# A Graph Model of the Topology of Physical, Link and Network Layers of an Enterprise Network

Anton Andreev, Aleksandr Kolosov, Anatoly Voronin, Iurii Bogoiavlenskii
Petrozavodsk State University
Petrozavodsk, Russia
{andreev, akolosov}@cs.petrsu.ru, voronin@petrsu.ru, ybgv@cs.petrsu.ru

*Abstract*—**Many network management tasks require a network topology graph as its input. However, the lack of standard methods of network elements detection, coupled with the incompleteness and heterogeneity of the available topology data, complicate the network topology discovery process. In these conditions it is rational to separate the task of collecting data about the current topology of the network from the tasks of analyzing the collected data and building a topology graph. This approach requires a definition of the modern enterprise networks topology graphs family. Contribution of the paper is the graph model of the topology of physical, link and network layers of a modern enterprise Ethernet and IP based network. A set of theorems proved in the paper allows to infer parts of the topology graph, which are not described by data available on the network devices. Three-step process of building a topology graph using the network data and inference procedures is also proposed in the paper.**

## I. INTRODUCTION

The current trends of the information technologies market [1], including rapid growth in the number of users as well as the widespread adoption of the virtualization technologies, have led to significant increases in the size and complexity of the ICT-infrastructure of modern network service providers (NSP). Additionally, midsize providers operating their enterprise networks and possibly providing transit services now make up the largest category of all modern NSPs [2].

For NSPs, a key to providing high quality reliable service lies in the systematic approach to and the implementation of efficient network management methods, which, in turn, requires having a complete and detailed description of the network topology.

Moreover, for enterprises that maintain all three basic layers of their networks (physical, link, and network layers), it is very important to have complete and detailed descriptions of each layer's topology.

As an example, the virtual local area network (VLAN) technology, widely used in enterprise networks, significantly improves the flexibility of the organization of the network's physical layer by increasing the complexity of its link layer structure. This impacts a lot of network management tasks, including capacity planning, failure cause identification, and network documentation production.

A common method of visual representation of the network topology on a given level is a graph, in which vertices represent managed elements (network devices, ports, end points of data transmission protocols, etc.) and edges represent hierarchy and data exchange connections.

One of the main problems of the automation of the process of building a topology graph is the lack of standardization in the methods of detecting the elements of the network environment and connections between them.

This necessitates an analysis of a large number of heterogeneous data sources not designed for this specific task: address forwarding tables, outputs of STP protocols, LLDP, CDP and ARP protocols cache, IP protocol routing tables, and possibly others.

However, none of these protocols and technologies of data source analysis and representation, when used apart, can guarantee completeness and the relevance of the resulting information as it relates to building network topology graphs.

In practice, the use of the existing methods of data source analysis is further complicated by the lack or inaccessibility of standardized interfaces for accessing these sources. As a result, a most complete and accurate topology graph can be produced only by examining all available and accessible sources of data about the network topology.

In these conditions it is rational [3], [4] to break the task of building a topology graph into two stages: (1) collecting data on the current topology of the network from all available sources; and (2) analyzing the collected data with the purpose of building fragments of the graph. This approach leads to a need to build a model that would determine common characteristics of all modern network topologies — a model that would allow us to build graphs of various existing networks. For purposes of this paper, we will call this model a graph model of the network topology.

A generalized graph model of an enterprise network's physical, link, and network layers presented in this paper was developed with the goal of unifying descriptions of various algorithms for collecting data about the network topology, as well as formal justification of the methods for building parts of the network topology graphs, data about which cannot be received from the network devices.

The paper also describes a three-step process of building a network topology graph that is independent of specific data sources, and offers a formal rationale for the steps of the process.

### A. Related work

All existing algorithms of building network topology graphs are based on using a single data source and, as a result, reflect only a very narrow set of the network topology characteristics on a given layer. Consequently, the graph models produced by these algorithms are very limited in their application as generalized graph models of the network topology.

For instance, the network topology graphs described in [5], [6] are built based on the available address forwarding tables and therefore describe only the link layer of the network. Similarly, in the [7], the network is inspected only from the viewpoint of the Spanning Tree Protocol, running on the link layer. A possibility of presence in the network of VLANs built on the IEEE 802.1Q standard is partially considered in [4] and [8]; however, the model suggested in [4] does not describe the network layer topology, and model suggested in [8] does not differ logical connections from physical connections and therefore could not describe interfaces that have connections in several VLANs.

The model suggested by the algorithm [9] accounts for the presence in the network of routers and IP-subnets; however, a number of the assumptions and allowances made by the authors does not allow for the use of the model to describe complex modern networks.

### B. Organization of the paper

The rest of this paper is organized as follows. The next section describes the proposed graph model of an enterprise network topology and subsections describe physical, link and network layers of this model. Section III describes an abstract network topology graph building process and provides a formal proofs of its stages. Section IV concludes the paper.

## II. A GRAPH MODEL OF THE NETWORK TOPOLOGY

Topology of a network is determined by the topologies of its physical, link, and network layers, as well as by the hierarchical connections between elements of those topologies. The most common technologies of implementation of the said layers in enterprise networks are Ethernet and IP. Based on this, we will formulate the key requirements to the model we are developing.

The model must be able to describe basic elements of the physical layer of the network (devices and ports of network connections), indicate to which device various ports belong, and identify the characteristics of such relationships, including any physical connections over a transmission medium.

The model also has to account for the possibility of the presence in the network of aggregated links formed by port trunking in accordance with the IEEE 802.3ad standard. Interaction with the transmission medium via one or more ports is performed on the device's OS level, which, in turn, provides to its other components an abstraction of a point of access to the transmission medium, independent from the method of its implementation. This abstraction, called a physical interface, also needs to be reflected in the model.

The basic unit of the Ethernet-based network's link layer topology is a broadcast domain. The model should provide the ability to represent a broadcast domain as a set of ports that belong to different devices and that are capable of transmitting data frames to each other. Furthermore, if the VLAN standard IEEE 802.1Q is used, each port may be included in more than one broadcast domain, which are identified by their integer labels.

The model has to be able to describe any variation of the broadcast domain organization as well as reflect the specifics of network switches and bridges defined by the IEEE 802.1D standard.

The topology of the network layer that is based on the IP protocol is defined by breaking each broadcast domain's ports into their own IP-subnetworks. Each port may belong to more than one such subnetwork.

The model should be able to depict groupings of the ports by their respective subnetworks and also represent the data routing connections between the subnetworks. This model does not consider the case of complex routing policies, nor does it consider the application of the Network Address Translation.

As we move forward, all elements of the developed model will be presented using the sample network (Fig. 1), which consists of three IP-subnetworks containing one workstation each. Each subnetwork is implemented within a separate VLAN. The isolation of the broadcast domains by VLANs is done by two switches, to which the workstations are physically connected. The router interfaces providing connections between the subnetworks are also physically connected to the switches.

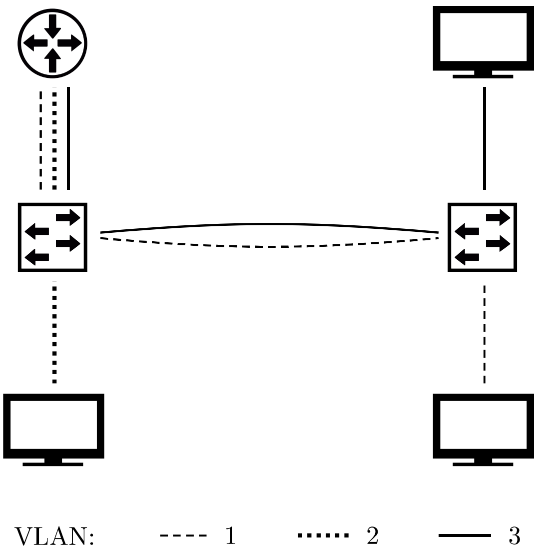

Fig. 1. The layout of the sample network

### A. Physical layer

Let us take a nonempty finite set of network devices $D$. A set of all ports of any device $d \in D$ we will call $P_d$ and a set of all of the ports of all of the devices — $P$. For any $d \in D$, a set $P_d$ is nonempty and finite. For any two $d_1, d_2 \in D$, $P_{d1} \cap P_{d2} = \varnothing$. Let us define the association relation between the ports and the devices $A^1$ in such a manner that $(p, d) \in A^1$

and $(d, p) \in A^1$ when and only when $p \in P$, $d \in D$, and $p \in P_d$. The relation $A^1$ is binary, symmetric and irreflexive.

On the given set $P$, let us take a physical connection relation $L^2$, which is binary, symmetric, transitive and irreflexive. Ports that are $L^1$-related to each other must be associated with different devices, i.e. $\forall(p_1, p_2) \in L^1$, there is no such $d \in D$ for that $p_1, p_2 \in P_d$. Let us interpret the relation $L^1$ as the following: the ports $p_1, p_2 \in L^1$ are connected by the same data transmission media.

A set of all physical interfaces of a device $d \in D$ is a partition of the ports of this device. We will call this division $I^1_{d1}$ and the set of all physical interfaces of all of the devices in the network we will call $I^1$.

The structure of the physical layer of a network can be described with a connected undirected graph $G^1 = \langle V^1, E^1 \rangle$, in which vertices represent the network's devices and ports ($V^1 = D \cup P$) and edges represent association relations between them as well as their connections at the physical layer ($E^1 = A^1 \cup L^1$). A graph of the physical layer of the sample network (that was introduced in the fig. 1) is presented in the Fig. 2. On this graph, squares represent devices and circles represent ports.

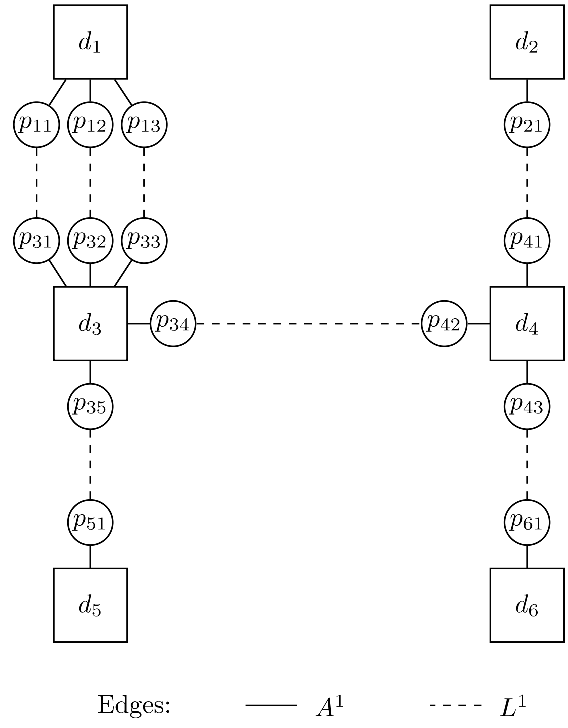

Fig. 2. Graph of the physical layer of the sample network

### B. Link layer

Each physical interface $pi \in I^1$ is assigned a finite set of labels $VID_{pi} \subset \mathbb{N}_0$, which correspond to the VLAN identifiers that were configured on that interface. In a case when a device associated with the ports from the set $pi$ does not support the VLAN technology, $VID_{pi} = \{0\}$. Additionally, each device $d \in D$ is assigned a set of labels ($VID_d$) that include labels of all of the device's ports.

Analogous to the physical layer, to deal with VLANs configured on each of its physical interface, the OS of each device $d \in D$ creates a separate link interface $(pi, v)$, where $pi \in I^1_d$, $v \in VID_{pi}$. Let a nonempty set of all link layers of all devices be called $I^2$. And let us define the association relation between link interfaces and devices as $A^2$, so that $(li, d) \in A^2$ and $(d, li) \in A^2$ are if and only if $li = (pi, v)$, $d \in D$ and for all $p \in pi$ accomplishes $(p, d) \in A^1$. The relation $A^2$ is binary, symmetric and irreflexive. And a set of all link interfaces associated with a certain device $d$ we will call $I^2_d$.

On the set $I^2$, let us define the link layer connection relation $L^2$ so that two not blocked link interfaces $li_1 = (pi_1, v_1) \in I^2_{d1}$, $li_2 = (pi_2, v_2) \in I^2_{d2}$, associated with different devices, are $L^2$-related, if $p_1 \in pi_1$, $p_2 \in pi_2$ exist, where $(p_1, p_2) \in L^1$. The relation $L^2$ is binary, symmetric, transitive and irreflexive. Let us interpret the relation $L^2$ as the following: two devices can communicate at the link layer via not blocked (e.g. as a result of the work of the STP protocol) link interfaces that are connected at the physical layer.

Certain devices of the network (e.g. switches and bridges) can be configured to forward transit data frames between their link layer interfaces. On the set $I^2$, let us look at the commutation relation $F^2$, which is binary, symmetric, transitive and irreflexive. The interfaces that are $F^2$-related to each other must be associated with the same device, i.e. $\forall(li_1, li_2) \in F^2$, exists $d \in D$ such that $li_1, li^2 \in I^2_d$. Let us interpret the relation $F^2$ as the following: the configuration of a device provides for a possibility to forward transit data frames between two different link layer interfaces $(li_1, li_2) \in F^2$.

The structure of the link layer of a network can be described with a connected undirected graph $G^2 = \langle V^2, E^2 \rangle$, in which vertices represent the network devices and link interfaces ($V^2 = D \cup I^2$) and edges represent association relations between the devices and the interfaces, commutation relations and connections at the link layer ($E^2 = A^2 \cup F^2 \cup L^2$). A graph of the link layer of the sample network is presented in the Fig. 3. On this graph, vertices representing link interfaces are depicted as ellipses with ports and VLAN identifiers (refer to Fig. 2) noted inside. Devices are depicted as squares.

Let us take a look at a graph $\widehat{G}^2 = \langle I^2, F^2 \cup L^2 \rangle$ that is a subgraph of $G^2$. The presence of a path between the link layer interfaces in the graph $G^2$ corresponds to the possibility of their communication at the link layer either directly or via a chain of switching devices. As such, the sets of vertices of connected components of the graph $G^2$ turn out to be the broadcast domains of the network. Partition of the set $I^2$, each element of which represents one broadcast domain, will be called $BD$.

For an edge-simple path in the graph $\widehat{G}^2$ that does not have two consecutive commutation edges, we will use the term *link layer path*. According to the IEEE 801.1D standard, in the graph $G^2$, there cannot be more than one link layer path between any two link layer interfaces.

**Property 1.** In the graph $G^2$, between two link layer interfaces, a link layer path exists if and only if the interfaces belong to the same broadcast domain.
**Proof.** Trivial, derived from the definitions of the broadcast domain and the link layer path. ■

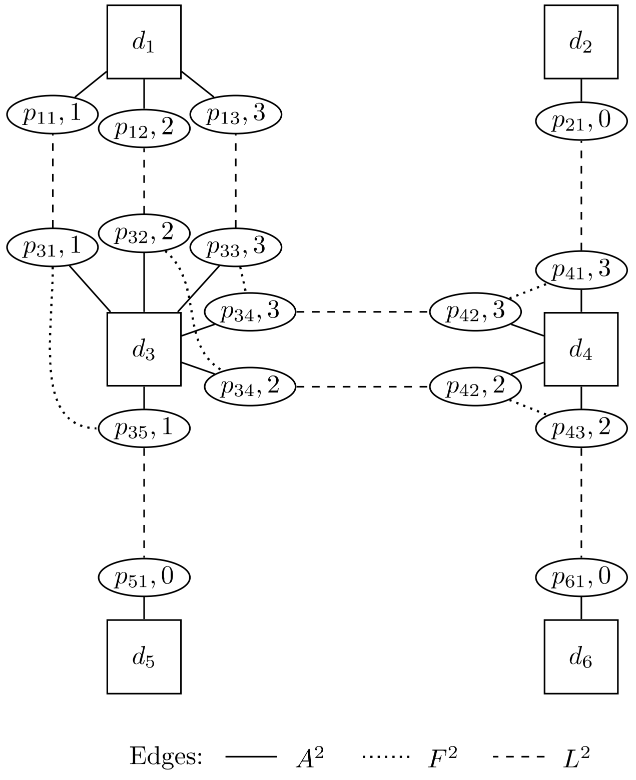

Fig. 3. Graph of the link layer of the sample network

### C. Network layer

Let us introduce a set of identifiers of all the network's subnets $N$. For each subnet with an identifier $n \in N$, there is a determined finite set of all possible identifiers of the subnet's hosts — $H_n$. To transmit and receive data packets at the network layer via a single or multiple link interfaces, the OS of each device $d \in D$ creates a network interface $(S, n, h)$, where $S \subset I_d^2$, $n \in N$, $h \in H_n$. Let us call a nonempty set of all of the network interfaces of all network's devices $I^3$. Let us define the association relation between network interfaces and devices $A^3$ so that $(ni, d) \in A^3$ and $(d, ni) \in A^3$ if and only if $ni = (S, n, h) \in I^3$, $d \in D$ and for all $li \in S$, $(li, d) \in A^2$ is accomplished. The relation $A^3$ is binary, symmetric and irreflexive. The set of all network interfaces associated with any device $d$ will be called $I_d^3$.

Two devices can communicate at the network layer via network interfaces that belong to the same broadcast domain and have the same subnet identifier. On the set $I^3$, let us define the the network layer connection relation $L^3$ so that two network interfaces $ni_1 = (S_1, n_1, h_1) \in I^3$ and $ni_2 = (S_2, n_2, h_2) \in I^3$, associated with different devices, are $L^3$-related to each other if $n_1 = n_2$ and exist $dom \in BD$, $li_1 \in S_1$, $li_2 \in S^2$ so that $li_1, li_2 \in dom$. The relation $L^3$ is binary, symmetric, transitive and irreflexive.

In each subnet there may be devices (routers) that can forward transit datagrams between all connected subnets. Analogous to the link layer, on the set $I^3$, let us look at the routing relation $F^3$ that is binary, symmetric, transitive and irreflexive. The interfaces that are $F^3$-related to each other must be associated with the same device, i.e. $\forall (ni_1, ni_2) \in F^3$, exists $d \in D$ such that $ni_1, ni_2 \in I_d^3$. Let us interpret the relation $F^3$ as the following: the configuration of a device $d \in D$ provides for a possibility of datagram transmission between two network layer interfaces $(ni_1, ni_2) \in F^3$.

The structure of the network layer of a network can be described with a connected undirected graph $G^3 = \langle V^3, E^3 \rangle$, in which vertices represent the network devices and network layer interfaces ($V^3 = D \cup I^3$) and edges represent association relations between the devices and the interfaces, routing relations and connections at the network layer ($E^3 = A^3 \cup L^3 \cup F^3$). A graph of the network layer of the sample network is presented in the Fig. 4. On this graph, vertices representing network interfaces are depicted as rectangles with link interfaces, subnet identifiers and host identifiers defining each interface noted inside. There $li_{11} = (p_{11}, 1)$, $li_{12} = (p_{12}, 2)$, $li_{13} = (p_{13}, 3)$, $li_{21} = (p_{21}, 0)$, $li_{51} = (p_{51}, 0)$, $li_{61} = (p_{61}, 0)$. Devices are depicted as squares.

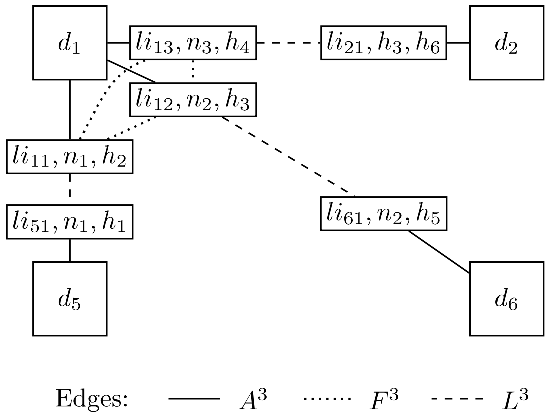

Fig. 4. Graph of the network layer of the sample network

Let us take a look at a graph $\widehat{G}^3 = \langle I^3, L^3 \rangle$ that is a subgraph of $G^3$. Presence of a path between the network interfaces in the graph $\widehat{G}^3$ corresponds to the possibility of their direct communication at the network layer. As such, the sets of vertices of connected components of the graph $\widehat{G}^3$ turn out to be subnets of the network. Partition of the set $I^3$, each element of which represents one subnet, will be called $SN$.

For an edge-simple path in the subgraph $\langle I^3, L^3 \cup F^3 \rangle$ of the graph $G^3$ that does not have two consecutive routing or connection edges at the network layer, we will use the term *network layer path*.

**Property 2.** A network layer path between two network interfaces contains a routing edge if and only if both interfaces belong to different subnets.
**Proof.** Trivial, derived from definitions of a broadcast domain and a network layer path. ■

The structure of any given network can be described with a connected undirected graph $G = \langle V, E \rangle$ — the network topology graph — in which the set of vertices is $V = V^1 \cup V^2 \cup V^3 = D \cup P \cup I^2 \cup I^3$ and the set of edges is $E = E^1 \cup E^2 \cup E^3 = A^1 \cup L^1 \cup A^2 \cup F^2 \cup L^2 \cup A^3 \cup F^3 \cup L^3$. The network topology graph does not contain loops and multiple edges, it may, however, contain cycles.

## III. BUILDING A NETWORK TOPOLOGY GRAPH

The process of building a network topology graph based on the data set acquired from querying the network devices (e.g. with the SNMP protocol) consists of three steps. The first step is building graph fragments that describe the devices (with their ports and interfaces), existence of which follows directly from the results of the input data analysis.

Assuming the completeness of the input data, the graph that we get as a result of the first step of the process describes all devices within the network. However, as has already been mentioned, this is an unrealistic condition for real life networks. As such, inference of the devices, data about which are missing from the network input data, is done at the second step of the graph building process, based on the analysis of the reachability sets of the network's link layer interfaces [5]. For example, graph fragments that correspond to unmanageable or uncooperative hubs and switches are discovered during the second step.

The goal of the third step is to build link layer connection edges, based on the analysis of the reachability sets, as well as physical and network layer connection edges, based on the definitions of the model.

Let us now take a closer look at the methods of building and analyzing reachability sets of the network's link layer interfaces that were used on the second and third steps of our graph building process.

### *A. Methods of building reachability sets of the link layer interfaces*

Within the network, one link layer interface is reachable from another link layer interface if the first interface is an endpoint for data transmission at the link layer for the second interface. Having data about reachability between interfaces allows for inference of the devices, data about which are missing from the input data used in the graph building process, as well as the link layer connection edges.

Let us examine reachability within a network topology graph $G$. We will use the term *reachability path* for a link layer path in which the first and the last edges are not commutation edges. On the set $I^2$, we will introduce a reachability relation $\leftrightarrow$ so that $li_1 \leftrightarrow li_2$ if from $li_1$ to $li_2$ in the graph $G$ exists a reachability path. In this case we will say that $li_2$ is reachable from $li_1$. The relation $\leftrightarrow$ is binary, symmetric and irreflexive. In the graph of the link layer of the sample network presented in the Fig. 3, in the broadcast domain formed by a VLAN with an identifier 1, the following pairs of interfaces reachable from each other: $(\{p_{11}\}, 1)$ and $(\{p_{31}\}, 1)$, $(\{p_{35}\}, 1)$ and $(\{p_{51}\}, 0)$, $(\{p_{11}\}, 1)$ and $(\{p_{51}\}, 0)$.

We will say that a port $p \in P$ is reachable from an interface $li_1 \in I^2$ if exists such an interface $(pi, v) \in I^2$ reachable from $li_1$ for which $p \in pi$.

For each $li \in I^2$, we will introduce a set $RS_{li} \subset I^2$ that includes all interfaces reachable from $li$. The set of interfaces reachable from the interfaces that are $F^2$-related to the $li$, we will define as $CRS_{li}$. In the graph presented in the Fig. 3, the set $RS$ of the interface $(\{p_{34}\}, 3)$ equals to $\{(\{p_{42}\}, 3), (\{p_{21}\}, 0)\}$ and the set $CRS$ equals to $\{(\{p_{13}\}, 3)\}$.

Reachability sets can be built using the input data of the network topology graph building process. Most data sources provide information about reachability of the network's ports from the network's interfaces. Following the definition of reachability and the Property 1, in order for a port $p$ to be reachable from an interface $li_1$, it is necessary that the interface $li_1 = (pi_1, v_1)$ and some $li_2 = (pi_2, v_2)$ (where $p \in pi_2$) belong to the same broadcast domain. If there are no data as to whether the reachable port belongs to a VLAN, we will assume that either $v_1 = v_2$ (if it is known that the device supports VLAN) or $v_2 = 0$.

The accuracy of inferring the network's elements through the reachability set analysis directly depends on the completeness of the reachability data [5], [10]. And the common problem of the network's topology data incompleteness and inaccessibility leads to a need to find indirect data about reachability, which is possible with the following proposition.

**Proposition 1.** If a link interface $li_2$ is reachable from an interface $li_1$, then all interfaces reachable from interfaces commutating with $li_2$ are reachable from $li_1$, i.e. $CRS_{li2} \subset RS_{li1}$.
**Proof.** Let $li_1 \in I^2_{d1}$, $li_2, li_3 \in I^2_{d2}$, where $d_1, d_2 \in D$. Let also $(li_2, li_3) \in F^2$, $li_1 \leftrightarrow li_2$ and exists such a link interface $li_4 \in I^2_{d3}$ (where $d_3 \in D$) for that $li_3 \leftrightarrow li_4$. Following to the definition of reachability there are paths between interfaces $(li_1, \ldots, li_2)$ and $(li_3, \ldots, li_4)$. But since $(li_2, li_3) \in F^2$ then graph $G$ contains path $(li_1, \ldots, li_2, li_3, \ldots, li_4)$ that satisfies definition of reachability. So, $li_4 \in RS_{li1}$. ■

Reachability sets are built at the beginning of the second step of the graph building process, after which they are supplemented based on the symmetry of the reachability relation and the Proposition 1.

### *B. Methods of inference of the network devices*

We will call a device, two or more interfaces of which are in the link layer connection relation with interfaces of other devices, an intermediate for these devices. Presence of an intermediate device may be inferred from the analysis of reachability sets with the following proposition.

**Proposition 2.** If two reachable from each other interfaces $li_1$ and $li_2$ have shared reachable interfaces, then those shared interfaces are not associated with the devices whose interfaces are located on the reachability path between $li_1$ and $li_2$.
**Proof.** Let us take a look at link interfaces $li_1, li_2, li_3 \in I^2$. Let $li_3$ be reachable from $li_1$ and $li_2$, and also $li_1 \leftrightarrow li_2$, i.e. there is a unique reachability path $w = (li_1, \ldots, li_2)$. Let us suppose that $w = (li_1, \ldots, li_3, \ldots, li_2)$. Let us denote the neighbor for $li_3$ interfaces that present in the path $w$ before and after $li_3$ as $li_4$ and $li_5$ respectively (here $li_4$ may coincide with $li_1$, $li_5$ — with $li_2$). Following to the definition of reachability, the interface $li_3$ must be either $L^2$-related or $F^2$-related to interfaces $li_4$ and $li_5$. Following to the definition of the relation $L^2$, interface $li_3$ may be in this relation with only one other interface, thus following to the definition of reachability $li_3$ could not be reachable from both $li_1$ and $li_2$ at the same time, i.e. $li_3$ does not located on the path $w$. In addition $li_3$ could not be $F^2$-related to $li_4$ nor to $li_5$ (and thereafter be associated with the same devices as $li_4$ and $li_5$) since in such case the path from $li_1$ to $li_2$ or $li_3$ will end with a commutation edge

that contradicts the condition of reachability $li_3$ from $li_1$ and $li_2$. ■

Whether an interface belongs to a transitional device can be determined with the following proposition.

**Proposition 3.** If for two reachable from each other link interfaces $li_1$ and $li_2$, the following conditions are met: (1) exists such an interface $li_3$ reachable from one of the two interfaces and unreachable from the other one; (2) $li_3$ is not reachable from the interfaces that commutate with $li_1$ or $li_2$; then $li_3$ is located on the reachability path between $li_1$ and $li_2$.

**Proof.** Let $li_1 \leftrightarrow li_2$ and $li_3 \leftrightarrow li_1$ but $li_3$ is not reachable from $li_2$ and interfaces are $F^2$-related to $li_2$. Following to the Property 1 interfaces $li_1, li_2, li_3$ are in the same broadcast domain $dom \in BD$. There is a link layer path $w_1 = (li_1, \ldots, li_3)$ that has no commutation edges on its ends because $li_1 \leftrightarrow li_3$. The link layer path $w_2 = (li_3, \ldots, li_2)$ exists following to the Property 1. Therefore following to the definition of a link layer path and the requirement of uniqueness of such a path there is either a link layer path $w_3 = (li_1, \ldots, li_3, \ldots, li_2)$, or a link layer path $w_4 = (li_1, \ldots, li_2, \ldots, li_3)$. The path $w_4$ could not be a link layer path since it ends with $L^2$ edge and following to the definition of reachability either $li_1 \leftrightarrow li_2$, or $li_3 \in CRS_{li2}$ that contradicts the conditions of the proposition. Let suppose that the path $w_3$ is not a reachability path. Then, since $li_1 \leftrightarrow li_2$, there is a link layer path $w_5 = (li_1, \ldots, li_2) \neq w_3$ that contradicts the requirements of link layer path uniqueness. In other words, $li_3$ is placed on a reachability path between $li_1$ and $li_2$. Similarly the proposition could be proved for interfaces that is reachable from $li_2$ but not from $li_1$. ■

The next proposition allows to determine whether there is a commutation edge between interfaces of a transitional device.

**Proposition 4.** If for a device $d \in D$ and two reachable from each other link layer interfaces $li_1$ and $li_2$ the following conditions are met: (1) $li_3 \in I^2_d$ is reachable from $li_1$, $li_4 \in I^2_d$ is reachable from $li_2$; (2) the interfaces from $I^2_d$, different from $li_3$ and $li_4$, are not reachable from $li_1$ and $li_2$ respectively; (3) the interfaces associated with $d \in D$ are not reachable from the interfaces commutating with $li_1$ or $li_2$; then $li_3$ and $li_4$ are in $F^2$-related.

**Proof.** Following to Proposition 3 interfaces $li_3$ and $li_4$ are placed in the reachability path $w = (li_1, \ldots, li_3, \ldots, li_4, \ldots, li_2)$. Subpaths $w_1 = (li_1, \ldots, li_3)$ and $w_1 = (li_4, \ldots, li_2)$ of the path $w$ has $L^2$-edges on both ends (according to the definition of reachability). Since $li_3$ and $li_4$ are associated with the same device $d$ they could not be $L^2$-related with each other. Let us suppose that they are not $F^2$-related. Then the subpath $w_3 = (li_3, \ldots, li_4)$ of the path $w$ must pass through one more interface $li_5 \in I^2_d$ because $li_3$ and $li_4$ already appear in edges $L^2$ in subpaths $w_2$ and $w_3$ and they could be $F^2$-related only to other interfaces from $I^2_d$. To be specific, we assume that $(li_3, li_5) \in F^2$. Then following to the definition of a link layer path the subpath $w_4 = (li_5, \ldots, li_2)$ of the path $w$ could not start with commutation edge. But in such case the path $w_4$ satisfies the definition of a reachability path and $li_5 \leftrightarrow li_2$ that contradicts the conditions of the proposition. Thereby $(li_3, li_4) \in F^2$. ■

Building transitional devices inferred through the analysis of reachability sets is done at the second step of the process of building a network topology graph.

### C. Methods of building connection edges

Let us introduce a set of criteria for the presence of a connection edge between pairs of interfaces.

The first theorem offers a criterion for the presence of connections with border devices, which can be hosts or routers.

**Theorem 1.** If a link interface $li_2$ is the only reachable interface from a link interface $li_1$, then $li_1$ and $li_2$ are $L^2$-related.

**Proof.** Since $li_1 \leftrightarrow li_2$ then there is a reachability path $w = (li_1, \ldots, li_2)$ in the graph $G$. According to the definition, the path $w$ could not include other interfaces of devices $d_1$ and $d_2$. Let us then assume that $w$ includes interface $li_3$ of some device $d_3$. If $w = (li_1, li_3, \ldots, li_2)$ then, according to the definition of a path of reachability, $(li_1, li_3) \in L^2$ and then $li_1 \leftrightarrow li_3$, which contradicts the conditions of the theorem. If $w = (li_1, \ldots, li_3, li_2)$ then, following the definition of a reachability path, there must exist an interface $li_4 \in I^2_{d2}$ such that $(li_4, li_3) \in F^2$ and $li_4$ is placed in the path $w$ before $li_3$: $w = (li_1, \ldots, li_4, li_3, li_2)$. But then, according to the definition of reachability, $li_1 \leftrightarrow li_4$ that also contradicts the conditions of the theorem. Thereby $w$ could not include other interfaces of devices $d_1$ and $d_2$ and interfaces of other devices. So then $w = (li_1, li_2)$ and then $(li_1, li_2) \in L^2$. ■

The second theorem offers a criterion for the presence of a connection between interfaces of different devices.

**Theorem 2.** Interfaces $li_1$ and $li_2$ are $L^2$-related if and only if $RS_{li1} = CRS_{li2} \cup \{li_2\}$ and $RS_{li2} = CRS_{li1} \cup \{li_1\}$.

**Proof.** Let us assume that interfaces $li_1, li_2 \in I^2$ are $L^2$-related. Then, according to the definition of reachability, $li_2 \in RS_{li_1}$, and according to the Proposition 1, $CRS_{li2} \subset RS_{li1}$. Following the definition of reachability and the condition of uniqueness of a link layer path, $RS_{li1}$ could not include other link interfaces since any path to them would pass through $li_2$ and the interface commutating with it. Thereby $RS_{li1} = CRS_{li2} \cup \{li_2\}$. Analogously, is holds true for $RS_{li2}$. Inversely, let us assume that $RS_{li1} = CRS_{li2} \cup \{li_2\}$. Let us also assume that $li_1$ is $L^2$-related not to $li_2$ but to $li_3 \in CRS_{li2}$. Then, according to the definition of reachability there is a link layer path $(li_1, \ldots, li_2)$ that does not end with edge $F^3$ and also a path $(li_3, \ldots, li_2)$ which ends with an edge of commutation. Since $(li_1, li_3) \in L^2$ path $(li_1, li_3, \ldots, li_2)$ is also a link layer path which is impossible according to the restriction of link layer path uniqueness. This theorem could be proved similarly for $RS_{li2}$. ■

As a result, based on the described theorems and model definitions, we can define the following criteria of presence of connection edges between interfaces.

**Criterion 1.** Following the Theorem 1, if from an interface $li_1 \in I^2$ only one interface $li_2 \in I^2$ is reachable, then $(li_1, li_2) \in L^2$.

**Criterion 2.** Following the Theorem 2, if two interfaces $li_1$ and $li_2$ are reachable from each other, and $RS_{li1} = CRS_{li2} \cup \{li_2\}$, then $(li_1, li_2) \in L^2$.

**Criterion 3.** Based on the definition of the relation $L^2$, if two link interfaces $li_1 = (pi_1, v_1)$ and $li_2 = (pi_2, v_2)$ are $L^2$-related, then for each $p_1 \in pi_1$ exists such $p_2 \in pi_2$, so that $(p_1, p_2) \in L^1$ and vice versa.

**Criterion 4.** Based on the definition of the relation $L^2$, if two ports $p_1$ and $p_2$ are $L^1$-related, then exist such VLAN identifier $v \in VID_{p1} \cap VID_{p2}$ and physical interfaces $pi_1$ and $pi_2$ so that $p_1 \in pi_1$ and $p_2 \in pi_2$, and link interfaces $li_1 = (pi_1, v)$, $li_2 = (pi_2, v)$ that are not blocked, then $(li_1, li_2) \in L^2$.

**Criterion 5.** Based on the definition of the relation $L^3$, if for two link interfaces $ni_1 = (S_1, n, h_1)$ and $ni_2 = (S_2, n, h_2)$ exist interfaces $li_1 \in S_1$ and $li_2 \in S_2$ that belong to one broadcast domain, then $(ni_1, ni_2) \in L^3$.

Building graph edges using the listed criteria is done at the third step of the network topology graph building process.

## IV. CONCLUSION

Modern approach to building a complete and detailed topology graph of an enterprise network amounts to the consecutive application of a large number of independent methods of collecting and analyzing data from heterogeneous sources about network devices and connections between them. The combined implementation of all available algorithms with a goal of building a maximally accurate and detailed network topology graph requires the development of a generalized model of a network topology that could provide a universal method of describing topology graphs of various certain real-life networks.

This work offers a graph model of the physical, link, and network layers of an enterprise network built on the base of Ethernet and IP protocols. The model description contains definitions of graphs of each of the network's layers as well as their properties. Additionally, within the model, a set of criteria has been developed to build the elements of the topology graph, data about which is missing from the available data sources. Finally, the paper describes an independent from specific data sources process of building a network topology graph, as well as its formal justification with the suggested model.

In the future we plan to expand the current model by adding the ability to describe virtual private network tunnels and virtual machines and network equipment, as well as potentially consider various characteristics and features of wireless network technologies.

## ACKNOWLEDGMENTS

The authors thank the reviewers, whose comments helped to improve this paper. Special thanks to Sofia and Shawn Gilman for assistance in preparing the text of the paper.

## REFERENCES

[1] "ICT Market Report 2015/16. Executive Summary". EITO in collaboration with IDC, July 2015. — Web: http://www.eito.com/Executive-Summary-ICT-Market-Report-2015/16 (accessed 14 september 2016)

[2] Y.-W.E. Sung et al, "Towards systematic design of enterprise networks", *Networking, IEEE/ACM Transactions on*. IEEE, 2011. vol. 19, no. 3. pp. 695–708.

[3] Vad. V. Voevodin, K. S. Stefanov, "Automatic detection and description of supercomputer network infrastructure", *Numerical methods and programming*. 2014. vol. 15, no. 4. pp. 560–568. (in Russian)

[4] L. Zichao et al, "Ethernet topology discovery for virtual local area networks with incomplete information", *Network infrastructure and digital content (ic-nidc), 2014 4th IEEE international conference on*. 2014. pp. 252–256.

[5] H. Gobjuka, Y. Breitbart, "Ethernet topology discovery for networks with incomplete information", *Networking, IEEE/ACM Transactionson*. 2010. vol. 18, no. 4. pp. 1220–1233.

[6] Y. Sun, Z. Shi, Z. Wu, "A discovery algorithm for physical topology in switched ethernets", *Local computer networks, 2005. 30th anniversary. the IEEE conference on*. IEEE, 2005. pp. 311–317.

[7] M.-H. Son et al, "Physical topology discovery for metro ethernet networks", *ETRI Journal*. 2005. vol. 4, no. 27. pp. 355–366.

[8] Xiaobo Ma, Tingting Yu, "An Algorithm of Physical Network Topology Discovery in Multi-VLANs", *TELKOMNIKA*. 2016. vol. 14, no.3A. pp. 375–379.

[9] Y. Bejerano, "Taking the skeletons out of the closets: A simple and efficient topology discovery scheme for large ethernet lans", *Networking, IEEE/ACM Transactions on*. 2009. vol. 17, no. 5. pp. 1385–1398.

[10] Y. Breitbart, H. Gobjuka, "Characterization of layer-2 unique topolo -gies", *Information Processing Letters*. 2008. vol. 105, no. 2. pp. 52–57.