.

Secondly, we also adjust our DHT signaling and redirection logic to avoid a potential source of transient loops. If a DBridge receives a TRILL encapsulated frame originating from a DBridge that is currently the location of the host in the MAC learning table of the receiver, the frame is silently discarded. The rationale here is that in terms of host mobility, one or both switches contain invalid information in the MAC learning table. The receiver chooses the safe option that protects the path between the switches from a transient forwarding loop.

Finally, we have also implemented an optional feature in the signaling logic of DBridges that reduces the number of DHT signaling messages in the network, when the hosts have a high degree of mobility. *Deferred DHT signaling* causes intermediate DBridges (see Section 5.6 for redirection chains) to suppress DHT signaling to the originator switch. Instead, the destination DBridge signals the current location of the host to the originator when it decapsulates and forwards the native Ethernet frame. Deferred signaling is implemented as an optional feature in DBridges as it breaks data plane compatibility with routing bridges. Our evaluation in this paper is

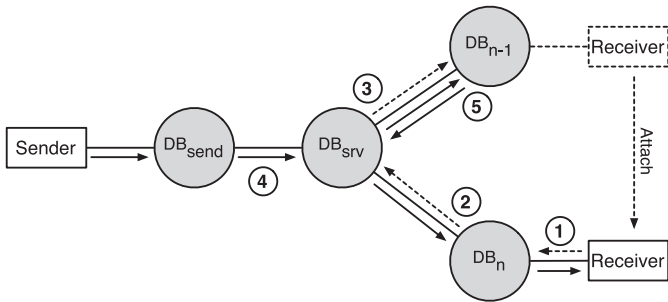


Fig. 1. Host mobility support overview; dashed lines indicate DHT state repair signaling; solid lines indicate host frames.

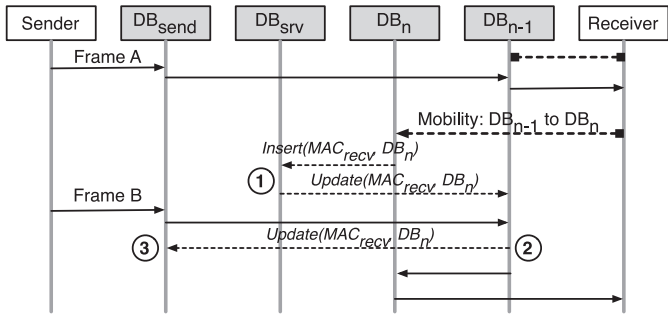


Fig. 2. Active location information updates in the network.

performed with routing bridges compatible DHT signaling mechanism, i.e., with deferred signaling disabled.

5.5. Host mobility support

Fig. 1 presents an example of an active location update mechanism in a network with four switches (DB_{send} , DB_{srv} , DB_n , and DB_{n-1}), and two hosts (sender, and receiver). DB_{srv} is responsible for storing the location information of the receiver. After the receiver attaches itself to DB_n and (1) emits an Ethernet frame on the link (or triggers some other explicit signaling mechanism available), the switch begins the active location update process in the network. Using DHT signaling, it informs DB_{srv} of the location of the receiver (2). It in turn informs the previous attachment point of the receiver (DB_{n-1}) of the location information change (3). After this, frames from the sender to the receiver (4) using stale location information (destined to DB_{n-1}) are (5) redirected to DB_n by DB_{n-1} . This completes the first phase of the active location update mechanism in the network.

Fig. 2 presents the active location update mechanism as a sequence of events and discusses the second phase of the process. A mobility event that attaches the receiver to DB_n begins the active location update mechanism once it sees a signal indicating the attachment. It causes DB_n to use DHT signaling to inform the DHT server of MAC_{recv} that the location has changed. When DB_{srv} receives the signaling message (1), it updates the information and signals the changed information to the previous attachment point of the receiver. The information is signaled as an “update” operation to enable frames destined to the receiver to be relayed to its current location instead of the previous attachment point.

When the previous attachment point (DB_{n-1}) receives frames destined to MAC_{recv} , it must redirect the frames towards its current attachment point. Additionally, it also (2) signals the changed location information of the receiver to the originating switch (DB_{send}). Updating stale location information (3) is the second phase of active location information update mechanism. Once all switches

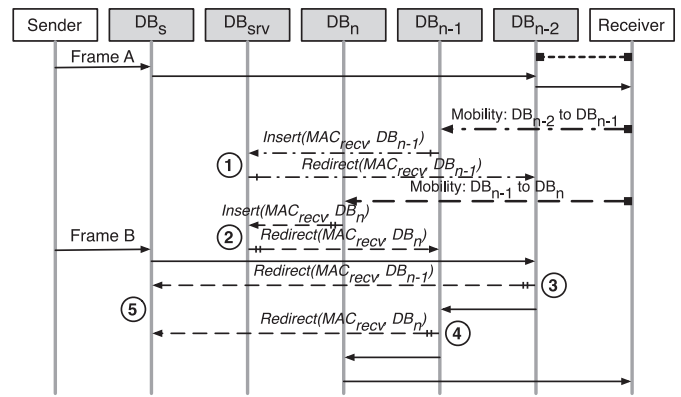


Fig. 3. A redirection chain in DBridge networks; dashed lines indicate DHT-related signaling; solid lines indicate user data frames.

that have attached hosts with active flows to the receiver have up-to-date view of its location, the update process is complete.

5.6. Redirection chains

The redirection of frames towards the mobile host that begins with the DHT server informing the old switch of the new location of the host will only repair the state in the network in a reactive way. In other words, switches that have attached hosts communicating with the mobile host will contain stale location information until it is updated by the DHT signaling mechanism.

When the mobile host moves again, the DHT server again updates the new location of the mobile host to the old switch, thus creating a redirection chain in the network. There are now two switches in the network where the mobile host has been active but has moved from. Each time the host moved, the state repair process updated the location in the switch, but the update is only performed on the originating switch of the mobile host.

Thus, elsewhere in the network, there may be a stale switch that has not seen frames from the mobile host in its new location and considers the host to be still located behind the first switch. The stale switch will forward frames to the first switch, which will then redirect the frames to the second switch which will finally redirect the frames to the current location of the mobile host, creating a redirection chain in the network.

Fig. 3 presents the creation sequence for a redirection chain in the network. Here, the sender has an open flow to the receiver host, and the receiver attaches itself to a number of switches in the network. Initially, the receiver is attached to switch DB_{n-2} . Next, it attaches to switch DB_{n-1} which begins a state repair process for the receiver in the network. First, the new switch (1) signals a new location information entry to the DHT server DB_{srv} , updating the location information of the receiver, which in turn informs the old switch (DB_{n-2}) of the location change by issuing a redirect request. Then, before the sender has sent another frame to the receiver, the receiver moves again, attaching itself to DB_n . DB_n in turn will (2) begin the state repair process in the network for the receiver. As a result, the previous location (DB_{n-1}) is now redirecting frames destined to receiver to DB_n , creating a redirection chain (length of 2) in the network.

After the redirect chain has been created, the sender sends a frame towards the receiver, which is forwarded by the attached switch (DB_s) to DB_{n-2} , because location information is stale in the switch. Switch DB_{n-2} in turn (3) forwards the frame using the redirect information towards DB_{n-1} and additionally sends a location update to DB_s , indicating that receiver is found behind DB_{n-1} . At DB_{n-1} , (4) the same procedure repeats, resulting in another loca-

tion update frame to be sent to DB_s , which informs the switch of the current location of the receiver.

Depending on the network topology and link congestion, DB_s (5) receives two location updates in rapid succession for receiver. If the location updates come in order, DB_s will use the optimal path forwarding for frames destined to receiver. If frame reordering occurs, the path will be non-optimal. However, end-host service is not disrupted, as the frames are forwarded temporarily through the redirect chain. The path is updated later when a new location update frame is sent by the switches involved in the redirection chain to DB_s .

Redirection chains allow reactive location updates in the network to function, regardless of the overall traffic or mobility pattern in the network. As long as the mobile host is detectable at the newly attached switch by some means, the state repair process will eventually fix the network so that all relevant switches know the host's new location. Redirection chains also minimize the number of incorrectly delivered frames without using complicated signaling mechanisms and buffering at switches.

The DHT signaling mechanism introduces a small overhead in the network in the form of unicast frames and frame processing on switches. Redirection chains increase this overhead by the number of redirection switches, as the default behavior is to signal the location update from each intermediate switch. This can also introduce frame reordering and cause the path for the source-destination pair to be non-optimal for longer periods.

5.7. Transient forwarding loops

The creation of a transient forwarding loop requires that the network has in-flight DHT signaling frames that contain location information about the mobile host. As Ethernet network end-to-end latencies are typically in the micro to milliseconds while typical handover durations are at least an order of magnitude higher, there is no practical danger in most network deployments during normal operation.

In our evaluation, we found that the causes for the forwarding loops are always timing-related. When a switch in the network receives location information about a host in temporally incorrect order, there is a possibility that a transient forwarding loop is created in the network, relating to that host. The duration of the forwarding loop is always related to the frequency of outgoing frames from the host, because the next outgoing frame will begin the state repair process in the network.

Typically, temporally incorrect ordering occurs when a host attaches to another switch and emits a frame (which the switch receives) before all in-flight frames related to the location of the host have been received in the network. A frame emitted from the new location will begin the state repair process in the network, which first signals the DHT server storing the location information of the host, and subsequently the old switch of the host that the location has changed. If the new location then receives a DHT signaling frame that indicates that the host is located behind the old switch, a forwarding loop has been created in the network.

6. Evaluation setup

Our evaluation concentrates on examining the characteristics of the network when hundreds of hosts with a high level of mobility are performing communication tasks, either to each other or to an external network. Our network topology is modeled to represent a relatively unstructured wireless access network. We primarily use two different traffic models for hosts to stress the system in different ways. Our primary traffic model approximates a typical mobile network operator's view of host communication, i.e., primarily downstream connections to external networks, with heavy

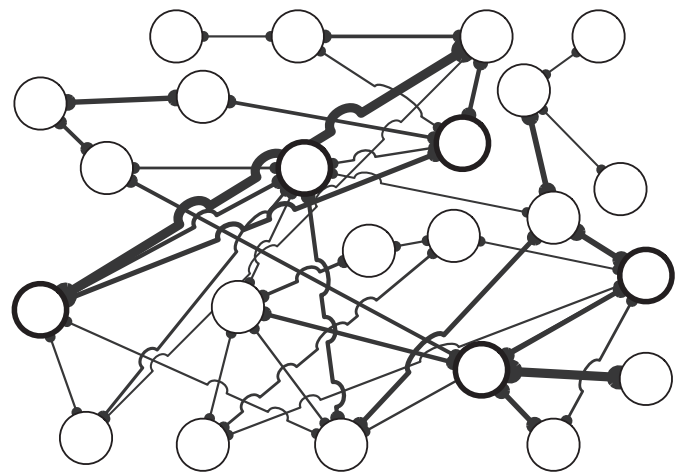


Fig. 4. The EBONE topology; thicker outline nodes act as DHT servers. Link line thickness is in relation to link capacity.

tailed flow lengths and flow initiation frequencies. In addition, we also evaluate the network characteristics with a peer-to-peer traffic model that can be considered a “worst-case scenario” from the point of view of network processing. Finally, to illustrate some of the underlying mechanisms in the evaluation, we use a simple traffic model where each mobile host in the network opens a single outgoing flow to another randomly selected mobile host.

The number of hosts and the traffic models, combined with our mobility model allows us to evaluate various characteristics of our design and compare the results to a network with RBridges. The comparison to a network with RBridges can also be seen as a comparison between a more general “conventional Ethernet network” and our solution. We also intentionally designed our baseline evaluation traffic and mobility models to verify the correctness of our protocol (e.g., high degree of overall mobility in the network, and in the case of peer-to-peer traffic model, highly dynamic flows). Our evaluation concentrates on four different aspects of the network behavior:

- control plane traffic in the network,
- path length of Ethernet frames, normalized to the optimal paths,
- the duration of the convergence time in the network after a mobility event, and
- the packet loss rate on mobile hosts during testing.

6.1. Evaluation topology

Fig. 4 presents the topology used throughout our evaluation. It is a simplified model of the EBONE backbone topology from Rocketfuel [37]. We have simplified the actual topological data by combining separate routers located at the same location to a single node and combined the link weights accordingly.

We use a simplified version of the EBONE topology as our test platform for several reasons. First, we want to validate our claim that the design is fully functional in more generalized topologies than the typical tiered data center model. Secondly, our test topology is sufficiently complex (i.e., 23 switches, and 38 switch-to-switch links) that we can easily observe the DHT-specific behaviors such as path stretch, which could be difficult to see in smaller or more structured topologies. Finally, we wanted a reasonably unstructured topology upon which implement our mobility model. This allows more unpredictable movement of hosts in the network than in tree or ring-like topologies, which are inherently more structured.

Table 1
Evaluation topology characteristics.

Switches	Links	Hosts
23	38	400

Table 2
Evaluation traffic model characteristics. Values given are approximated based on corresponding probability distributions.

Model	Max #flows	Flow duration	Traffic rate
Downstream	$5 \cdot \text{hosts}$	47.8 s	100 ms
Peer-to-peer	$(\text{hosts} - 1)^2$	460 ms	12.5 ms
Synthetic-cvg	hosts	120 s	100 ms
Synthetic-intr	hosts to $5 \cdot \text{hosts}$	120 s	5 s/10 ms

On the switches, the SPF algorithm used by IS-IS builds a 22-link minimum spanning tree for frame forwarding to each other switch in the network. In addition, a single minimum spanning tree is selected as the multicast forwarding tree in the network. As we are not interested in the capacity or the throughput of the network as a whole, we have not modeled specific bandwidth characteristics for the virtual links.

We use the Linux network emulator (netem) [38] to introduce a fixed one-millisecond delay on each switch-to-switch link in the topology. While the delay would be exceedingly high in real-world scenarios for most fixed line Ethernet links, it gives us a lower bound time in the network between the two end points of an Ethernet frame.

Finally, we use a network simulator to emulate 400 hosts in the network, initially distributed equally between all 23 switches in the network. Each host is emulated with a full network stack, including the link layer. The emulated hosts generate real-time Ethernet traffic directly into the network that the switches forward. Table 1 summarizes the basic characteristics of our evaluation topology.

6.2. Traffic model

We use two separate traffic models to analyze various network characteristics when hosts are mobile. In addition, we use additional “synthetic” traffic models to illustrate some characteristics of the network. Table 2 summarizes the characteristics of traffic models we use in our experimental evaluation. In the table, the maximum number of flows refers to the maximum potential communicating pairs, the flow duration describes the mean duration of individual flows, and the traffic rate describes the mean rate of frames emitted in an individual flow. If the traffic rate is not symmetric, the first value indicates the upstream traffic rate, and the latter the downstream traffic rate.

The first traffic model is loosely based on a week-long dataset during January 2012, from a European mobile operator. The dataset is filtered to only contain HTTP flows, which are analyzed for two separate metrics: flow duration and flow frequency on a host basis. We use the (pseudo-anonymized) source IP address of the flow to uniquely identify hosts. The resulting traffic model should roughly estimate the network effect of mobility on user data in a typical cellular access network, where hosts connect to the Internet at large through specific gateways in the network but have little or no direct communication between each other. We refer to this traffic model as a “downstream traffic model” throughout the evaluation.

The second traffic model is based on the work done in our previous paper [15] to observe various characteristics in the network when hosts are not mobile and communicate using a peer-to-peer-like traffic pattern. This traffic model can be construed as

Table 3
Evaluation mobility model.

Model	Distribution	Parameters
Baseline	Lognormal	$\mu = 0.853, \sigma = 1.78$

a worst-case behavior for an Ethernet network with mobile hosts, as frequent location updates are required in the network to prevent the blackholing of forwarded frames for extended periods of time. Additionally, the peer-to-peer traffic model generates “work” for each mobile host at random intervals. Each work item creates bidirectional flows to a number of random mobile hosts in the network.

Finally, in our first synthetic traffic model, each host opens a single outgoing flow to another randomly selected host in the network. The duration of the flow is set to the length of the test. Additionally, the host sends packets on that flow at fixed-length intervals (100 ms). This traffic model is used to illustrate the differences in network convergence characteristics between RBRidges and DBRidges.

In our second synthetic traffic model, each mobile host creates one or more flows to one or more static hosts in the network. The static hosts send frames to the mobile host every 10 ms, while the mobile host sends an upstream frame every 5 s on each flow. This traffic model is used to highlight connection interruption characteristics in terms of lost frames from the perspective of the mobile host.

6.3. Mobility model

Our mobility model is based on real-world data from the Netradar project [39]. The Netradar project allows users to measure their Internet connection quality by installing a measurement client to their mobile devices. Each measurement run tests the bandwidth and latency of the connection and collects information such as the network and GPS location of the user during the test. The measurement client also supports “passive measurements” that are run in the background.

We use a data set containing approximately 28,000 active measurements from a period from August 2013 to the end of September 2013 from the Android measurement client. From each measurement, we analyze the network changes of the user and extract the interval of the changes, where the network technology (e.g., UMTS) remains identical. After extensive filtering, we are left with roughly 1500 network change intervals, which we use to build a distribution for the handover frequency rate. The handover frequency data indicates that the typical handover frequency in our test area is heavy tailed. Based on a maximum likelihood estimation on our filtered handover frequency data, we conclude that a lognormal distribution with $\mu = 0.853, \sigma = 1.78$ parameters (also presented in Table 3) will roughly model the frequency distribution and act as a baseline distribution for our evaluation.

We combine the handover frequency distribution with a mobility model that is topology-aware, i.e., the hosts are allowed to move between switches that are direct neighbors of each other. This should roughly model a situation where users with mobile devices move around in a network that contains the base stations (or access points) of a mobile network in a given area.

Furthermore, to evaluate the network effect of mobility, we also use external and host-based signaling to trigger the location updates in the network during the evaluation. The two types of signaling have significant ramifications for the end-host service in the network and the various characteristics we analyze in our evaluation.

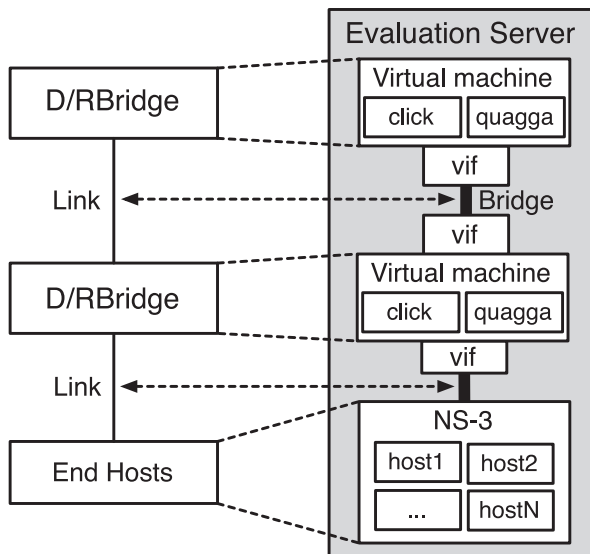


Fig. 5. Virtualized test environment architecture.

External signaling uses an ARP announcement (i.e., a gratuitous ARP request) emitted directly after a mobility event to indicate the location change to the network. This type of behavior is found in many virtualized services that support migration of virtual machines from one physical server to another. On a more general level, any external signaling mechanism (e.g., 802.1X or DHCP) that allows switches to register the attachment of a host would also work. In networks with conventional switches, an ARP announcement will update all MAC learning tables nearly simultaneously, thus minimizing the service disruption period for the end host.

Host-based signaling uses the host's own frames to trigger location updates in the network. As the Ethernet network has no knowledge that a host has moved before it emits a frame, this method of signaling may have significant service disruption, directly proportional to the frame emit frequency of the host. Furthermore, the types of the frames a host emits after a mobility event has a significant effect on the service disruption. Emitted unicast frames will only repair the location information on the path to the destination host, while broadcast frames (typically ARP requests or announcements) repair the network identically with the external signaling mechanism.

6.4. Evaluation environment

Our evaluation environment generates real-time traffic flows inside a virtualized network. The network traffic is generated using the real-time model of Network Simulator 3 (ns-3) [40] that emulates the mobile hosts in the network. The virtual network is realized through several minimalistic virtual machines run on a single server. Each of the virtual machines acts as a single forwarding device (i.e., a switch) in our test topology.

Fig. 5 presents the test environment architecture on a high level. The virtual machine instances (guest) use a minimalistic Linux distribution (OpenWRT [41]) to run our switching software. The guests export a *tap* device in the host operating system for each virtual network interface card (*vif*). The two devices are connected by the emulation server using a Linux bridge component to create a point-to-point link between two guests. The tap devices on the edges of the virtual network are not connected to another guest. The ns-3 simulator uses the edge devices as connection points to the virtual network created by the guests. Each edge tap device can support multiple ns-3 hosts because MAC addresses for the packets emitted by ns-3 can be spoofed. To the virtual net-

work, this presents itself as a single port with multiple hosts behind it.

We use the click modular router (Click) [42] to implement the forwarding plane due to its easy extensibility and flexibility. Click routers are built from elements that are connected together to create a directed acyclic graph for packet processing. We have designed and implemented a forwarding plane implementation in Click for both the standard TRILL protocol as well as our DHT extensions. The control plane portion of the R Bridges base specification is available from Oracle in the OpenSolaris project [43]. It is implemented in Quagga [44], a freely available multiprotocol router suite. We slightly modified the control plane implementation to function outside of OpenSolaris, and we implemented the necessary control plane changes to support our DHT extension.

The ns-3 process generates flows by creating virtual hosts that communicate with each other through the virtual network. Each host implements a full Internet stack and simplistic application layer protocols for data exchange on top of UDP or TCP. Hosts are initially divided into equally sized groups inside the virtual network topology. The number of hosts attached to each switch during the testing varies based on the mobility model used. As we are not interested in the throughput of the network as a whole, our flows send only small frames as either uniformly random or fixed intervals for the duration of the flow.

7. Evaluation results

We have evaluated our system design through specific test cases and analyzed the resulting output generated with the topologies, traffic models, and mobility model described above. To guarantee a level of confidence in our evaluation results, we performed specific test cases multiple times using unique pseudo-random generator seeds in ns-3. This causes the mobile hosts in the network to behave differently during each test run, allowing us to collect more independent events from the network and to analyze certain aspects of the network behavior using averages instead of single-run tests where the effect of the pseudo-random number generator fully dictates the results (barring the minute timing issues imposed by the virtual machine hypervisor).

In the following sections, we first go over the effect of mobility on the signaling load in the network by comparing the amount of “control traffic” in the network between routing bridges and DBridges. We also show how the frequency of mobility events in the network affects the signaling load and analyze the amount of wasted signaling traffic in the network.

Next, we discuss the effects of mobility on the maximum convergence time of the network, again comparing conventional switching with our system design. We first illustrate the behavior through a simple example, and then analyze a set of more detailed results from different traffic models.

We then analyze the effect of the SEATTLE DHT scheme on the average path length of user data frames traversing the network during the convergence of the network. We look at both the path length of the frames themselves and analyze the length of the redirection chains in the network.

The evaluation is concluded with an analysis of flow quality from the perspective of the mobile host. We evaluate the effect of host mobility on incoming flows by analyzing connection interruptions in a number of concurrent downstream flows.

7.1. Signaling load on mobility

The principal benefit of the active location information update mechanism is to eliminate flooding from the Ethernet network by relying unknown destination frames through a separate switch and by creating a proxy ARP-like system to eliminate broadcast

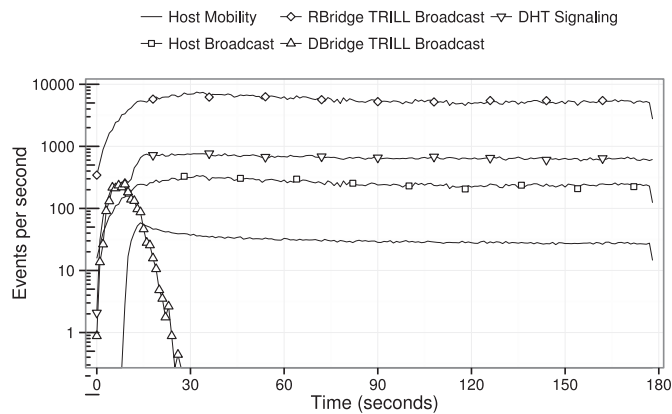


Fig. 6. Network event frequency with peer-to-peer traffic model using explicit announcements.

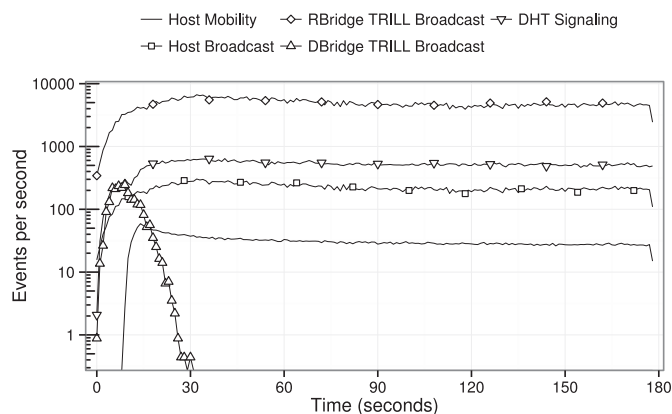


Fig. 7. Network event frequency with peer-to-peer traffic model using host-based triggering.

ARP traffic. In our design, the DHT information management is carried out by a separate signaling protocol. In this section, we evaluate the overhead our signaling protocol has on the network (based on the amount of signaling frames emitted) and compare it to the overhead that flooding causes in networks based on routing bridges. We also analyze how both versions of switches react when we alter the rate of mobility events in the network.

Figs. 6 and 7 present the frequency of various events in a network with peer-to-peer-like traffic model using explicit announcements and host-based triggering of location updates. In the figures, the ARP requests generated by the hosts as part of the traffic model overshadow most of the explicit ARP announcements in conjunction with each finished host mobility event. For example, looking at Fig. 6, out of the roughly 5400 frames per second of TRILL encapsulated broadcasting for routing bridges, only 12% (i.e., $30 \text{ avg_mobility_events} * 22 \text{ links}$) are the result of the explicit announcements.

DBridges eliminate most of the ARP-based broadcasting in the network, replacing it with DHT-based signaling. Only a small amount of broadcasting is necessary at the start of the testing to bootstrap the DHT. The DHT signaling reaches an average of 630 and 510 frames per second using explicit announcements and host-based triggering, respectively. Overall, to enable both host service and mobility support in the network, DBridges use approximately an order of magnitude less traffic to accomplish identical functionality with routing bridges. In summary, the host mobility support in DBridges plays a small role compared to the benefits of eliminating the ARP broadcasting from the network, when the host

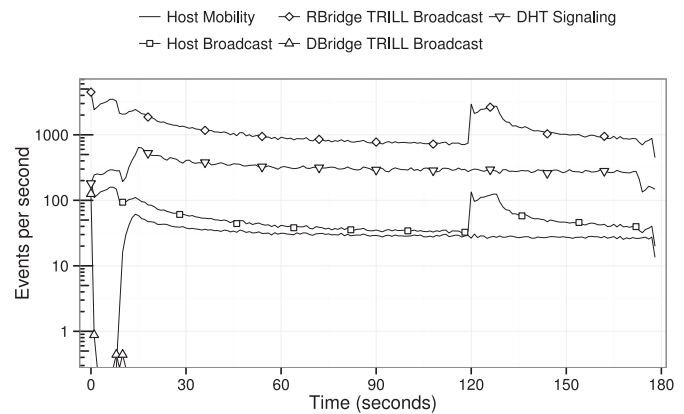


Fig. 8. Network event frequency with downstream traffic model using explicit announcements.

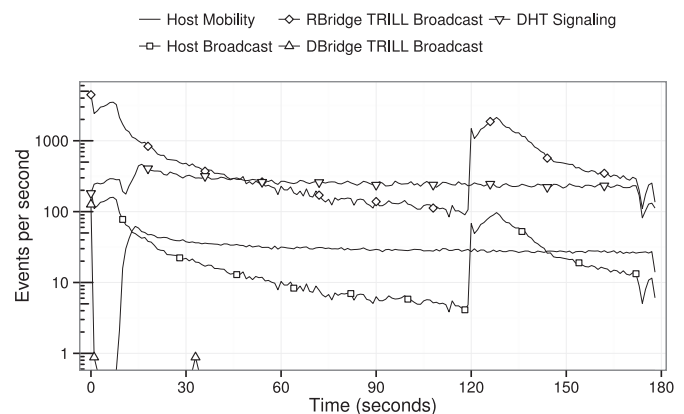


Fig. 9. Network event frequency with downstream traffic model using host-based triggering.

traffic matrix resembles a peer-to-peer communication model. In terms of signaling traffic, if excessive flooding in the network is not an issue in the first place, DBridges will not bring any benefits to host mobility support over passive location information updates.

We also evaluated the signaling load in a network with a more typical traffic model. Figs. 8 and 9 show a similar time series of event frequency in a network where the hosts perform downstream data communication with five static hosts in the network. Here, we see a significant reduction of TRILL encapsulated broadcast traffic compared to the peer-to-peer traffic model, as each mobile host has only five potential hosts it communicates with (i.e., at worst $(\text{mobile_hosts} * \text{static_hosts}) * 2$ requests with downstream, and $\text{mobile_hosts} * (\text{mobile_hosts} - 1)$ with peer-to-peer traffic models). As a result, we can also see significant differences between the explicit announcements and host-based triggering methods because half of the overall broadcast traffic is generated by the explicit announcements.

We can also see the effect of mobile host ARP cache behavior on the overall signaling traffic in both figures. As there are fewer IP-to-MAC mappings for each host to discover, the number of ARP request frames the hosts send (i.e., the host broadcast in the figures) gradually reduces because increasing number of hosts have the IP-to-MAC mappings of all the static hosts in the test. Beginning at 120 s, the ARP cache in the mobile hosts will refresh the mapping entries to ensure their validity.

Comparing the RBridges TRILL encapsulated broadcast and DBridges DHT signaling traffic using explicit signaling, the one-hop DHT functionality reduces the average traffic amount in the test

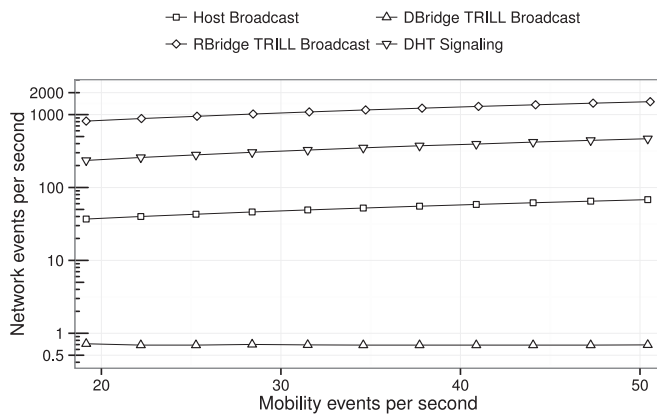


Fig. 10. Network event frequency scaling using downstream traffic model with explicit signaling.

network by approximately 75% over the full duration of the test. Note that during the first 120 s of the test, the host-originated ARP requests (i.e., ARP requests for new IP-to-MAC mappings) reduce to almost zero, as the “host broadcast” curve in the figure meets the “host mobility” curve. This behavior is also reflected in the TRILL broadcast traffic routing bridges generate, while the DBridges-created DHT signaling traffic is independent of the host behavior.

The differences between the systems are more evident with host-based triggering. As can be seen from Fig. 9, the benefit in signaling traffic offered by DBridges is significantly reduced as the test proceeds. Depending on the configuration of the ARP caches in mobile hosts, the traffic matrix, and the size of the topology, networks with routing bridges can actually exhibit less signaling traffic (i.e., broadcast frames) than DBridges with DHT signaling. This behavior is caused by the DHT signaling mechanisms associated with the location information state repair process. DBridges will always update the old location on any host-originated Ethernet frame. Consequently, the repair process will eventually converge all other DBridges in the network to the new location of the mobile host through additional DHT signaling generated by the old location. While this can result in overall traffic that exceeds the broadcast traffic generated by hosts, we will also show in Sections 7.3 and 7.5 that it will improve the quality of service mobile hosts receive in these types of network environments.

Fig. 10 shows the effect of mobility event frequency on the average number of network events (e.g., host broadcasting, RBridge TRILL broadcasts, etc.) during the test case. As can be seen from the figure, both routing bridges and DBridges scale similarly with host broadcasting when the frequency of mobility events in the network increases. In addition, we see that the number of broadcast TRILL frames in the network with DBridges does not depend on the number of mobility events in the network, because the broadcasting is only used in the beginning of the test when the DHT information is incomplete.

Overall, DBridges reduce the average number of signaling-related network events by an order of magnitude in the network in our testing. On the other hand, if we compare the scaling characteristics of the signaling between routing bridges and DBridges, we can actually see that the relevant signaling traffic (i.e., RBridge TRILL broadcast and DHT signaling) increases more rapidly with DBridges. In total, an increase of 164% mobility events per second causes an increase of 85% in broadcasting traffic for routing bridges and an increase of 98% in DHT-signaling traffic for DBridges, making both scale sublinearly with the number of mobility events per second in the network.

The differences between scaling characteristics can be explained by the differences in the signaling model. Routing bridges update the network as a whole through a single broadcast primitive. In practical terms, each broadcast frame is replicated onto 22 links in the test topology. DBridges, on the other hand, use a varying number of signaling messages to update the stale location information in the network. At minimum, a single signaling message is sent to the DHT server storing the location information or the old location of the mobile host. The former case occurs when the old location of the mobile host was the DHT server storing the location information, and the latter case occurs when the mobile host attaches to the DHT server storing the location information. A typical case sees two signaling messages plus a single additional signaling message to update any switch in the network that tries to reach the mobile host through its old location. More importantly, as the frequency of mobility events in the network increases, the reactive update mechanism in DBridges also increases the number of redirection chains in the network. Redirection chains increase the number of signaling messages by one for each switch involved in the chain. Note also that each individual signaling message in the discussion above in relation to the results in Fig. 10 may represent multiple network events, as each message travels one or more links in the test topology.

7.2. Signaling waste

Both active and passive location information updating can create wasted signaling traffic. We define wasted signaling traffic in terms of host communication patterns. Thus, relevant signaling traffic updates location information on switches that have directly attached hosts communicating with the mobile host. Once the mobile host attaches to another switch, most of the existing location information becomes invalid in the network. Consequently, if the information is updated on a switch that has no directly attached hosts communicating with the mobile host during its attachment to a remote switch, the signaling traffic is wasted.

In the majority of traffic matrices, the prolific flooding in networks with passive location information updates creates significant waste in terms of signaling traffic. Each flooded frame will update location information in all MAC learning tables in the network segment, regardless of the relevance of the information. As a result, the amount of waste is directly dependent on the traffic patterns in the network.

Active location information updates happen reactively, so they are only sent to switches with stale location information. Consequently, the signaling traffic is always relevant at that moment in time, even though timing-related issues may invalidate the location information during the traversal of the frame. Typically, wasted signaling traffic can occur due to redirection chains and the underpinnings of the Ethernet communication model itself. If the host does not emit a frame after it attaches to a new switch and there are no external signaling mechanisms to detect the attachment, the reactive update mechanism will act as if the host is still at the old location.

Fig. 11 presents the amount of signaling frames wasted per mobility event with the downstream traffic model. The largest waste in terms of signaling traffic occurs with routing bridges using explicit signaling. Out of the 23 switches in our test topology, at most five switches (i.e., the switches that have the directly connected static hosts) in the network actually need the updated location information, leaving 78% of the delivered frames wasted. We see a steady step-wise climb from 75% all the way to 100% in the number of frames wasted. Each step approximately matches the percentage of switches out of the total number of switches in the network that can have hosts actively communicating with the mobile host when it performs the mobility event. For example, if the mo-

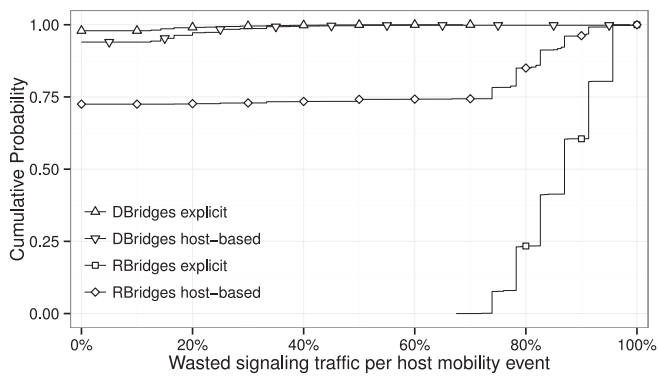


Fig. 11. Signaling traffic waste per host mobility event in the downstream traffic model.

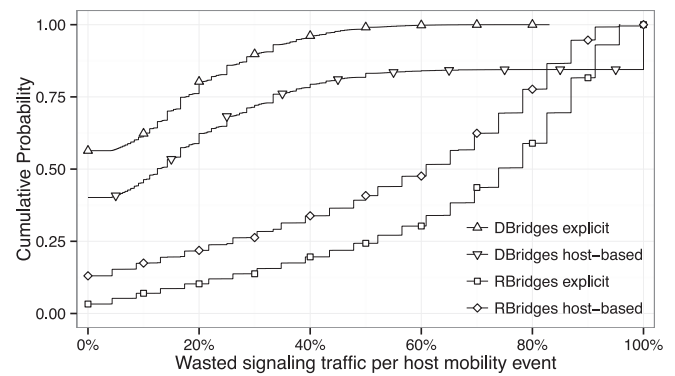


Fig. 12. Signaling traffic waste per host mobility event in the peer-to-peer traffic model.

mobile host has an open flow to each of the 5 static hosts, the waste for that mobility event is 78% (i.e., the switches with the 5 static hosts need the location update, and 18 other switches do not require the information).

DBridges waste practically no signaling traffic in the downstream traffic model using explicit signaling. Since the explicit signaling updates both the DHT server and the old location almost instantaneously, a large majority of the DHT signaling messages are relevant (i.e., updates to switches with stale location information). Redirection chains can still occur with explicit signaling since the location updates are sent reactively. If the interval between a static and a mobile host communicating spans multiple switches where the mobile host has been attached, the first hop switch of the static host will forward the frames to the last known location. In this case, the length of the redirection chain is the number of switches visited during the interval, and the number of wasted signaling messages generated is the length of the redirection chain, minus the current location of the mobile host.

With host-based triggering, the majority of signaling in the network is relevant in both architectures. Approximately 73% of the signaling events using routing bridges waste no traffic, and we only see significant waste in a similar situation with the previous case, i.e., when a broadcast frame updates the whole network to the new state but only a small number of switches require the information.

DBridges generate slightly more waste using host-based triggering compared to explicit signaling. The primary reason for this is the fact that if a mobile host has previously been active in the network but remains silent after a mobility event, the network will generate wasted traffic when updating switches with stale information. Each signaling frame that updates the location information in switches during the silent period is wasted, because the information points to the old location of the host.

We have also measured signaling traffic waste with the peer-to-peer traffic model, presented in Fig. 12. The overall waste is significantly different from the downstream traffic model, mostly because in the peer-to-peer traffic model, mobile hosts communicate with each other. The step-wise trend is still evident, especially with routing bridges. However, the increase based on the potential five static hosts in the downstream case has been replaced with a significantly higher number of steps, which aligns with the number of switches in our evaluation topology.

Routing bridges still generate significantly more waste in terms of signaling traffic. With explicit signaling, each step is in relation to the number of active hosts communicating with the mobile host during its attachment to a switch. Thus, the waste can theoretically vary from 0% (i.e., the mobile host has active flows with other hosts in the network behind all other switches) to 100% (i.e., no

active flows). The difference between explicit announcement and host-based triggering is also significantly smaller than with the downstream traffic model. For explicit signaling, this effect is a direct result of the number of hosts communicating in the network. As there are more communicating host pairs in the network at any given time, there are also more relevant switches where the location information must be updated, causing less waste to be generated.

DBridges also create significant waste with the peer-to-peer traffic model, even when using explicit signaling. The root cause for the waste is identical with the downstream traffic model (i.e., redirection chains). However, due to the number of communicating hosts in the network, the setup creates significantly more redirection chains than the downstream traffic model. Note that with the peer-to-peer traffic model using explicit signaling, DBridges still have an order of magnitude less signaling waste in the network, compared to routing bridges.

Host-based triggering relies on the mobile hosts to emit frames after they have attached to another switch. This is evident with routing bridges, where the distribution of wasted signaling traffic closely follows explicit announcements instead of resembling the distribution in the downstream traffic model. With routing bridges, the frame types play a significant part in how many signaling frames are wasted while the mobile host stays attached to a switch. If it only emits unicast frames, there is very little waste because the majority of those frames are delivered to switches with actively communicating peers. However, due to the number of communicating pairs in the network, the testing sees constant ARP request traffic in the network, as seen in Section 7.1. A broadcast frame makes the mobility event identical with the explicit announcement mobility events, i.e., the amount of waste is directly dependent on the number of active flows between hosts behind other switches in the network.

DBridges show a very similar behavior for wasted signaling traffic between host-based triggering and explicit announcements. As mentioned in Section 5, the location information update process in the network is triggered in the same way, regardless of the frame type. Networks with host-based triggering do generate more signaling traffic waste for similar reasons as in the downstream traffic model case. The old switch is not informed of the new location nearly instantaneously. Until the host emits a frame, the old switch informs stale switches of the wrong location (i.e., itself). As an extreme case, the host may also stay completely silent during its attachment to a switch, at which point all the signaling traffic relating to the host is wasted. This can also be seen in the figure as a significant number of mobility events (approximately 15%), where all of the signaling traffic is wasted.

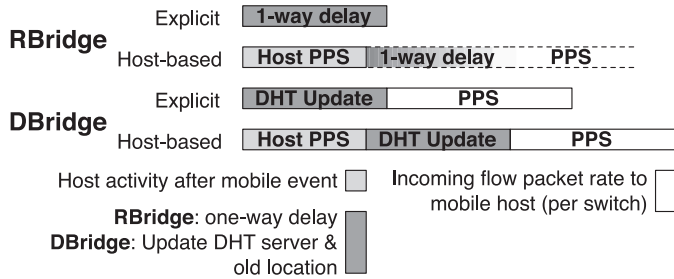


Fig. 13. Convergence time in the network.

7.3. Network convergence after a mobility event

We can measure and compare the convergence time of a mobility event with routing bridges and DBridges. We define the convergence time as the duration it takes for the relevant switches to update to the new location information of the mobile host after the mobility event finishes. Relevant switches are the set of switches in the network that have directly attached hosts actively communicating with the mobile host when the mobility event occurs.

Fig. 13 illustrates the convergence time, comprised of several intervals. Convergence time for a mobility event consists of one to three intervals, depending on the signaling method, there is an interval of time before the mobile host becomes active. With explicit signaling, the activity is near instantaneous, and with host-based triggering, the interval depends on the frame emit frequency of the mobile host (host PPS).

Next, the activity updates a number of switches in the network. With explicit signaling, routing bridges are updated throughout the network, and the interval and convergence time can be reduced to a maximum one-way delay in the network, i.e., when the broadcast frame reaches the last switch. With DBridges, the activity allows the first hop switch of the mobile host to update the location information in the DHT server and subsequently in the old location (DHT update).

With host-based triggering, convergence time with routing bridges primarily depends on the frame types the host sends. Once the host emits a broadcast frame, the convergence time is identical with explicit signaling; i.e., the frame updates all switches in the network. If the host only sends unicast frames, only the first-hop routing bridges of the destination hosts are updated. In the end, the convergence time for routing bridges with no explicit signaling is at most the MAC table timeouts on the switches. For DBridges, the frame types are irrelevant, and the first emitted frame after a mobility event will trigger the location information update process in the network.

At this point, DBridges have only updated the DHT server and the old location. Any DBridge that has a directly attached host communicating with the mobile host will still forward frames to the old location. The final interval in the total convergence time is governed by the rate of incoming packets from the various peers communicating with the mobile host. Each frame sent to the old location will update the location information in the switch that sent it; thus the network is converged when the last relevant switch with stale information is updated. Similarly to routing bridges, the maximum convergence time for DBridges without explicit signaling is at most the MAC table timeout on the switches.

We begin the evaluation by looking at a simplified traffic model, where each host opens a single flow at the beginning of the test to another randomly selected host in the network. The flow stays active for the duration of the test, sending frames to its peer every 100ms and the peer echoing the frame back to the host. In total, each mobile host has a single upstream and a single downstream

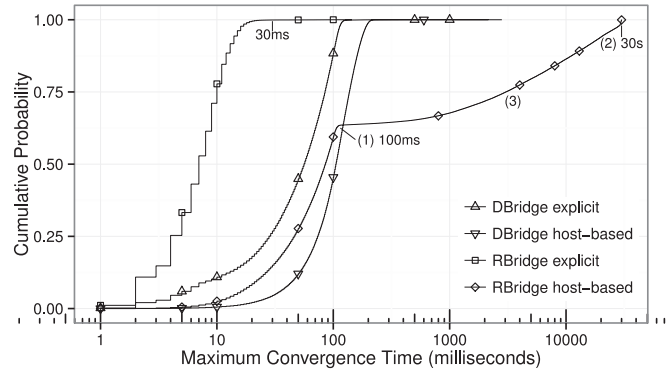


Fig. 14. Convergence time of a mobility event with a simplified traffic model.

flow to two randomly selected hosts in the network. We introduce mobility in the network using our default mobility model, described in Section 6.3. Finally, to emulate timing out of stale information in MAC tables due to host inactivity, we reduce the MAC table timeout for location information to 30 s in routing bridges. We also similarly reduce the remote host location information timeout in DBridges to 30 s.

Fig. 14 presents the convergence time of mobility events. With the traffic model, we can quantify the intervals in Fig. 13. The activity interval (host PPS) of the host after a mobility event is the host frame send interval if we use host-based triggering, i.e., 0–100 ms. For explicit signaling, the activity interval is zero.

The other significant part of the convergence time is the interval between mobile host activity and the location update in the first-hop switch of the downstream peer (PPS in Fig. 13). Routing bridges rely on packets sent by the mobile host to update its location on switches, resulting in an additional delay of 0–30 s. DBridges rely on the frames sent by the mobile host to initiate a location update in the whole network, resulting in an additional delay of only 0–100 ms (based on the downstream peer frame send interval).

7.3.1. Routing bridges

Routing bridges using explicit signaling converge very quickly, as the broadcasted ARP announcement reaches each switch in the network in a few milliseconds. The convergence time is directly dependent on the number of hops the broadcast frame travels in the network, and the processing delays in each intermediate switch. Over 99% of the mobility events converge within 30ms. The rest can be accounted to various timing issues and processing delays on the virtual machines during the testing.

The convergence time for a mobility event with host-based triggering is split into two independent phases divided by an inflection point in the curve. The mobile host frame emit interval adds (1) a uniform delay of up to 100 milliseconds to the convergence time. As can be seen from the figure, approximately 65% of the mobility events converge within this interval. In practice, this first 100 ms of convergence events consist of three separate cases.

In the best case, if the downstream and upstream peers of the mobile host are attached to the same switch, any frame sent will update location information in a shared switch. The location information update in the shared switch enables the downstream peer to send frames to the mobile host. In addition, it is also possible that the downstream peer of the mobile host is attached to the same switch as the mobile host. In this case, a frame emitted to the upstream peer of the mobile host will update the local switch first, allowing the downstream peer to correctly send frames to the mobile host. Finally, due to the mobility model, it is possible that the downstream peer of the mobile host attaches to a switch that

has up-to-date location information for the mobile host in its MAC learning database. This event may occur at any point after a mobility event, including the first 100 ms.

At the extreme (2), the convergence time is the MAC learning table timeout in the switches (i.e., 30 s). Here, the switch where the downstream peer of the mobile host is attached has not received a frame from the mobile host in the last 30 s. The next frame the downstream peer sends is flooded to the network, reaching the mobile host. The mobile host subsequently replies to the flooded frame, updating the location information on the first hop switch of the downstream peer.

Finally, after the inflection point in the curve, the events (3) converge anywhere between 100 ms and 30 s. These events correspond to a behavior where the downstream peer of the mobile host randomly attaches to a switch that has up-to-date location information for the mobile host. The state of the information is a result of receiving a frame from the mobile host. That frame may have been sent either during the current attachment of the mobile host or during any of the past attachments to the same switch in the last 30 s.

7.3.2. DBridges

We can see that the convergence behavior of flows is very similar between the two signaling models when the network is fully deployed with DBridges. As described earlier, the convergence time for DBridges consists of two independent phases.

First, the location information update process begins when the mobile host emits a frame at its new location. In our simplified test case, this occurs immediately with explicit signaling, and between 0 and 100 ms with host-based triggering. Next, the time is increased by the link and processing latency of the DHT signaling when the new location of the mobile host updates the DHT server of the location and the subsequent signaling of the new location information to the old location of the mobile host. This adds a few milliseconds of delay to the convergence time. Once the old location is updated, the convergence time increased in proportion to the send interval of the incoming flow peer of the mobile host. For both signaling models, this interval is also 0–100 ms in our simplified test case. Finally, the convergence time is increased by a few milliseconds as the old location informs the switch of the incoming flow peer that its location information is stale.

In total, the maximum convergence time for DBridges is roughly 115 ms with explicit signaling (i.e., a maximum incoming flow peer send interval of 100 ms, plus 15 ms of link latency and processing delay) and roughly 215 ms using host-based triggering. The only difference between the signaling models is that host-based triggering adds an additional maximum of 100 ms of time to the total convergence time. However, there are very rare cases with both models where the signaling frame reordering causes the convergence time to increase to up to five seconds.

In rare cases, when redirection chains are created to forward the frame towards its current location and multiple switches signal their intermediate location information to the switch that originated the frame, two signaling frames can get reordered in the network due to shortest-path forwarding. In this case, the switch that hosted the stale location information will not end up with the current location of the destination host when the redirection process is finished. It will continue sending frames to a wrong switch that will continue redirecting them towards the correct destination switch. To throttle DHT-related signaling in the network, there is a 5-s threshold value in DHT signaling updates to switches that relate to redirection. End-host service is not disrupted during this period, but rather the frames are traversing a non-optimal path in the network for a longer period of time.

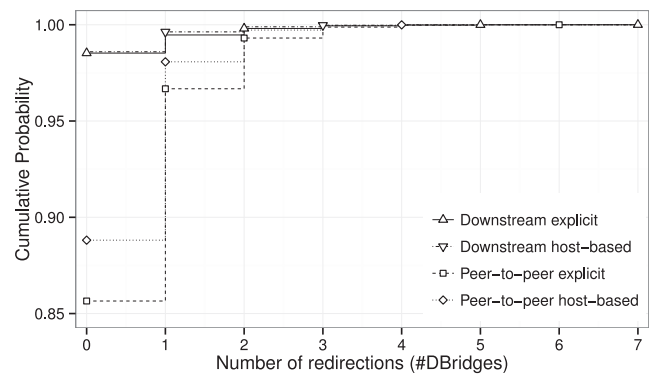


Fig. 15. Redirection chain length of active flows after host mobility event.

7.4. Effect of redirection on path length

Frame redirection occurs when a host attaches from one switch to another and emits any frame that begins the state repair process in the network. Initially, the old switch is updated with the new location, which then reactively propagates this information to other switches in the network, which try to reach the mobile host through the old switch. As hosts move around in the network, the network may be in a state where multiple redirections are required to deliver the Ethernet frame to the mobile host.

We evaluate several things in this section that relate to the state repair process in the network. First, we show how the path length of host Ethernet frames changes with frame redirection during the convergence period in the network after a host attaches to a new switch. Path lengths are normalized to the optimal path length for each source–destination switch pair. Optimal path length is determined by the SPF trees built for each switch. Next, we look at the effect of mobility on the network by analyzing the number and length of redirection chains in the network.

Table 4 presents the normalized path length of Ethernet frames received by the mobile host while the network has not yet converged to the new location information. As we see from the table, the distribution of normalized path lengths between the traffic models (i.e., peer-to-peer and downstream) is nearly identical, as the path lengths are principally affected by the diameter of the topology.

Compared to routing bridges, DBridges typically introduce approximately half a hop of extra path to the frames traversing the network. However this increase is only relevant when the switch that encapsulated the Ethernet frame has stale location information for the mobile host. Also note that in terms of number of hops, DBridges may in some rare cases, depending on the topology and the SPF trees, forward traffic in a way that actually reduces the number of physical hops traversed in the network. In our evaluation, an extra half hop introduces approximately 500 μ s of latency to the frame delivery, compared to routing bridges.

We also measured the length of redirection chains in the network caused by host mobility. Fig. 15 shows redirection chain lengths in our topology, with both traffic models and signaling schemes. The overwhelming majority (over 87%) of host mobility events cause no redirection chains in the network, regardless of the traffic model or signaling scheme used.

The differences in the number of redirection chains between the traffic models are explained by the difference in communication patterns. Hosts using the peer-to-peer model may communicate with a larger number of hosts in the network at any given time, generating a significantly higher number of active flows. Each active flow after a mobility event (i.e., after either initiator or the destination moves) causes a redirection event in the network.

Table 4
Normalized path length of Ethernet frames during convergence in a fully deployed DBridge network.

Traffic model	Min	10th Pct.	Median/mean	90th Pct.	Max
Peer-to-peer explicit	0.67	1	1.5/1.68	3	12
Downstream explicit	0.67	1	1.5/1.73	3	6
Peer-to-peer host-based	0.67	1	1.5/1.72	3	13
Downstream host-based	0.67	1	1.5/1.74	3	9

This increases the probability that a redirection chain is created if the destination switch of the mobility event has stale information. With the downstream traffic model, each mobile host has only five separate switches (i.e., the static servers each mobile host creates random flows to) that require redirection after the mobile host attaches to a new switch.

There are also differences between the two signaling schemes we use to trigger location information updating in the network. Explicit signaling generates both more frequent updates in the network and also more numerous redirection chains. This is a direct result of the signaling model. If the mobile host remains silent while attached to a switch, the potential redirection chain that the mobile host can generate in the network is shortened by one, as the chain will have skipped the switch that never received a frame from the mobile host during its attachment. In addition, silent hosts reduce the number of location updates in the network, thus reducing the amount of zero-length redirection chains.

7.5. Flow connection interruptions for mobile hosts

Mobility in conventional Ethernet networks is a best-effort service. Without explicit handover protocols to orchestrate the process, the network can lose packets when the mobile host switches from one location to another. For the applications running on the mobile host, this presents itself as data loss or delays in data delivery. The more rapidly the network can update the location information in the network, the shorter the service disruption is.

We evaluate host connection interruption by creating downstream flows from each mobile host to a static host in the network that emits packets at fixed 10-millisecond intervals using UDP, mimicking a streaming service. At the mobile hosts, we measure the packet inter-arrival rate as they move in the network and normalize the results to the packet send frequency (i.e., a value of 1 or less indicates that no connection interruption was detected during the mobility event). The mobile hosts also send upstream packets to the static host at 5-s intervals, selected based on the RTP [45] fixed minimum interval value for transmission of control packets.

Fig. 16 presents the connection interruption time the application stack of the mobile host sees after a mobility event. In a simple scenario where each mobile host is connected only to a single host at a time, both systems are seeing identical connection interruption times in the respective signaling models.

With explicit signaling, routing bridges will update to the new location based on the one-way delay to the respective static host the mobile host is communicating with. Over 99% of the mobility events cause a connection interruption no longer than 20 ms (i.e., 2 units of 10 ms), regardless of the system used. DBridges, on the other hand, update the old location of the mobile host, incurring a “one-way delay”, first from the new location to the DHT server and then from the server to the old location. In practice, the connection interruption time with DBridges is identical with routing bridges, as the location update process finishes within a similar time as the one-way delay of the ARP request propagation in the network.

Note that with both systems, we also see a very small fraction of connection interruptions, where the normalized time is approx-

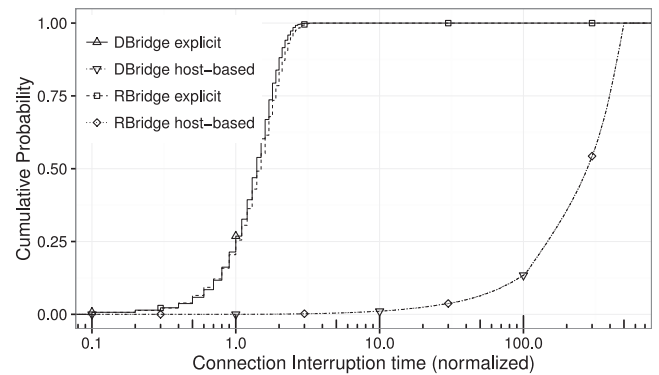


Fig. 16. Mobile-host connection interruption time per mobility event using downstream traffic.

imately 100 units (i.e., one second of real time in our evaluation). This is related to ARP functionality in ns-3, when the initial connection to the static host is created. If the mobile host initiates the connection right before a mobility event, it is possible to send the ARP request (either broadcast, or unicast in the case of DBridges), and get the reply to the old location of the mobile host. As a result, the mobile host is unable to resolve the IP-to-MAC mapping of the static host it is connecting to, causing the ns-3 instance to send another ARP request to the network one second after the initial request is sent.

There are also no differences between the behavior of routing bridges and DBridges when using host-based signaling to trigger location changes in the network. In both cases, the connection interruption time is dictated by the mobility model and the fixed upstream packets sent every 500 normalized units. Note that the minimum mobility interval in this evaluation is set to one second, which shows as a small inflection point at the normalized time value of 100. The mean connection interruption time for host-based triggering is approximately 270 units, consisting of the mean upstream packet interval (i.e., 250 units), a one-way delay to the static host in the network (approximately one unit), the mean of the static host packet send interval (half an unit), and finally a one-way delay back to the mobile host (approximately one unit).

Fig. 17 presents a similar evaluation, where the number of flows from each mobile host is increased to 3 and 5. In the test, each mobile host connects to three randomly selected or all static hosts in the network. Each flow replicates the test conditions of the one-flow case, including the five-second upstream send interval, and uses host-based signaling to trigger the location update in the network. In the figure, we denote the one-flow case as the “baseline” and compare the connection interruption time of all flows from the mobile hosts with three and five concurrent flows.

Routing bridges only see a nominal benefit from the increased flow counts because they only update location information based on received Ethernet frames. This leads to a situation where mobile hosts individually repair each flow after mobility event when the upstream packet is sent. Note that since the minimum mobility event interval can be as short as one second (100 normalized

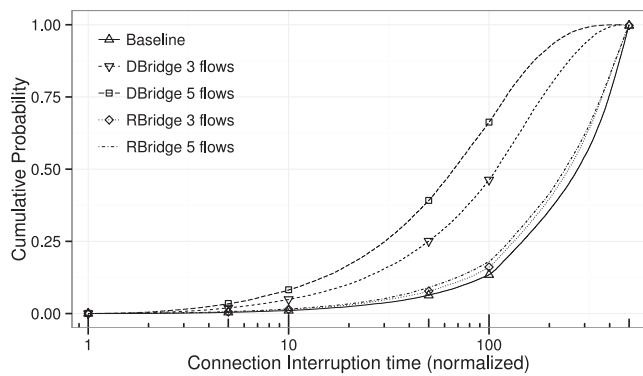


Fig. 17. Connection interruption time as a function of flows, using host-based triggering.

units), hosts can have several mobility events between each upstream packet sent. As there is no broadcast traffic after the initial flow creation, increasing the number of flows by establishing them to different static hosts in the network does not significantly improve the connection interruption times of the independent flows.

DBridges, and more generally the one-hop DHT mechanism in SEATTLE are designed so that any emitted frame from a host will begin the location update process in the network, regardless of the frame type. As a result, we see a direct reduction of connection interruption time when the mobile hosts use three and five separate flows to different static hosts in the network. An upstream frame emitted from the mobile host will begin a location update process in the network and update all stale location information in the first hop DBridges of the static hosts. Additionally, the connection interruption time for our solution is also significantly more uniform, typically repairing all the flows after mobile host activity in a few milliseconds.

The nominal improvement is a result of an increased probability for the sent frame to update the location information in two switches that have attached static hosts. When a mobile host sends a frame to a static host in the network and the first hop switch is shared by both the mobile host and another static host with a flow to the mobile host, the single sent frame updates two relevant switches. As more flows are created, the probability that the host is active after a mobility event and the probability that the mobile host happens to attach to a switch with a static host is increased.

8. Conclusions

We have presented an extensive evaluation of link layer mobility in Ethernet through comparison between conventional Ethernet behavior (i.e., routing bridges) and our DBridges proposal. To our knowledge, this is the first extensive evaluation of the effect of mobility in Ethernet networks, both from the perspective of the switches and the mobile hosts.

While Ethernet, without any explicit handover support, remains a best-effort service with regards to mobility, there is still the question of the efficiency and quality of service for the two typical signaling cases to handle mobility on the link layer (i.e., explicit signaling after mobility or host-based triggering). We show that a location-aware solution such as DBridges in most cases behaves significantly more efficiently than conventional Ethernet, and at worst is on par with it. In addition, we also show that DBridges can bring real benefits to end-hosts in the network at the cost of potentially incurring more mobility-related signaling in the network when compared to standard Ethernet. With host-based signaling, standard Ethernet networks may exhibit long periods of user data blackholing due to invalid location information in the

network. Conversely, DBridges allow the whole network to converge to the current location of the end-host as long as the end-host emits a single Ethernet frame to the network.

Mobility without explicit handover signaling brings a host of issues into play (such as transient forwarding loops), typically related to the timing of events inside the Ethernet network. We rationalize and show through our evaluation that while these problems exist in DBridges, they are exceedingly rare and are mitigated by either the RBridges base specification (through a hop count), by the design of DBridges, or due to practicalities of the network.

We conclude that while the SEATTLE one-hop DHT scheme (and more generally, a system that explicitly tracks location updates in the Ethernet network) is not primarily designed to support host mobility, it is well-suited for it. We can improve the quality of service for most mobile hosts in the network while simultaneously retaining the improved scalability characteristics of the scheme. Furthermore, the improved scalability characteristics reduce both the amount of signaling in the network, and improves its efficiency. The efficiency also directly translates to less end-host processing in networks where explicit signaling via ARP announcements is used.

Acknowledgments

This work was performed in the SIGMONA research project funded by the Finnish Funding Agency for Technology Innovation and industry.

References

- [1] G. Jung, M. Hiltunen, K. Joshi, R. Schlichting, C. Pu, Mistral: dynamically managing power, performance, and adaptation cost in cloud infrastructures, in: *Distributed Computing Systems (ICDCS)*, 2010 IEEE 30th International Conference on, 2010, pp. 62–73.
- [2] Z. Xiao, W. Song, Q. Chen, Dynamic resource allocation using virtual machines for cloud computing environment, *Parallel Distrib. Syst. IEEE Trans.* 24 (6) (2013) 1107–1117.
- [3] C. Clark, K. Fraser, S. Hand, J.G. Hansen, E. Jul, C. Limpach, I. Pratt, A. Warfield, Live migration of virtual machines, in: *Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation - Volume 2*, in: *NSDI'05*, USENIX Association, Berkeley, CA, USA, 2005, pp. 273–286.
- [4] Port-Based Network Access Control, Standard, IEEE Computer Society, 2010. IEEE Std 802.1X.
- [5] C. Perkins, et al., IP Mobility Support for IPv4, Revised, RFC 3344, IETF, 2010.
- [6] C. Perkins, D. Johnson, J. Arkko, Mobility Support in IPv6, RFC 6275, IETF, 2011.
- [7] IEEE Trial-Use Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation, Standard, IEEE Computer Society, 2003. IEEE Std 802.11F (Withdrawn 2006).
- [8] Air Interface for Broadband Wireless Access Systems, Standard, IEEE Computer Society, 2012. IEEE Std 802.16.
- [9] Low-Rate Wireless Personal Area Networks (LR-WPANs), Standard, IEEE Computer Society, 2011. IEEE Std 802.15.4.
- [10] Media Independent Handover Services, Standard, IEEE Computer Society, 2008. IEEE Std 802.21.
- [11] G. Chiruvolu, A. Ge, D. Elie-Dit-Cosaque, M. Ali, J. Rouyer, Issues and approaches on extending ethernet beyond lans, *Commun. Mag. IEEE* 42 (3) (2004) 80–86.
- [12] D. Allan, N. Bragg, A. McGuire, A. Reid, Ethernet as carrier transport infrastructure, *Commun. Mag. IEEE* 44 (2) (2006) 95–101.
- [13] B. Stephens, A. Cox, W. Felter, C. Dixon, J. Carter, Past: scalable ethernet for data centers, in: *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, in: *CoNEXT '12*, ACM, New York, NY, USA, 2012, pp. 49–60.
- [14] M. Scott, A. Moore, J. Crowcroft, Addressing the scalability of ethernet with moose, in: *Proceedings of the DC CAVES Workshop*, 2009.
- [15] N. Varis, J. Manner, M. Sarella, T. Kiravuo, Dbridges: flexible floodless frame forwarding, *Comput. Netw.* 57 (17) (2013) 3601–3616.
- [16] C. Kim, M. Caesar, J. Rexford, Floodless in SEATTLE: a scalable ethernet architecture for large enterprises, in: *SIGCOMM*, ACM, 2008, pp. 3–14.
- [17] R. Perlman, D. Eastlake, D. Dutt, S. Gai, A. Ghanwani, Routing bridges (rbridges): base protocol specification, RFC 6325, IETF, 2010.
- [18] B. Han, V. Gopalakrishnan, L. Ji, S. Lee, Network function virtualization: challenges and opportunities for innovations, *IEEE Commun. Mag.* 53 (2) (2015) 90–97.
- [19] M. Kuroda, M. Inoue, A. Okubo, T. Sakakura, K. Shimizu, F. Adachi, Scalable mobile ethernet and fast vertical handover, in: *Wireless Communications and Networking Conference*, 2004. WCNC. 2004 IEEE, pp. 659–664 Vol.2.

- [20] P. Ameigeiras, J.J. Ramos-munoz, L. Schumacher, J. Prados-Garzon, J. Navarro-Ortiz, J.M. Lopez-soler, Link-level access cloud architecture design based on SDN for 5G networks, *IEEE Netw.* 29 (2) (2015) 24–31.
- [21] J. Banik, Y. Luo, J. Seawright, J. Xu, M. Tacca, A. Fumagalli, B. Sarikaya, L. Xue, A software defined semi distributed mobility management system based on layer 2 backhaul network, in: *Proceedings of the IEEE 82nd Vehicular Technology Conference (VTC Fall)*, 2015, pp. 1–5.
- [22] N. Varis, J. Manner, J. Heinonen, A layer-2 approach for mobility and transport in the mobile backhaul, in: *Proceedings of the 11th International Conference on ITS Telecommunications (ITST)*, 2011, pp. 268–273.
- [23] F. Van Quickenborne, F. De Greve, F. De Turck, P. Demeester, Managing ethernet aggregation networks for fast moving users, *Commun. Mag. IEEE* 44 (10) (2006) 78–85.
- [24] J. Mudigonda, P. Yalagandula, M. Al-Fares, J. Mogul C., SPAIN: Cots data-center Ethernet for multipathing over arbitrary topologies, in: *NSDI, USENIX*, 2010, pp. 265–280.
- [25] T.L. Rodeheffer, C.A. Thekkath, D.C. Anderson, SmartBridge: a scalable bridge architecture, in: *SIGCOMM, ACM*, 2000, pp. 205–216.
- [26] M. Casado, T. Garfinkel, A. Akella, M.J. Freedman, D. Boneh, N. McKeown, S. Shenker, SANE: a protection architecture for enterprise networks, in: *Proceedings of the 15th Conference on USENIX Security Symposium - Volume 15*, in: *USENIX-SS'06*, USENIX Association, Berkeley, CA, USA, 2006.
- [27] M. Casado, M.J. Freedman, J. Pettit, J. Luo, N. McKeown, S. Shenker, Ethane: taking control of the enterprise, in: *Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, in: *SIGCOMM '07*, ACM, New York, NY, USA, 2007, pp. 1–12.
- [28] A. Greenberg, J.R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D.A. Maltz, P. Patel, S. Sengupta, VL2: a scalable and flexible data center network, in: *SIGCOMM, ACM*, 2009, pp. 51–62.
- [29] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, OpenFlow: enabling innovation in campus networks, *SIGCOMM Comput. Commun. Rev.* 38 (2) (2008) 69–74.
- [30] L. Fang, F. Chiussi, D. Bansal, V. Gill, T. Lin, J. Cox, G. Ratterree, Hierarchical SDN for the hyper-scale, hyper-elastic data center and cloud, in: *Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research*, in: *SOSR '15*, ACM, 2015, pp. 7:1–7:13.
- [31] K. Agarwal, C. Dixon, E. Rozner, J. Carter, Shadow MACs: scalable label-switching for commodity ethernet, in: *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*, in: *HotSDN '14*, ACM, 2014, pp. 157–162.
- [32] Intermediate System to Intermediate System Intra-Domain Routing Information Exchange Protocol, Standard 10589:2002, ISO/IEC, 2002.
- [33] H. Zhai, et al., TRILL: ESADI (End Station Address Distribution Information) Protocol, Internet-Draft, IETF, 2014.
- [34] D. Karger, E. Lehman, T. Leighton, R. Panigrahy, M. Levine, D. Lewin, Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the World Wide Web, in: *STOC, ACM*, 1997, pp. 654–663.
- [35] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, H. Balakrishnan, Chord: a scalable peer-to-peer lookup service for Internet applications, in: *SIGCOMM, ACM*, 2001, pp. 149–160.
- [36] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, W. Vogels, Dynamo: amazon's highly available key-value store, in: *SOSP, ACM*, 2007, pp. 205–220.
- [37] Rocketfuel, Rocketfuel Maps and Data, <http://www.cs.washington.edu/research/networking/rocketfuel/>.
- [38] NetEm, netem - Linux Network Emulation, <http://www.linuxfoundation.org/collaborate/workgroups/networking/netem>.
- [39] S. Sonntag, L. Schulte, J. Manner, Mobile network measurements - it's not all about signal strength, in: *Wireless Communications and Networking Conference (WCNC)*, 2013 IEEE, 2013, pp. 4624–4629.
- [40] NS3, Network Simulator 3 (ns-3), <http://www.nsnam.org>.
- [41] openwrt, OpenWrt - Wireless Freedom, <https://openwrt.org/>.
- [42] E. Kohler, R. Morris, B. Chen, J. Jannotti, M.F. Kaashoek, The click modular router, *ACM Trans. Comput. Syst.* 18 (2000) 263–297.
- [43] Oracle, OpenSolaris rbridge (IETF TRILL) Control Plane Support, <http://solaris.java.net/>.
- [44] Quagga, The Quagga Routing Suite, <http://www.nongnu.org/quagga/>.
- [45] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, RTP: A Transport Protocol for Real-Time Applications, RFC 3550, IETF, 2003.