Graph Model of an IEEE802.1 Based Network Structure and Its Application for Enterprise ICT-infrastructure Discovery

Anton Andreev Iurii A. Bogoiavlenskii Aleksandr Kolosov

Department of Computer Science
Petrozavodsk State University

May 14, 2015

Areas of Network graph use

ICT-infrastructure graph (the Network) — most convinient way to represent data about its structure.

- Network documentation
- Network management and optimization
- Modelling and designing of network
- Load simulation for network segments

The Network graph discovery problem

- Lack of standard tools for device detection
- Heterogeneity and incompleteness of data
- Diversity of network devices
- Network's constant changing
- Comlex structure of Network (due to VLANs, IP-subnets, VPN)

Project goal

The Nest platform

- Network graph building
- Graph visualizing
- Tools for interaction with graph
- Data flows modeling
- Depiction and visualization of spatial and organizational structures of an enterprise

The goal

To develop, implement and test methods for automatized discovery of Network structure graph

Network structure modelling

Importance of Network structure model

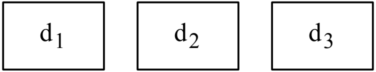
- Formal relationship description
- Abstraction for devices
- Math tools for algorithms
- Ability to generate certain Network graphs

The modelling problem

- Modelling of static structure of 1, 2, 3 layers of OSI model
- Modelling of Networks built in accordance to IEEE 802.1 и IP (RFC 791) standards
- Description of logical structures (VLAN, IP-subnets)

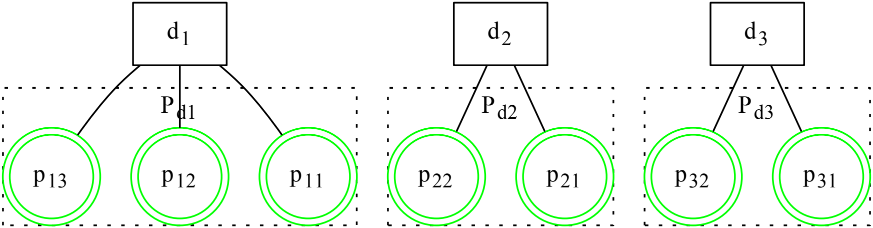
Physical layer structure modelling

D — set of devices;
$d_1, d_2, d_3 \in D$



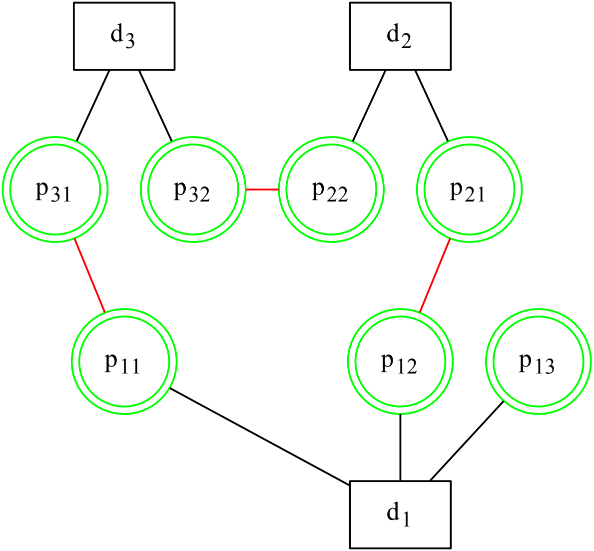
Physical layer structure modelling

P — set of ports;
$P_d \subseteq P$ — set of ports of device $d \in D$;
O — set of edges of ownership



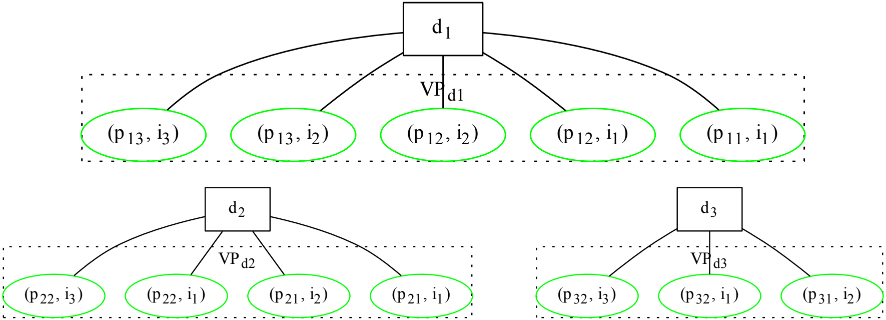
Physical layer structure modelling

L — set of edges between ports



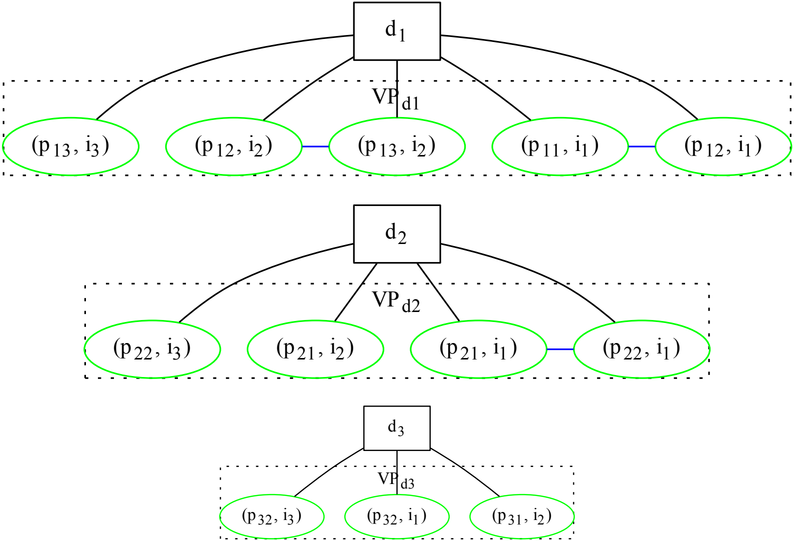
Link layer structure modelling

VP — set of interfaces;
$VP_d \subseteq VP$ — set of interfaces of device $d \in D$;
VO — set of edges of ownership for interfaces



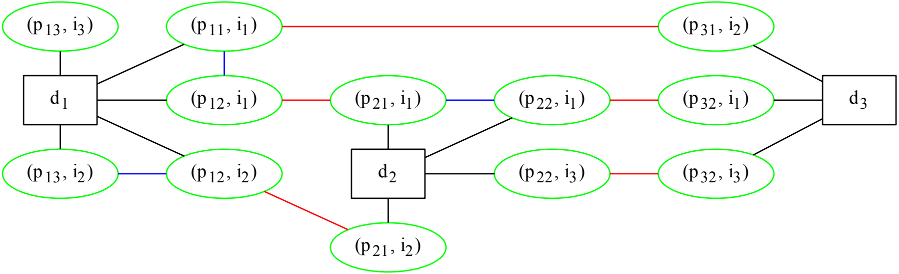
Link layer structure modelling

VC — set of switching edges;



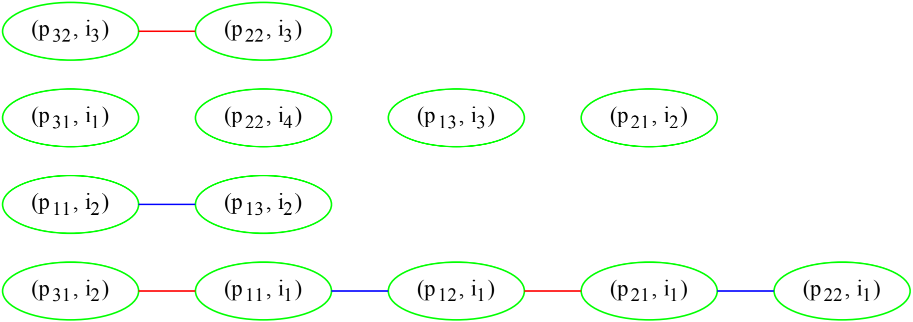
Link layer structure modelling

VL — set of edges between interfaces of different devices



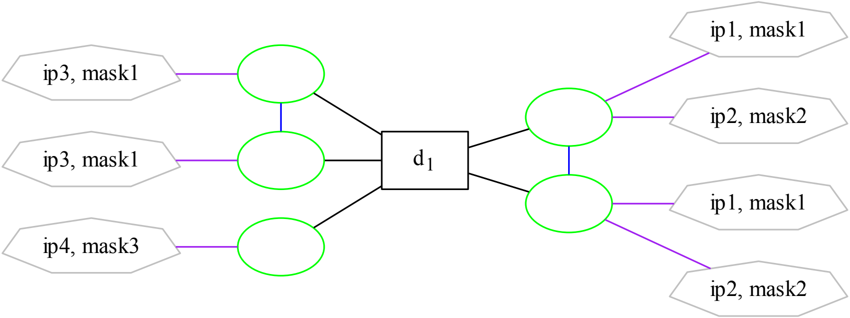
Broadcast domains modelling

BD — set of broadcast domains in network



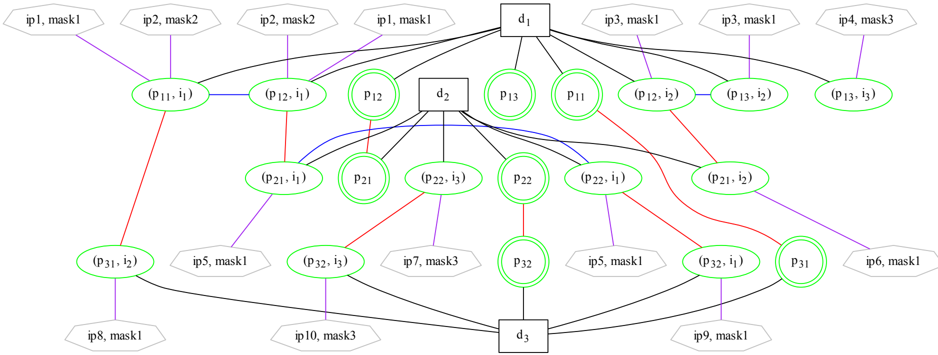
Network layer structure modelling

NP — set of network interfaces;
NO — set of edges of ownership for network interfaces.



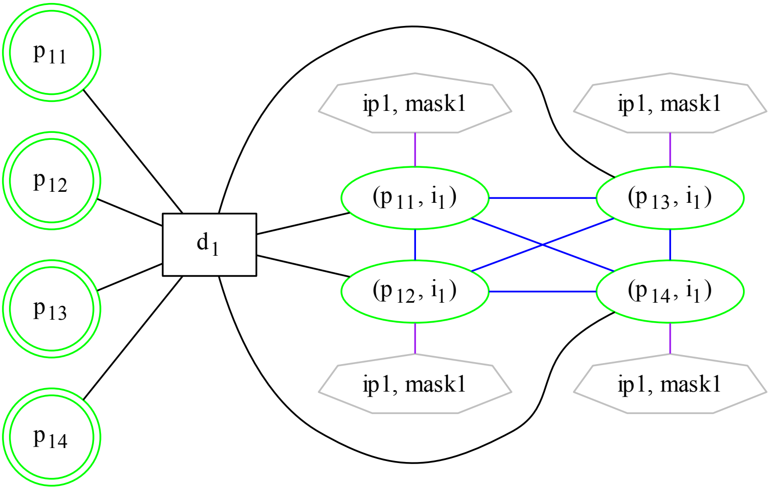
Common Network structure modelling

Graph $G =< V, E >$;
$V = D \cup P \cup VP \cup NP$, $E = O \cup VO \cup NO \cup L \cup VC \cup VL$



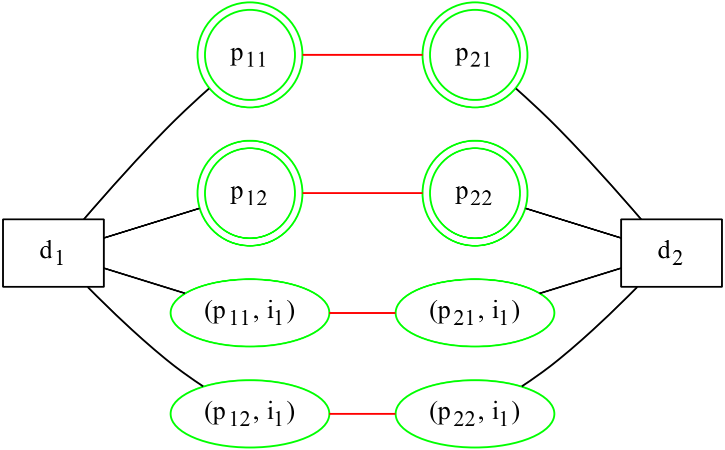
Examples

Switch



Examples

Link aggregation



Available data on Network structure

Retrieving using Simple Network Management Protocol

Ports and interfaces

- IF-MIB
- IP-MIB
- Q-BRIDGE-MIB, VTP-MIB, CISCO-VLAN-MEMBERSHIP-MIB, ...

Physical and logical connections

- Cisco Discovery Protocol: CISCO-CDP-MIB
- Link Layer Discovery Protocol: LLDP-MIB
- Spanning Tree Protocol: BRIDGE-MIB
- Address Forwarding Table: BRIDGE-MIB
- Address Resolution Protocol: RFC1213-MIB, IP-MIB

Graph discovery algorithm

1. **Data collection**
   - Data about devices (MACs, names, etc.)
   - Data about VLAN and IP
   - Data about connections
2. **Graph vertices creation**
   - Vertices creation for devices accessible via SNMP
   - Vertices creation for unaccessible devices using indirect data
3. **Search for potential connections**
   - Using data about direct connections
   - Using reachability data
4. **Graph edges building**

Potential connections filtering example

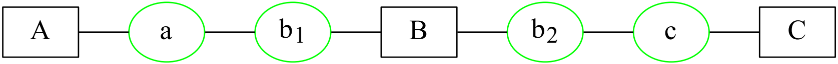
$$A, B, C \in D$$

$a \in VP_A$	$b_1 \in VP_B$	$b_2 \in VP_B$	$c \in VP_C$
b_1	a	c	b_2
c			a

$$\Downarrow$$

$a \in VP_A$	$b_1 \in VP_B$	$b_2 \in VP_B$	$c \in VP_C$
b_1	a	c	b_2

$$\Downarrow$$



Algorithm implementation

Nestopo — Nest subsystem for automatized Network graph building

- Availability for Network structure graph building
- Use Simple Network Management Protocol (SNMP) for data retrieving
- Handling standard MIBs (Management Information Base)
- Availability to add vendor specific MIBs for handling
- Subsystem configuration:
  - Start address for Network traversing
  - Devices access parameters
  - Algorithm partial execution

Testing of Nestopo

Testing necessity

- Built graph correctness verification
- Various devices interaction examination
- Data incompleteness influence analysis

Testing methods

- Real network testing
- Use of virtual labs and Network simulators
- Automated testing using network structure generation

Testing Nestopo using PetrSU network

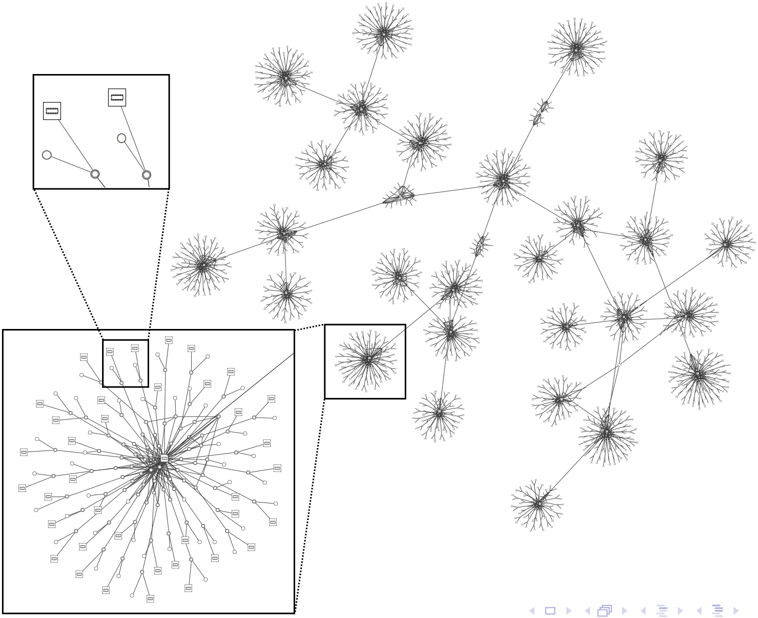
Benefits

- Maximal network structure realism
- Interaction with real devices

Disadvantages

- Network structure change unpredictability
- Result graph virification difficulty
- Lack of influence on configuration
- Lack of tests variety
- Long data collection duration

Graph of physical structure of PetrSU Network



Testing Nestopo using GNS3 labs

GNS3 — Graphical Network Simulator 3

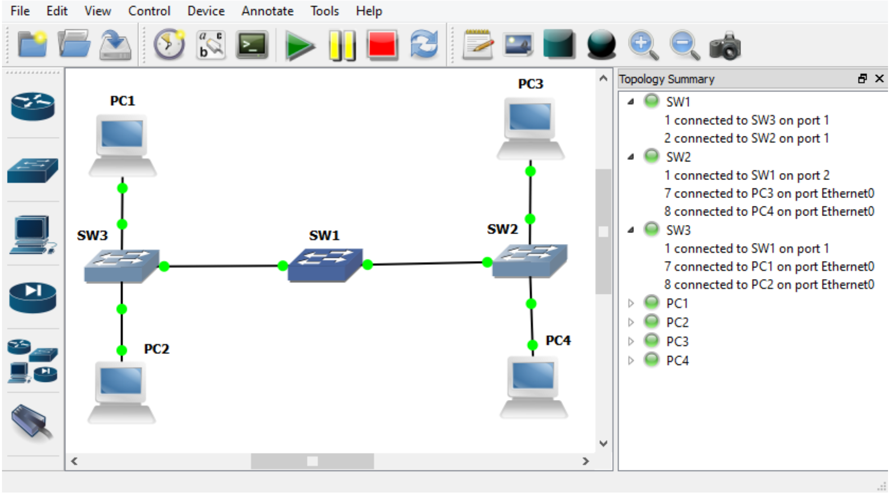
Benefits

- Configuration opportunities
- Network behaviour predictability
- Simplicity of result graphs verification

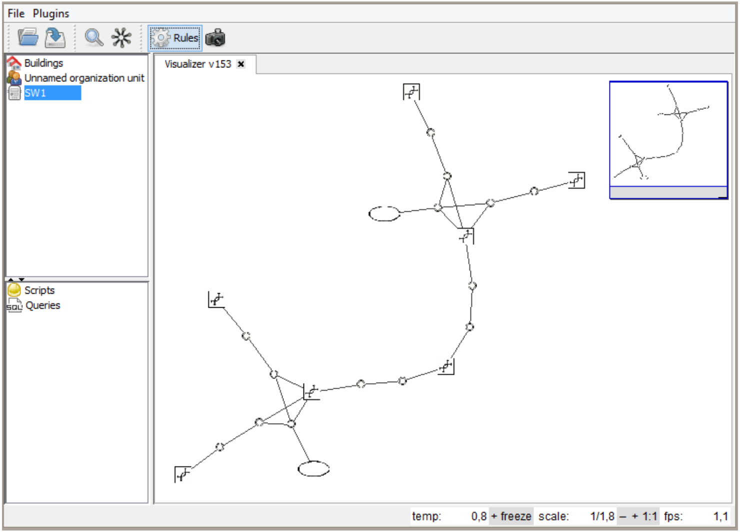
Disadvantages

- Long duration of handmade configuration
- Tiny scale of Networks
- Invariety of available devices

Testing Nestopo with GNS3 labs



Testing Nestopo with GNS3 labs



Testing using Network structure generation

Benefits

- A lot of varied network structures with any scale
- Formation of any type of data
- Availability of testing automatization
- Imitation of mechanics that can't be seen in accessible networks

Disadvantages

- Need for implementing generation mechanics
- Can't replace testing in real networks

Testing using generation: Netgen subsystem

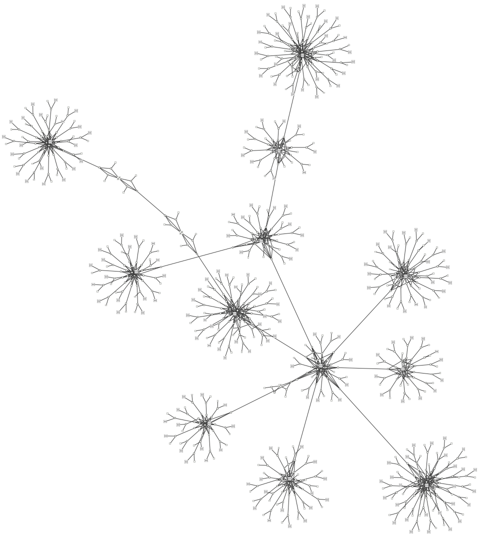
Generation method

1. Sequential generation of three layers of model
2. Physical structure generation in shape of scale-free network (Barabasi-Albert algorithm)
3. Data generation

Opportunities of Netgen

1. Structure generation configuration: number of nodes, VLAN sizes, etc.
2. Data generation configuration: data completeness degree
3. Storing result graph in various formats

Testing using generation: examples



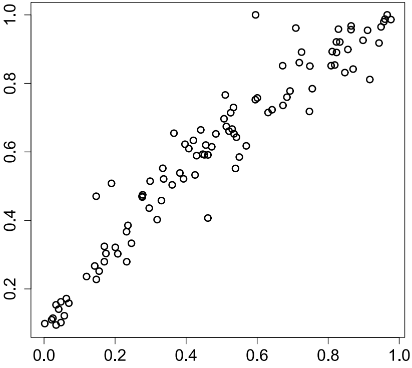
Generated	
Devices	15
Including hubs	3
Hosts	202
Links	216

Discovered	
Devices	14
Including hubs	2
Hosts	202
Links	195

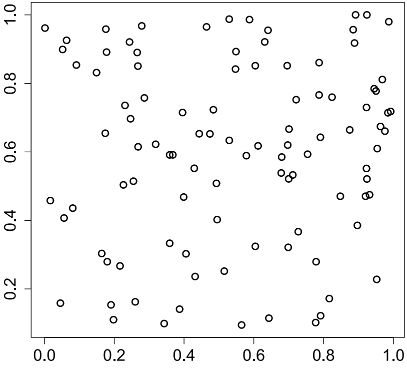
Matched	
Devices	14
Hosts	202
Links	195

Testing using generation: examples

The dependence of the found links part on the data completeness:



AFT completeness



ARP-cache completeness

Results

- Abstract Network structure model
- Algorithm of an automatized network graph discovering
- Program system for network graph discovering
- Subsystems of random network structure generation and automated testing

Code metrics

NetworkModel: 921 LOC, 5 classes
Nestopo: 5832 LOC, 57 classes
Netgen: 2245 LOC, 12 classes

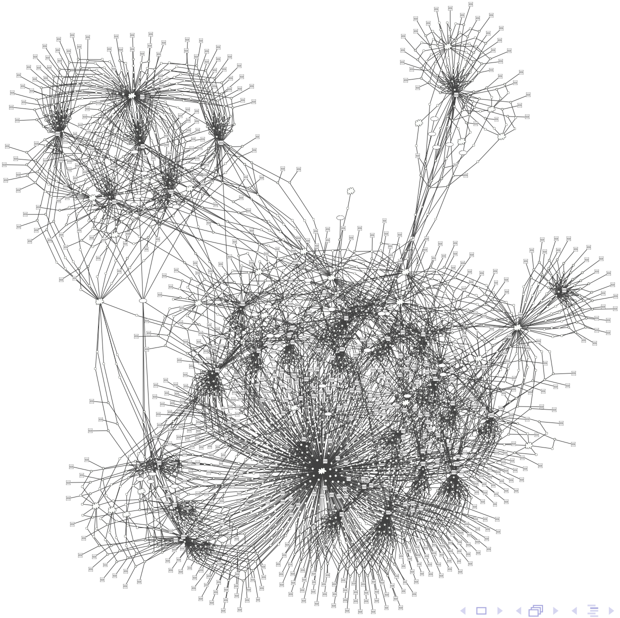
Further work

Search for additional data sources
Wireless network structure modeling
Algorithm improving

Thank you for your attention!

andreev@cs.karelia.ru

Graph of structure of PetrSU Network



Related articles

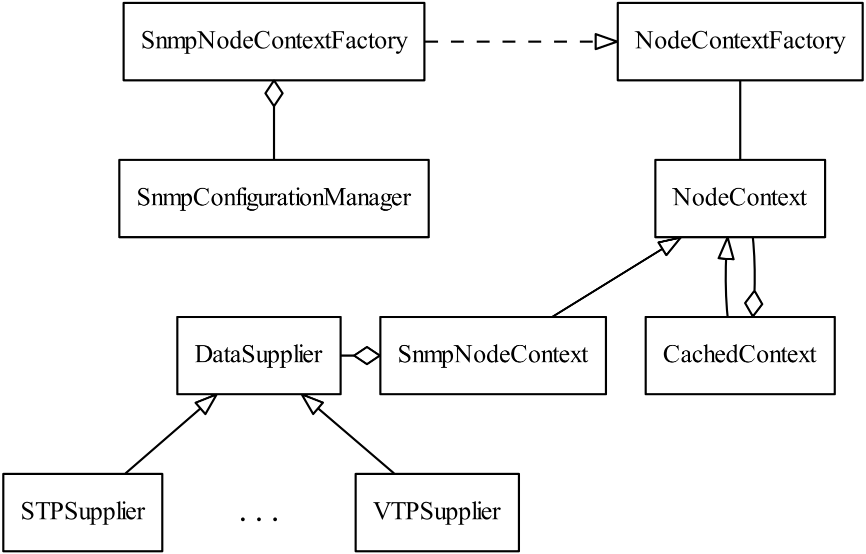
Hassan Gobjuka, Yuri J. Breitbart.
Ethernet Topology Discovery for Networks With Incomplete Information. 2010

В. В. Воеводин, К. С. Стефанов.
Автоматическое определение и описание сетевой инфраструктуры суперкомпьютеров. 2014

Li Zichao, Hu Ziwei, Zhang Geng, Ma Yan
Ethernet topology discovery for virtual local area networks with incomplete information. 2014

Nestopo architecture

Class diagram of NodeContext compoment



Nestopo architecture

Class diagram of components Collector, Creator, Connector