Modeling Tool of TRILL Protocol

Filip Holik, Simeon Karamazov
University of Pardubice,
Faculty of Electrical Engineering and Informatics,
Pardubice, Czech Republic.
filip.holik@student.upce.cz

***Abstract*—This paper proposes the first modeling tool of TRILL protocol. The application implements TRILL features, which include the most fundamental concepts of IS-IS protocol. Simplified STP protocol is also implemented. The application can be used to didactically and interactively present the advantages of the TRILL protocol over STP. Alternatively, the tool can be utilized as a decision tool for TRILL deployment in enterprise networks. The functionality of the TRILL protocol is discussed with emphasis on the main differences between STP. The application was tested on five scenarios of TRILL deployment in a typical enterprise network. Data traffic of these different deployments was compared using the application. Finally, results are further discussed.**

***Index Terms*—Modeling Tool; TRILL Protocol; STP Protocol; Deployment.**

I. Introduction

The rising demand for High Definition (HD) videos, interactive online applications, social media, and online games has increased the amount of data transferred over the Internet. It is estimated that Internet traffic will increase from 6 GB transferred per capita in 2014 to 18 GB during the next four years [1]. This growth is putting higher requirements on modern data centers. These requirements are low latency, high performance, effective power consumption and easy management. In order to comply with such extensive demands, servers in data centers are becoming virtualized. The change from physical to virtualized servers brings additional problems into networking infrastructures. Data networks have to deal with a dynamic change of the topology and increasing throughput demands, while ensuring high availability and redundancy. Due to the extremely high performance requirements, typical data networks in data centers are built as a large flat networks. These networks normally only use the switching technologies built on layer two of the ISO/OSI model. Over the last 25 years, the main bridging protocol used in the data link layer was STP (Spanning Tree Protocol). During the following years, STP protocol was replaced by more advanced variants, like RSTP (Rapid STP) and MST (Multiple STP). Nevertheless, even these upgrades do not ensure adequate performance for modern data centers.

Recently, a new generation of protocols that combines the bridging functionality with technologies from routing protocols was introduced. One of them is the TRILL (Transparent Interconnection of Lots of Links) protocol, firstly introduced in 2011 [2]. The TRILL protocol is significantly improving the behavior of switched networks, when compared to STP protocol. TRILL decreases the failover delays, allows multipath, loads balance among equal costs, and delivers multicast frames more effectively [3].

Current research in the area of TRILL protocol is either focusing on performance comparison [4], or trying to expand TRILL's features [5-9]. Piccolo in [5] is extending TRILL's simple hierarchy into a two-level design, allowing interconnection of two networks without merging them. Ibanez [6] explored the concept of full frame flooding for path exploration. This approach does not require significant complexity of the link-state routing protocol and can lower network latency. Coundron [7] integrates TRILL with additional communication layer protocols to achieve robust multipath communication. Alternative to the TRILL protocol is offering a near zero configuration while outperforming STP was described in [8]. Gohar [9] used TRILL features for encapsulating Proxy Mobile IPv6 messages to reduce network overhead of these messages.

Unfortunately, no current research addresses the need for modeling tools, presenting features and advantages of the TRILL protocol. This is caused by the fact that some of TRILL's features are implemented in hardware. In order to simulate this protocol, an ad-hoc software solution has to be implemented. Currently, there is no such application due to its complexity. The only available approach to emulate a network running a TRILL protocol is to use a software implementation of the TRILL protocol [10], available for Linux Operating Systems. One Linux machine can then act as a switch running TRILL protocol. Using this tool, a TRILL network can be built. Unfortunately, this approach requires multiple devices (either virtualized or physical) and a slow and difficult configuration process, which is not transparent. Moreover, no visualization of current network behavior can be displayed.

We proposed the first application for visual and didactical modeling of the behavior of the TRILL protocol. Our application allows us to model different data networks and compare these networks features if STP, TRILL or their combination is used. The purpose of the application is to interactively show behavior of the TRILL protocol, which can be used for educational purposes. The application can be also used to explore different deployments of the TRILL protocol; and thus help in TRILL implementation in enterprise networks. However, it is important to mention that our application is not the complete simulation of the TRILL protocol, and therefore, cannot be used for precise performance testing.

The rest of the paper is organized as follows. Section II presents the fundamental principles of the TRILL protocol. Section III presents the modeling tool of TRILL protocol and implementation differences between the tool and real TRILL standard. Finally, section IV describes our

application in five different use cases of TRILL deployment in a typical enterprise network.

II. TRILL PROTOCOL

TRILL protocol was developed to overcome STP limitations. Like STP, TRILL is protecting the network from creating loops. However, unlike STP, TRILL uses the best path to reach a destination, load balance of the traffic, foes not block redundant links, and does not require any configuration. Moreover, TRILL is a backward compatible with STP [2]. The basic principle of the TRILL protocol, allowing these features, is the usage of a routing protocol: IS-IS (Intermediate System to Intermediate System). TRILL is using IS-IS for the best path delivery. TRILL is also using additional routing concepts, like reverse path forwarding of multicast frames, or usage of TTL (Time-To-Live).

These features make TRILL ideal for implementation in high demand like in data centers, modern substation networks [11], or sensor networks [12]. In order to support TRILL, transmission of additional information located in the header of the Ethernet frame is required. TRILL uses MAC-in-MAC encapsulation, where the original frame with TRILL header is encapsulated into a new Ethernet frame. This mechanism ensures transparency for end devices and traditional L2 switches not supporting TRILL protocol. These devices use only a standard outer L2 header and do not see an encapsulated TRILL frame. The only field, which TRILL edits in the outer frame is the FCS (Frame Check Sequence), which is calculated from the entire frame instead of just the header. This ensures the integrity of the entire frame, including the encapsulated TRILL frame. TRILL's encapsulation mechanism increases the data size of the frame, which can cause problems on older devices, if they do not support frames larger than 1500B.

The switch, which runs the TRILL protocol is referred to as RBridge (Routing Bridge). RBridge processes two types of data frames: classical frames and frames with a double Ethernet header (encapsulated TRILL header). TRILL header has following fields, which have to be implemented in every TRILL implementation:

i. Ethernet type (EtherType) – containing value of TRILL protocol: 0x22F3
ii. Protocol version (V), Reserved bits (R), Multicast bit (M), Optional bits (OptL), Hop count
iii. Egress nickname – destination RBridge or distribution tree
iv. Ingress nickname – first-hop RBridge

The functionality of the TRILL protocol can be separated into three layers: control, data, and management.

The control layer is responsible primarily for computing the best paths in the network. These paths are computed by IS-IS protocol. IS-IS protocol also ensures convergence by maintaining neighborship via sending hello multicast frames. These messages contain information about network topology and neighbor's status. The control layer also uses different data structures for each forwarding method. Unicast frames are forwarded using a routing table with the list of all RBridges, next hop to reach them, and egress port. Multicast frames forwarding uses distribution trees, which are calculated on every RBridge. The control layer is also responsible for pruning multicast frames. They are not sent to these parts of the networks, where there are no receivers.

The data layer forwards frames to the correct destination and modifies frames accordingly. The most common forwarding mechanism is based on a destination MAC address, but additional hardware implementations exist. Frames are forwarded according to the number of destinations: unicast, multicast or broadcast. Unicast frames are forwarded similarly like on normal L2 switches, except using TRILL routing table and egress nickname for delivery. Multicast frames are marked with an M bit in the TRILL header and forwarded via an appropriate distribution tree. The data layer is also responsible for Fine-Grained Labeling, which allows usage of up to 16 million VLANs. This mechanism is replacing 802.1Q standard with maximum limit of 4096 VLANs.

The management layer allows configuration of the TRILL protocol, if required. In normal conditions, TRILL works without any explicit configuration.

III. MODELING TOOL OF TRILL PROTOCOL

Our modeling tool implements the most important features of the TRILL protocol. The purpose of the application is to present TRILL behavior and its advantages over STP protocol.

A. Application Introduction

The modeling tool implements TRILL protocol, STP protocol and essential parts of the Ethernet and IS-IS protocol. The application uses a *static* modeling system, and therefore, time necessary for conducting calculations and process frames are omitted. The application supports the creation and dynamic modification of the following entities of computer networks: RBridge, switch and end device. Links can be created between every device. All these objects *are temporal, mobile and endogenous*. The model is *discrete* (evolution is modeled per steps) and contains online animation.

The application uses *agent-oriented* architecture. Every active networking device acts as a *reactive agent*. The purpose of these *agents* is to perform an action after receiving a frame. More advanced networking devices interact with the other connected *agents* in order to learn a network topology. This behavior causes the *agent* to become an *initiative agent* – it has to generate traffic itself, according to the inner protocol (CDP, STP, IS-IS) rules.

The application was created in the *Repast Simphony* framework version 2.2. This framework is specialized on *agent-oriented* modeling and simulations. The base language of the framework is Java.

The model allows the comparison of TRILL with STP, because the basic features of the traditional STP are also implemented.

B. Implementation Details

Packages. The model is using four packages, which are displayed in Figure 1. Agents define the logic of different networking devices. GUI is used for graphical representation of the program. Structures contain data structures used by the agents. TrillModel controls modeling and communication with the Repast framework.

The Functionality of the Model. Functionality of the TRILL protocol implemented in the model corresponds to the specification [2], but some features are simplified. These differences are described in the following part.

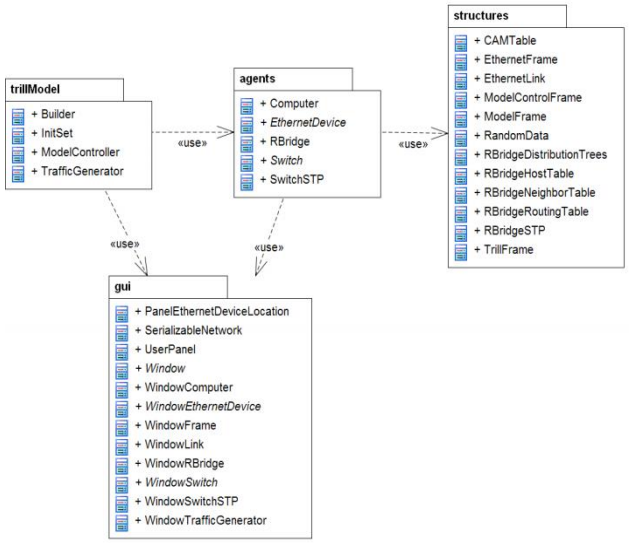

Figure 1: UML diagram of application packages

RBridge Neighborship. Every RBridge periodically sends the IS-IS multicast hello frames. Neighborship is formed, if two RBridges exchange these frames. Every RBridge has its own neighborship table. It is important to mention, that received hello frames are exchanged only amongst adjacent RBridges.

The exchange of information about the topology between the neighbors is simplified and compared to the real implementation. The model contains a central data structure with the topology information. First, the RBridges gets the topology information from this data structure, and it does not have to exchange additional information in the frames. The Neighborship is formed after the single frame is successfully received, and the typical three-way handshake is, therefore not used. The last difference is the topology change, when every RBridge sends the changed information to its neighbors. In the model, frames do not contain this information, because RBridges receives them from the model's data structure. Despite these differences, like in the real protocol, every RBridge runs its own algorithm to calculate the current topology. The behavior of the protocol matches the results found in the real devices.

Routing Table of the RBridge. A routing table is a data structure located on every RBridge. A routing table contains all RBridges located in the network, hops to reach the RBridge, the cost of the link, and the hop count. All this information is implemented in the model, according to the TRILL real implementation. Hop count acts as a security mechanism, similar to the TTL. In case that the packet would be routed to the wrong path of the network, hop count inserted in the packet's header will ensure that the packet will be discarded. This situation can occur if the topology is changed, and TRILL does not have enough time for convergence.

Destination Table. The destination table is another data structure located on every RBridge and it is implemented according to the TRILL standard. This data structure contains a MAC address of every destination device and nickname of the destination of the RBridge. The destination RBridge is the switch, connecting to the particular destination device. This table is used specifically by the switches on the access layer. Switches located in the core layer usually have an empty destination table. The destination RBridge fills a destination table with MAC addresses located in the inner Ethernet frame, and nicknames of the first-hop (or the source) RBridge. In this way, the returning frame can be forwarded straight to the original first-hop RBridge and not via broadcasted frame, sent over a distribution tree.

Distribution Trees. Distribution trees are computed according to the TRILL standard as well. These trees are used in broadcast, multicast and unknown traffic. Unknown traffic is a type, where the destination is not found in the destination table and its location in the network is therefore unknown. Each topology in the model has, at the minimum, one distribution tree computed. Each RBridge needs to know only the device in the root of the tree. Dijkstra algorithm is then used to compute the shortest path to all other devices. This algorithm is run separately on every RBridge.

STP Implementation in RBridge. The RBridge terminates the STP domain in the topology based on classical switches. The RBridge located between the two classic switches separates the STP domain into two areas. This layout has many benefits. Firstly, one domain is not influenced when there is change or failure in the other domain. Further, the convergence within smaller areas is significantly faster. On the other hand, connecting two areas over a single link can result in the link overloads and data drops.

The model also implements additional features needed in the case of interconnecting domains over multiple RBridges. In this case, the designated RBridge has to be chosen in order to prevent sending duplicated frames between domains. The designated RBridge is the only device that is able to send frames between these domains. This device is also responsible for listening STP data traffic and recognizing multiple connections to the same domain. Data traffic has to be sent over a single port, chosen similarly as the root port chosen on classical switches. Other ports are unable to send or receive data frames. RBridges also cannot influence the STP domain in any way, and therefore, they do not generate any STP traffic.

The only difference in implementing these features is that the application does not separate the data traffic according to the different VLANs. Only a single VLAN is supported in the modeling tool.

Structure of the TRILL Frame. The structure of the TRILL frame is equal to the standard. The attributes are defined below:

Structure of the TRILL frame

```
public class TrillFrame implements Cloneable {
  public byte version=1;
  public byte reserved=0;
  public boolean multicast;
  public byte opLength=0;
  public short hopCount;
  public short egressNickname;
  public short ingressNickname;
  public EthernetFrame payload;
  //Methods omitted }
```

Traffic Generation. The application allows setting the volume of data traffic sent between different end nodes. All of the traffic volume is relative, and the links will display the utilization using different colors: black color (<25%), varying dark red hue (25-90%), and red (>90%). After selecting a concrete link, precise utilization can be displayed.

Model Verification. The model is verified according to the TRILL standard [2]. Verification is conducted using empirical methods performed by independent experts. During the verification phase, online animation allows them to check the state of the model in every single step. The model is also compared to the real network running TRILL protocol on different network topologies. All tests show correct model behavior.

IV. Use Case Study Of The TRILL Deployment In An Enterprise Network

The application was tested on a traditional three-layer enterprise network - this topology uses classical L2 switches in the core, distribution, and access layer; with significant redundancy between each layer. This architecture is typically deployed in larger enterprise networks, or in data centers [13]. The application allows testing of large topologies using different variants of networking devices and combining TRILL and STP protocols.

The goal of the study was to create several variants of the possible placements of RBridges in the existing enterprise network. These placements were examined in order to achieve optimal data traffic utilization. Study examined following scenarios:

i. Scenario 1: Classical L2 network with no RBridges
ii. Scenario 2: Ineffective deployment in access and distribution layer
iii. Scenario 3: Effective deployment in access and distribution layer
iv. Scenario 4: Effective deployment in core and distribution layer
v. Scenario 5: Effective deployment on all devices

These scenarios are displayed in Figure 2. The link colors correspond to the traffic utilization described in the "Traffic generation" section.

Scenario 1. The diagram shows the link utilization when traditional STP, using one VLAN, is used. It is clearly visible, that active links are highly utilized while blocked links are not utilized at all. These links can become active only if the primary ones fail.

Scenario 2. This is an example of the wrong RBridge placement, resulting in ineffective data traffic distribution. The four RBridges located in the access layer choose two designated RBridges, and both of them are then forwarding the traffic via the same switch located in the distribution layer. This switch is then sends the aggregated traffic to the core layer via one link, which is overloaded.

Scenario 3. This scenario is suitable in the cases, where the highest data traffic is transferred inside each block (does not span over the core layer). The deployed blocks with TRILL use multiple links to carry the data, so the average link utilization is, therefore only 29%. The middle block use traditional STP and the most loaded link is utilized at 57%.

Scenario 4. The fourth scenario showed the advantage of TRILL deployment in the core and distribution layers. This is effective in the case, that traffic does span the core. Link utilization between RBridges lowers to 50%.

Scenario 5. This scenario represents the best option – complete deployment of the TRILL. In reality, this solution would be possible only if maximum performance is demanded and financial budget is not a concern. The most loaded links in this scenario are between the end devices and access switches.

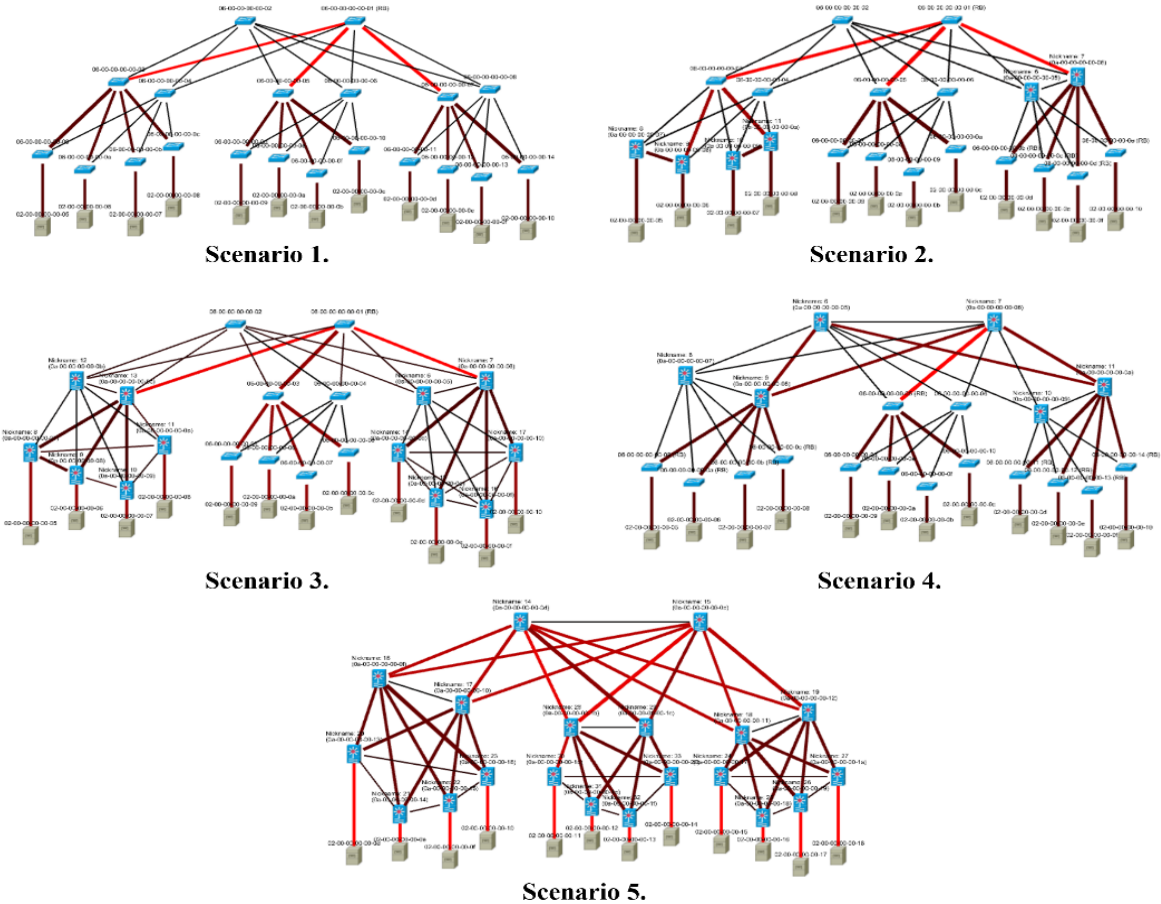

Figure 2: Deployment of the TRILL in enterprise network

V. CONCLUSION

Modern networks are transitioning from STP to more advanced bridging protocols like TRILL. This change will require tools allowing network administrators to explore how new protocol will influence network behavior. It is also necessary to consider possible deployments of the protocol to optimize network performance and minimize costs. We developed the first such modeling tool of TRILL protocol. The tool can be used to demonstrate TRILL's features, or as a supporting tool for TRILL deployment into an enterprise network. Interactive control of the application allows experimenting with traffic load or protocol behavior in the case of a link or device failure.

The application implements the most important TRILL features, including IS-IS behavior and simplified STP and Ethernet protocol. In our future work, we would like to extend the application to support additional features. Complex enterprise networks are often using VLANs, therefore we would like to implement this feature into the model together with option to show detailed statistics. Among the statistics, real time in micro-seconds could be displayed. We are also planning to translate the GUI of the application into English, as we are aware of the international interest of the program.

ACKNOWLEDGEMENT

This work and contribution is supported by the project of the student grant competition of the University of Pardubice, Faculty of Electrical Engineering and Informatics, Intelligent Smart Grid networks protection system, using software-defined networks, no. SGS_2016_016. We would like to thank to Ing. Tomas Kmonicek for his kind cooperation on creating this project.

REFERENCES

[1] Cisco, "Cisco Visual Networking Index: Forecast and Methodology". Retrieved from: http://goo.gl/aLP56a.,on: 15 November 2015.

[2] "IETF RFC 6325, Routing Bridges (RBridges): Base Protocol Specification"., 2011 . ISSN: 2070-1721.

[3] Selga J, Zaballos A, Navarro J, "Solutions to the Computer Networking Challenges of the Distribution Smart Grid". *In: IEEE Communications Letters*, vol 17, no 3, pp. 588-591, 2013.

[4] Lu, C., Jiang, Z., Yu, J., Zhang, G., Li, M., Liang, W. and Bi, J. "Understanding the overhead of large layer 2 data center networking: Measurement and analysis of TRILL as an example". *In Innovative Computing Technology (INTECH), 2013 Third International Conference*, pp. 555-560, 2013.

[5] Piccolo V, Amamou A, Dauchy W, Haddadou K. "Multi-tenant isolation in a TRILL based multi-campus network". *In Cloud Networking (CloudNet), 2015 IEEE 4th International Conference*, pp. 51-57, 2015.

[6] Ibanez G, Rojas, E. "All-path bridging: Path exploration as an efficient alternative to path computation in bridging standards". *In 2013 IEEE International Conference on Communications Workshops (ICC)*, pp. 1280-1285, 2013.

[7] Coudron, M., Secci, S., Maier, G., Pujolle, G. and Pattavina, A., "Boosting Cloud Communications through a Crosslayer Multipath Protocol Architecture". *In Future Networks and Services (SDN4FNS), 2013 IEEE SDN*, pp. 1-8, 2013.

[8] Lopes J, Sargento S, Zúquete A. "A Dependable Alternative to the Spanning Tree Protocol". *In Dependable Computing*, pp. 148-164, 2013.

[9] Gohar, M., Choi, S.I. and Koh, S.J., "Mobility support for Proxy Mobile IPv6 in TRILL-based mobile networks". *In 2015 Seventh International Conference on Ubiquitous and Future Networks*, pp. 677-681, 2015.

[10] Kazmi M, Sardar M, Khayam S. "Linux Implementation of TRILL Protocol". (2013). Retrieved from: http://wisnet.seecs.nust.edu.pk/projects/trill/index.html. on 21 November 2015.

[11] Balik L, Hornig O, Horalek J, Sobeslav V. " IED Remote Data Acquisition and Processing in Power Distribution". *In: Computational Collective Intelligence*, pp. 355-364, 2015.

[12] Behan M, Krejcar O. "Modern Smart Device-Based Concept of Sensoric Networks". *EURASIP Journal on Wireless Communications and Networking*, no. 1, pp. 1-13, 2013.

[13] Bernstein, G. "New Switch Architectures and the Impact to the 40/100GbE Transition in the Data Center". *In: The Data Center Journal*. 2013, Retrieved from http://goo.gl/XrQOCo. on 22 November 2015.