# Network Topology Discovery: a Problem of Incomplete Data Improvement

Anton Andreev, Anton Shabaev, Iurii Bogoiavlenskii
Petrozavodsk State University
Petrozavodsk, Russia
andreev@cs.petrsu.ru, ashabaev@petrsu.ru, ybgv@cs.petrsu.ru

***Abstract*—The logical and physical topology description of an enterprise computer network is required for many network management tasks. However, the automatized building of such description is complicated due to the connectivity data incompleteness. This leads to the necessity of solving the problem of incomplete data improvement. In this paper we used previous works on the network modeling and commonly used reachability set abstraction to solve the denoted problem. The contribution of this paper is a criterion for determining when the connectivity data is sufficient for building topology descriptions and the algorithm for incomplete data improvement based on the criteria. The algorithm implementation test and evaluations provided in the paper show its correctness, applicability for using in the real enterprise networks and its greater effectiveness comparing to the other existing methods.**

## I. Introduction

Network topology information — a knowledge of network devices and their physical and logical interconnections — is a key to a range of complex network management tasks such as performance analysis and topology scaling [1], ensuring the network topology reliability and reducing the number of internal connection [2], etc.

Modern computer networks are huge and dynamic. This makes building and maintaining of network topology description almost impossible to do manually. Therefore the problem of the topology discovery process automation arises [3], [4], [5], [6], [7], [8], [9], [10]. One of the most significant issues on the way of solving this problem is a possible link layer structure data incompleteness [4], [5], [6], [7], [11]. The incompleteness is caused by the aging mechanics in the data sources (like ARP-cache and Address Forwarding Tables), network equipment and software heterogeneity, and possible data unavailability due to the security and technical reasons. It is known that data incompleteness may cause errors and inaccuracy in the automatically built topology description [5], [4], however, it is possible to infer a part of missing information to make topology discovery process more accurate [4], [5], [13].

To solve the problem of network topology discovery automation, the authors previously have developed a generalized graph model of an enterprise network's physical, link, and network layer topology [12] and the algorithm for building an enterprise network topology using widespread data sources [13]. In these works, basic steps to cope with the data incompleteness were provided: unified use of a wide range of data sources and inferring of the missing data using data properties.

The goal of this paper is a further formal study of the incomplete data improvement problem as a part of the network topology discovery process, as well as the development and evaluation of an algorithm for solving this problem.

The rest of this paper is organized as follows. The next section provides a short description of the related work. Section III shortly describes the graph model of an enterprise network topology from [12]. Section IV provides a formal description of the incomplete data improvement problem. Section V describes the proposed algorithm for solving the problem, as well as its analysis and evaluation. Section VI provides the algorithm implementation testing results and its comparison to schemes from section II. Section VII concludes the paper.

## II. Related work

The problem of data incompleteness has been thoroughly studied as a subproblem of network topology discovery. The problem mostly impacts the link layer connections data in networks usually taken from Address Forwarding Tables (AFT). AFTs provide data not on the direct (physical) connections in the network but on the link layer interfaces *reachability*, i. e. if an interface can send link layer data frames to the other interface.

In [3] researchers stated the need for deep AFT data study due to the possibility of its incompleteness. They provided a model and a set of rules for dealing with incomplete data and building a network topology. Authors have practically proven that their method is able to build an accurate network topology even with 15%-complete AFTs. However, their method is only applicable to networks with tree-like structure and without VLANs. Also, possible conflicting situations during connection discovery may cause mistakes in the built topology.

Researchers in [4] described common properties of incomplete data on network structure. They proposed an algorithm for dealing with incomplete data and a theoretically-proven requirement for such data to be sufficient for topology discovery. However, this method may fail if the network contains transparent (hubs with no MAC address) or uncooperative (SNMP-unavailable) devices. Also the method is not applicable for networks with VLANs.

In [5] researchers have proven that the topology discovery for networks with incomplete data, as well as the deciding whether available data allows to find unique topology are both NP-hard problems in the number of network nodes. The researchers proposed a common model for the incomplete data and a set of heuristic rules for the data improvement. One of their algorithms for topology discovery has been practically proven to work with data that is 50% complete. However, they did not provide any theoretical study on the efficiency of their data improvement process. Also, two of three of their algorithms for topology discovery may work only in networks with specific restrictions on the network topology.

In [6], [7], [11] the researchers try to adapt the results form [5] for networks with VLANs. In [7] authors argue it is important to use more than one data source for the topology discovery and add ARP-cache to AFT. The using of ARP-cache may additionally extend possible incomplete data in AFT. Similarly, authors in [10] add Spanning Tree Protocol data to the AFT data which provide more knowledge on the direct connections. In [11] researchers relax the requirements of methods in [7] to require only data on the reachability between service network equipment (excluding hosts). However, none of these works provide a theoretical or practical study on the data incompleteness influence or any criteria for their methods to work.

## III. A GRAPH MODEL OF THE NETWORK TOPOLOGY

Let us provide a short description of the model from [12] for the purposes of using its elements further. The referenced article contains a description of the physical, link and network layers model, but for the purposes of our work here only the link layer and selected elements of the physical layer are needed.

### *A. Description of entities and relations*

Let us take a nonempty finite set of network devices $D$. A set of all ports of any device $d \in D$ we will denote $P_d$ and a set of all ports of all devices — $P$.

On the given set $P$, let us take a symmetric physical connection relation $L^{(1)}$. Two ports $p_1, p_2 \in P$ associated with different devices are $L^{(1)}$-related if they are connected by the same data transmission media. Here and further the upper index denotes the layer.

Let us define a finite set of labels $VID \subset \mathbb{N}_0$, which correspond to the VLAN identifiers that are used in the network. For the purposes of link layer data transmission using one or more (in case of link aggregation) physical ports each device $d \in D$ creates a link layer interface $(u, v)$ where $u \subset P_d$, $v \in VID$. A set of all link interfaces is noted as $I^{(2)}$. In a case when a device associated with the ports from the set $u$ does not support the VLAN technology, $v = 0$.

Let us define the symmetric association relation between link interfaces and devices as $A^{(2)}$, so that $(r, d) \in A^{(2)}$ if and only if $r = (u, v)$, $d \in D$ and $u \subset P_d$. A set of all link interfaces associated with a certain device $d$ we denote $I_d^{(2)}$.

Two devices can communicate at the link layer via not blocked (e. g. with STP) link interfaces, ports of which are connected on a physical layer. On the set $I^{(2)}$, let us define the symmetric link layer connection relation $L^{(2)}$ so that two not blocked link interfaces $r_1 = (u_1, v_1) \in I_{d1}^{(2)}$, $r_2 = (u_2, v_2) \in I_{d2}^{(2)}$, associated with different devices, are $L^{(2)}$-related if $p_1 \in u_1$, $p_2 \in u_2$ exist, where $(p_1, p_2) \in L^{(1)}$ and $r_1$, $r_2$ can send a link layer data frame to each other.

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 and be different. Relation $F^{(2)}$ can be interpreted as follows: the configuration of a device provides for a possibility to forward transit data frames between two link layer interfaces $(r_1, r_2) \in F^{(2)}$.

The structure of the link layer of any given network can be described with a connected undirected graph $G = \langle V, E \rangle$ — the link layer topology graph — in which the set of vertices is $V = D \cup I^{(2)}$ and the set of edges is $E = A^{(2)} \cup F^{(2)} \cup L^{(2)}$. The link layer topology graph does not contain loops and multiple edges, but it may contain cycles. This graph is a subgraph of the larger network topology graph described in [12].

To illustrate the elements of the model let's take a look at a sample network (Fig. 1), which consists of two separate VLAN containing one workstation each. The isolation of the broadcast domains by VLANs is done by two switches, to which the workstations are physically connected. The router is at the upper right of the figure.

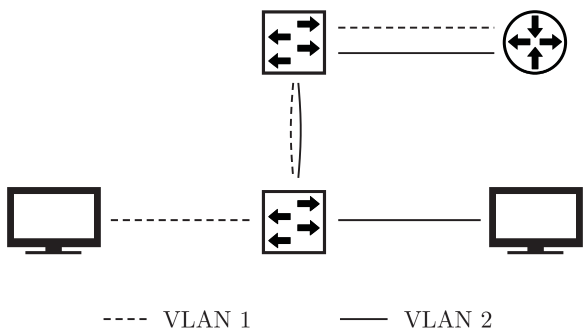


Fig. 1. The layout of the sample network

A link layer topology graph of the sample network is presented in the Fig. 2. Devices are depicted as squares. Vertices representing link interfaces are depicted as ellipses with ports and VLAN identifiers noted inside. Edge notation is presented in the picture.

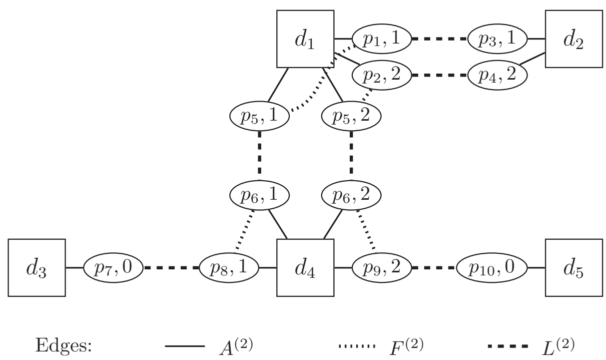


Fig. 2. Link layer topology graph of the sample network

Let us take a look at a graph $\widehat{G} = \langle I^{(2)}, F^{(2)} \cup L^{(2)} \rangle$. The presence of a path between the link layer interfaces in this graph 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 $\widehat{G}$ 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 denoted $BD$.

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

**Property 1.** *If in graph G there is a link layer path between any two interfaces then this path is the only existing link layer path between these two interfaces.*

From the definition of a link layer path comes the following property.

**Property 2.** *In the link layer graph, between two link layer interfaces, a link layer path exists if and only if the interfaces belong to the same broadcast domain.*

### B. Link layer reachability sets

Within the network, one link layer interface is reachable from another link layer interface if the first interface is a possible endpoint for data transmission at the link layer for the second interface.

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 $r_1 \leftrightarrow r_2$ if from $r_1$ to $r_2$ in the graph $G$ exists a reachability path. In this case, we will say that $r_2$ is reachable from $r_1$. The relation $\leftrightarrow$ is binary, symmetric and irreflexive. According to Property 2 any pair of reachable interfaces is in the same broadcast domain.

For each $r \in I^{(2)}$, we will introduce a reachability set $RS_r \subset I^{(2)}$ that includes all interfaces reachable from $r$. The set of interfaces reachable from the interfaces that are $F^{(2)}$-related to the $r$, we will define as $CRS_r$. Due to the Property 1 $RS_r \cap CRS_r = \varnothing$.

Let us cite Proposition 1 from [12] which will be needed later. If a link interface $r_2$ is reachable from an interface $r_1$, then all interfaces reachable from interfaces commutating with $r_2$ are reachable from $r_1$, i. e. $CRS_{r2} \subset RS_{r1}$.

The reachability sets are to be filled from the data of network devices usually using SNMP (Simple Network Management Protocol) mainly form BRIDGE-MIB (cache of Spanning Tree Protocol and Address Forwarding Tables), Q-BRIDGE-MIB (AFT for VLAN enabled networks), CISCO-CDP-MIB (cache of Cisco Discovery Protocol), LLDP-MIB (cache of Link Layer Discovery Protocol) and IP-MIB (ARP cache) [12], [13].

## IV. The problem of incomplete data improvement

The reachability sets are the main tool for search and building connections in the link layer topology graph used in [5], [7], [13] etc. In [13] these sets are used to seek link layer connections and commutation edges ($L^{(2)}$ and $F^{(2)}$). However, as was shown in [4], [5], [12], available data on network topology might be incomplete and may not provide ability to build reachability sets with accuracy required for network topology discovery. Therefore appears a task of handling reachability sets to make them more accurate and applicable for network topology discovery process — a problem of incomplete data improvement. Let us introduce a formal definition of this problem.

Let the link layer interfaces set $I^{(2)}$ be ordered in some arbitrary sequence and for any interface $r \in I^{(2)}$ there is an index $i \in \overline{1, |I^{(2)}|}$, i. e. $r = r_i$. Then we can represent reachability sets as a matrix $R$ with size $|I^{(2)}| \times |I^{(2)}|$ where each row $R(i,)$ corresponds to the reachability set of interface with index $i$. For elements of this matrix the following rule is applied:

$$R(i,j) = \begin{cases} 1, & r_j \leftrightarrow r_i \\ -1, & r_j \in CRS_{ri} \\ 0, & \text{otherwise} \end{cases} \qquad (1)$$

Matrix $R$ will be called as link layer reachability matrix. The main matrix diagonal contains only zeros since the interface cannot be reachable from itself following to reachability relation definition.

Let us provide an example of matrix $R$ for the sample network from Fig. 2. The order for interface set is as follows (from left to right by the Figure): $(p_1, 1)$, $(p_3, 1)$, $(p_2, 2)$, $(p_4, 2)$, $(p_5, 1)$, $(p_5, 2)$, $(p_6, 1)$, $(p_6, 2)$, $(p_7, 0)$, $(p_8, 1)$, $(p_9, 2)$, $(p_{10}, 0)$. Then we get the following matrix $R$:

$$\begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0
\end{pmatrix}$$

Let us define the completeness of link layer reachability data using matrix $R$. Let the matrix $R_{correct}$ be a correct link layer reachability matrix of some graph $G$, i. e. rule (1) is true for $R$ and $G$. Then let us take a look at some same-sized matrix $R_{actual}$ whose elements are taken from the set $\{-1, 0, 1\}$. Matrix $R_{actual}$ will be called a complete link layer reachability matrix for the graph $G$ if $R_{actual} = R_{correct}$ or $\|R_{actual} - R_{correct}\| = 0$. Here and further the Euclidean norm is used. Similarly, we will call reachability set $RS_{ri}$ of interface $r_i \in I^{(2)}$ complete if $R_{actual}(i,) = R_{correct}(i,)$.

We will call $C(R) = \|R_{actual} - R_{correct}\|$ (a distance between correct and actual matrices) as a level of incompleteness of matrix $R_{actual}$. For the graph $G$ we will call a number

$C = C_G = C(R_G)$ as a level of incompleteness of link layer reachability data for graph $G$.

Now we can define the problem of incomplete data improvement as the following:

$$C(R_G) \to \min_R \\ \text{initial conditions: } R_0, G \quad (2)$$

i. e. a search for such a link layer reachability matrix for which the level of incompleteness will be minimal, having a graph $G$ and an original reachability matrix $R_0$. The specifics of this problem is such that the correct reachability matrix is unknown and therefore a level of incompleteness is unknown.

## V. An algorithm for solving the problem of incomplete data improvement

An algorithm for building an enterprise network topology was developed by authors in [12] and [13]. It consists of four steps: polling the network devices with SNMP; building topology graph vertices; building and improving reachability sets; building connection edges and indirectly found vertices (transparent devices etc.). In this section a more advanced replacement for the third step is provided.

### A. Building an initial reachability matrix

Input for the reachability matrix initialization is an incomplete graph $G$ with vertices and edges built previously and sets of data collected from the devices. First, an initial reachability matrix $R_0$ is being built using all available data sources same as in [13]. Due to the data incompletion risks, this matrix may be not complete.

For the improvement process we will need information on link layer commutation. Here we will present it as a symmetric matrix $F$ so that $F(i,j) = 1$ if $I_i^{(2)}$ and $I_j^{(2)}$ are $F^{(2)}$-related and 0 otherwise.

Next, we need to prepare matrix $R_0$ for further use to make elements of this matrix not contradictory to the reachability relation properties. Algorithm 1 will synchronize matrix elements with each other using only the matrix $R_0$ itself and the commutation matrix $F$. In this process, reachability relation's symmetry property and the Proposition 1 are used.

After the algorithm 1 has handled the matrix $R_0$ the latter will contain all reachability knowledge that is available directly from the data collected from network devices.

**Algorithm 1** The preparation of the reachability matrix

```
for all i ∈ 1,|I^(2)| do
  for all j ∈ 1,|I^(2)| do
    if R(i,j) = 1 and R(j,i) ≠ 1 then
      Set R(j,i) = 1
    end if
  end for
end for
for all i ∈ 1,|I^(2)| do
  for all j ∈ 1,|I^(2)| do
    if F(i,j) = 1 then
      for all k ∈ 1,|I^(2)| do
        if R(i,k) = 1 then
          Set R(j,k) = −1
        else if R(j,k) = 1 then
          Set R(i,k) = −1
        end if
      end for
    end if
  end for
end for
```

### B. Improving the reachability matrix

To infer missing data in the reachability matrix $R$ we can use properties of reachability relation and Proposition 1 from [12]. Let us provide a criterion of when the missing record on the reachability could be inferred. For the matrix $R$ we can consider a directed weighted graph $G(R)$. Its vertices are link layer interfaces and the matrix $R$ is an adjacency matrix for this graph. If $R(i,j) = 0$, then there is no edge from $I_i^{(2)}$ to $I_j^{(2)}$.

**Theorem 1.** *Consider a non-equal interfaces $I_i^{(2)}$ and $I_j^{(2)}$ that are not in the $L^{(2)}$ relation with each other. The interface $I_j^{(2)}$ is reachable from interface $I_i^{(2)}$ if and only if there is an interface $I_k^{(2)}$ such as $I_i^{(2)} \leftrightarrow I_k^{(2)}$ and graph $G(R)$ contains a path from $I_k^{(2)}$ to $I_j^{(2)}$ through the edges with negative weights.*

*Proof: (direct).* Suppose that $I_j^{(2)}$ is reachable from $I_i^{(2)}$. By definition of the reachability relation, graph $G$ contains a reachability path $w_r = (I_i^{(2)}, \ldots, I_j^{(2)})$. Since $I_i^{(2)}$ and $I_j^{(2)}$ are not connected, according to the reachability definition there are interfaces $I_k^{(2)}$ and $I_t^{(2)}$ such as they are in the $F^{(2)}$ relation and are located on the path $w_r$: $w_r = (I_i^{(2)}, I_k^{(2)}, I_t^{(2)}, \ldots, I_j^{(2)})$. According to the reachability definition $I_i^{(2)} \leftrightarrow I_k^{(2)}$ and $I_t^{(2)} \leftrightarrow I_j^{(2)}$. Then, according to (1) $R(k,j) = -1$. Apparently, there exists such interface $I_k^{(2)}$ that $I_i^{(2)} \leftrightarrow I_k^{(2)}$ and $G(R)$ contains a path $w = (I_k^{(2)}, I_j^{(2)})$ through the edges with negative weights.

*(Converse).* Suppose $I_i^{(2)} \leftrightarrow I_k^{(2)}$ and $G(R)$ contains a path $w = (I_k^{(2)}, \ldots, I_j^{(2)})$ through the edges with negative weights. According to Proposition 1 all interfaces that are reachable from interfaces in the $F^{(2)}$ relation with $I_k^{(2)}$, are reachable from $I_i^{(2)}$. By definition of the graph $G(R)$ there are edges with negative weights from $I_k^{(2)}$ to each of such interface.

Suppose that the path $w$ contains only two vertices: $w = (I_k^{(2)}, I_j^{(2)})$. Then $I_i^{(2)} \leftrightarrow I_j^{(2)}$ according to Proposition 1.

Then, let us suppose that the path $w$ contains more than two vertices: $w = (I_k^{(2)}, I_t^{(2)}, \ldots, I_j^{(2)})$. According to Proposition 1 $I_i^{(2)} \leftrightarrow I_t^{(2)}$. Therefore, the interfaces $I_i^{(2)}$ and $I_j^{(2)}$ and the path $w_1 = (I_t^{(2)}, \ldots, I_j^{(2)})$ which is a subpath of $w$ satisfy the original condition of the reversed theorem. Since the interfaces $I_i^{(2)}$ and $I_j^{(2)}$ are non-equal the path $w$ is finite. Let its edge-length be equal to $N$. Then, the length of $w_1$ is $N-1$. Similarly using Proposition 1 we can discard further edges and get a path $w_{N-1} = (I_q^{(2)}, I_j^{(2)})$. Then, as has been shown previously $I_i^{(2)} \leftrightarrow I_j^{(2)}$. ■

Using the condition and the layout of the proof of the Theorem 1 we can provide an Algorithm 2 for building more complete reachability matrix. Input data for this algorithm is an initial prepared matrix $R_0$ and the commutation matrix $F$.

**Algorithm 2** The improvement of the incomplete reachability matrix

```
for all i ∈ 1,|I^(2)| do
  Set Visited = {i}
  Set Q = Empty queue
  for all j ∈ 1,|I^(2)| do
    if R(i,j) = 1 then
      Add j into Q
    end if
  end for
  while Q is not empty do
    Set j = dequeue(Q)
    Add j into Visited
    for all k ∈ 1,|I^(2)| do
      if R(j,k) = −1 then
        if R(i,k) ≠ 1 then
          Set R(i,k) = 1
        end if
        if R(k,i) ≠ 1 then
          Set R(k,i) = 1
        end if
        if t ∉ Visited then
          Add t into Q
        end if
      end if
    end for
  end while
end for
Call Algorithm 1.
```

Algorithm 2 is based on the breadth-first search through the edges with negative weights. At the end of this algorithm, the Algorithm 1 is called to synchronize all negative elements of matrix $R$ with the positive elements. After handling by the Algorithm 2 all elements of matrix $R$ that match condition of the Theorem 1 the Rule 1 will be true. The Algorithm 2 is correct since it follows the condition and the layout of the proof of the Theorem 1.

Let us consider a case excluded from conditions of the Theorem 1.

**Theorem 2.** *Consider interfaces $I_i^{(2)}$ and $I_j^{(2)}$ in the $L^{(2)}$ relation with each other. If originally $R_0(i,j) = R_0(j,i) = 0$, then Algorithm 2 does not tell if these interfaces are reachable from each other.*

*Proof: (by contradiction).* To find the reachability between $I_i^{(2)}$ and $I_j^{(2)}$ we must have such an interface $I_k^{(2)}$ that $I_i^{(2)} \leftrightarrow I_k^{(2)}$ and the graph $G(R)$ contains a path from $I_k^{(2)}$ to $I_j^{(2)}$ through the edges with negative weights. Let us suppose that such a path exists. Then, according to the definitions of the reachability relation and the graph $G(R)$, there is such a path $w = (I_i^{(2)}, \ldots, I_k^{(2)}, I_t^{(2)}, \ldots, I_q^{(2)}, I_j^{(2)})$ in graph $G$ in which the interfaces $I_k^{(2)}$ and $I_t^{(2)}$ are $L^{(2)}$-related ($I_t^{(2)}$ and $I_q^{(2)}$ might be equal). But according to the condition of the theorem $I_i^{(2)}$ and $I_j^{(2)}$ are $L^{(2)}$-related too.

Interfaces $I_i^{(2)}$ and $I_q^{(2)}$ could not be equal as then the path $w_1 = (I_k^{(2)}, I_t^{(2)}, \ldots, I_i^{(2)})$ (where $I_i^{(2)} = I_q^{(2)}$) in graph $G$ would not satisfy the definition of the reachability path (it starts with a commutation edge). But there could not be any other link layer path between $I_i^{(2)} = I_q^{(2)}$ and $I_k^{(2)}$ according to Property 1. Therefore it is impossible to use Theorem 1 and Algorithm 2 to handle the case in the current theorem and to find the reachability between $I_i^{(2)}$ and $I_j^{(2)}$. ■

Using Theorems 1 and 2 we can provide a criterion for the possibility of building the complete link layer reachability matrix $R$:

**Criterion 1.** *Let $C(R_0) \neq 0$. According to Theorems 1 and 2 Algorithms 1 and 2 will solve the problem of incomplete data improvement (2) with $C(R) = 0$ if for any two interfaces $I_i^{(2)}$ and $I_j^{(2)}$, which are in $L^{(2)}$ link layer connection relation, it is true that $R_0(i,j) = 1$ and/or $R_0(j,i) = 1$.*

So, the reachability matrix completeness directly depends on the available data on the link layer connections. The condition of Criterion 1 is easy to achieve in the real networks if all available data is used for reachability set creation (AFT, ARP, STP, CDP, LLDP cache), however in some cases even this method may not guarantee the condition to be met. To ensure the condition is met in the network same methods as in [4] could be used: ping every network device before data collection. A single major presumption for the provided criterion is that the network devices do not provide incorrect reachability data.

### C. Computational complexity of the algorithms

Let us estimate the (asymptotic) time complexity of the developed algorithms.

Algorithm 1 handles all link layer interfaces in the amount of $|I^{(2)}|$. For each of them, interfaces in commutation relation are handled in the maximal amount of $|I^{(2)}| - 1$. For each pair of commutating interfaces, the algorithm handles all interfaces reachable either from one or another interface of the pair, the maximal amount is $|I^{(2)}|$. Therefore the time Algorithm 1 complexity could be estimated as $O(|I^{(2)}|^3)$.

Algorithm 2 handles all link layer interfaces in the amount of $|I^{(2)}|$. For each of them the breadth-first search in the graph $G(R)$ with $|I^{(2)}|$ vertices is used which visits all vertices. In this BFS for each interface all other interfaces are handled. At the end the Algorithm 1 is called outside the iterations. Therefore the time Algorithm 2 complexity could be estimated as $O(|I^{(2)}|^3)$.

In our scope, for an average network with 1000 various devices and 10 VLANs the number of interfaces ($|I^{(2)}|$) is about 2000–3000.

## VI. Simulation, Testing and Comparison to Other Results

The Algorithms 1 and 2 were implemented as a part of the system for automated network topology graph building [12], [13]. They replace the reachability sets building algorithms presented in [13].

### A. Testing with the simulated data

The algorithms implementation were tested in 1000 generated networks of different sizes: 1–3 routers, 2–50 switches, 40–1000 hosts, 1–5 VLANs, 85–2000 interfaces. At first, we generate a network, then build a correct reachability matrix $R_{correct}$ and a matrix $R_0$, where for each pair $r_i, r_j \in I^{(2)}$ of connected interfaces $R_0(i,j) = 1$, and all other elements of matrix $R_0$ are zeros, meaning only 1–5% of reachability data available initially. Each time we were able to obtain a *complete* reachability matrix using the Algorithm 2, i. e. we were able to infer 95–99% of reachability data.

We also studied the situation when the condition of the Criterion 1 is not fully achieved. Using the same generation method we dropped out a number of randomly chosen reachability records for directly connected interfaces. We used 1000 networks sized from 40 to 130 devices, 1–5 VLANs and 84–500 interfaces. The initial completeness level was varied from 10 to 90%.

The results are presented in Table I. Column 2 and 3 contains average incompleteness level $C(R)$ and percentage of completeness after applying our method. Columns 4 and 5 contains a minimum an maximum amount of reachability records that the Algorithm was not able to discover. Column 6 contains average amount of undiscovered records.

TABLE I. Testing with missing direct connections in RS

| Missing connections | Avg. incompleteness | Avg. completeness, % | Min. undiscovered records | Max. undiscovered records | Avg. undiscovered records |
|---|---|---|---|---|---|
| 0 | 0 | 100 | 0 | 0 | 0 |
| 1 | 7.81 | 99.2 | 2 | 2102 | 50.3 |
| 2 | 9.66 | 98.68 | 4 | 3134 | 72.5 |
| 3 | 12.02 | 98.08 | 6 | 2644 | 92.1 |
| 4 | 13.79 | 97.77 | 8 | 2448 | 110.1 |
| 5 | 15.75 | 96.86 | 10 | 3344 | 152.2 |

According to the testing results, the Algorithm is still able to produce near-complete reachability sets in most cases when amount of missing reachability records for the directly connected interfaces is not very high. In some cases, the algorithm was not able to find only these initially missing records for connected interfaces. In other cases, some records may be missing due to the critical value of the missing records for connected interfaces of the service network devices — switches.

### B. Testing with the real data

The algorithm implementation was also tested in the real computer network of Petrozavodsk State University. For the discovery this network provides: 1 router Cisco 7600, 4 layer 3 switches Cisco 3750 and Cisco 3850, 52 various Cisco layer 2 switches accessible by SNMP and 296 non-accessible switches, 795 hosts and 101 VLANs configured (1148 devices in total). The amount of link layer interfaces is 6037.

The primary data sources on reachability in this network appeared to be CDP, STP and AFT. Using data provided by the network administrators and obtained by ourselves, we built a matrix $R_{correct}$ for this network. Using the algorithm implementation we managed to build a *complete* link layer reachability matrix based on the collected data.

10114 reachability records were found using Algorithm 1 and 79558 records using Algorithm 2 after that. The Total reachability records amount is 99270.

The reachability data inferred part is 90%. Such a high number is explained by the big amount of VLANs and inaccessible switches: a lot of data were initially unavailable, however CDP and STP cache have provided most of needed data on the direct connections.

### C. Comparison to other results

In [13] we developed the original reachability set improvement algorithm with complexity estimated as $O(|I^{(2)}|^6)$. Also [13] provided no criterion for the case when the algorithm is applicable, and no theoretical correctness evaluation. In our work we developed an algorithm with complexity estimated as $O(|I^{(2)}|^3)$ which is significantly better result. Also we provided an applicability criterion for this algorithm and the theoretical and practical correctness evaluation.

In [4] authors introduced a criterion for the set of input data to be usable for the topology discovery. In this approach, in each broadcast domain a root node is selected. For any network node, every interface, from which the root node is reachable, must contain only the interface of the root in its reachability sets. Reachability sets of every other interface must be complete. Then, the reachability sets are being prepared with an algorithm with $O(|D|^2)$ complexity where $|D|$ is a number of network nodes. In our work we provided a significantly lighter criteria which does not require any of interface to have a complete reachability set. Moreover, our approach is applicable to networks with VLANs and only a little more complex ($O(|I^{(2)}|^3)$).

In [5] authors originally introduced a concept of the reachability set and described some of its properties. For the improvement of the incomplete reachability sets they used a set of heuristic improvement rules. Mostly these rules are based on the knowledge of the already discovered link layer connections which may make the discovery process inaccurate at its beginning. In [5] authors have practically proven that their approach may work in the network with 50%-complete data but did not provided any theoretical research to support this. Their method of reachability set improvement is $O(|D|^3)$-complex. In our work we utilized an idea of reachability sets, but we are basing our reachability set term not on the IP-subnets but on the broadcast domains which makes it easy to use in networks with VLANs. We introduced and proved a condition in which our approach will work. This condition is significantly lighter than 50% complete reachability sets and requires only 1–5% RS completeness. At the same time, our algorithm is of the same complexity.

As an opposite to [6], [7], [11] we provided theoretical support of the correctness of our methods, provided an exact algorithm for reachability set improvement and a criterion for when our method is applicable.

## VII. Conclusion

The description of the logical and physical topology of an enterprise computer network is required for many network management tasks. However, automated building of such description is complicated due to the connectivity data incompleteness. Here is where the problem of incomplete data improvement arises.

The criterion provided in this paper allows us to determine when the connectivity data is sufficient for topology description building and when it might be improved to its completeness. Algorithms presented in this paper allow to solve the denoted problem when the criterion condition is met.

The algorithm implementation test and evaluations provided in the paper demonstrates its correctness and applicability for using in the real enterprise networks. It has also been shown that provided methods of the problem solving are more efficient than the other existing methods: the algorithm is able to infer up to 99% of missing data.

## Acknowledgment

The described research was supported by The Foundation for Assistance to Small Innovative Enterprises (FASIE), grant no. 12462GU/2017 from 02.04.2018, code 0038514.

## References

[1] S. Ravindran, C. Huang, K. Thulasiraman, "A dynamic managed VPN service: architecture and algorithms", *2006 IEEE International Conference on Communications*. 2006, Vol. 2, pp. 664–669.

[2] L. Sivakumar, J. Balabaskaran, K. Thulasiraman, S. Arumungam, "Virtual topologies for abstraction service for IP-VPNs" *2016 17th International Telecommunications Network Strategy and Planning Symposium (Networks)*. 2016,pp. 213–220.

[3] Sun Y., Shi Z., Wu Z., "A discovery algorithm for physical topology in switched ethernets", *The IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05) l*. 2005. pp. 311–317.

[4] 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.

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

[6] H. Gobjuka, "Topology discovery for virtual local area networks" *INFOCOM, 2010 Proceedings IEEE*. 2010, pp. 1–5.

[7] L. Zichao, H. Ziwei, Z. Geng, M. Yan, "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.

[8] X. Ma, T. Yu, "An algorithm of physical network topology discovery in multi-VLANs", *TELKOMNIKA*. 2016. vol. 14, no. 3A. pp. 375–379.

[9] S. Zhou, L. Cui, C. Fang, S. Chai, "Research on Network Topology Discovery Algorithm for Internet of Things Based on Multi-Protocol", *2018 10th International Conference on Modelling, Identification and Control (ICMIC)*. IEEE, 2018. pp. 1–6.

[10] He G., Liu Y., Wu X., "A New Algorithm for Physical Topology Discovery Based on STP and AFT", *2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics*. 2013. vol. 1. pp. 486–490.

[11] J. Zhou, Y. Ma, "Topology discovery algorithm for ethernet networks with incomplete information based on VLAN" *Network Infrastructure and Digital Content (IC-NIDC), 2016 IEEE International Conference on*. IEEE, 2016. pp. 396–400.

[12] A. A. Andreev, A. S. Kolosov, A. V. Voronin, I. A. Bogoiavlenskii, "A graph model of the topology of physical, link and network layers of an enterprise network", *Proceedings of the 19th Conference of Open Innovations Association FRUCT*. 2016, pp. 3–9.

[13] A. A. Andreev, I. A. Bogoiavlenskii, "An Algorithm for Building an Enterprise Network Topology Using Widespread Data Sources", *Proceedings of the 21th Conference of Open Innovations Association FRUCT*. 2017, pp. 34–43.