Formal Analysis Approach on Networks with Dynamic Behaviours

Gayan de Silva, Petr Matoušek, Ondřej Ryšavý, Miroslav Švéda

Faculty of Information Technology
Brno University of Technology
Brno, Czech Republic
Email: xdesil00@stud.fit.vutbr.cz, { matousp, rysavy, sveda }@fit.vutbr.cz

***Abstract*—Formal verification and validation techniques such as model checking are not widely used in computer networks. These methods are very useful to identify configuration errors, identify design problems and predict network behaviours under different network conditions. This paper describes the two main components of the formal verification process, formal modelling and the analysis process. For formal modelling, computer networks configured with dynamic routing protocols such as RIP, OSFP or EIGRP are considered. For the analysis, reachability and security properties are evaluated as the behavioural properties in the case of device or link failures. Graph Theory is used to implement the model and predict the network behaviours. The process of building the model, grouping the network states which have common behaviours and predicting behaviours are the core work of this paper. Furthermore this paper details a method to reduce the state space and hence eliminate the state space explosion.**

***Index Terms*—formal modelling, analysis, networking, reachability, routing, state space reduction**

I. Introduction

Modern computer networks are becoming large and more complex. Therefore, network administrators are compelled to use network management tools to monitor their networks. Current scanning and testing tools are useful for analysis of stable networks or for analysis of the networks after its topology has changed. These tools cannot predict network behaviours before changing the topology due to link or device failures. Therefore, the use of formal verification techniques to evaluate behaviours has become a demand and a necessity in this area.

Formal verification and validation techniques are used to check the correctness and the validity of required properties under different conditions. One of the most widely used methods is model checking. The main areas of the research are online reading the Cisco configuration files, transforming them into model, developing the formal model, developing the transition system, verifying the properties using model checking and incorporating the model into a simulator for testing the real environments. Since there are many dependencies and behaviours in computer networks, our initial objective is to build a model and develop an analytical process for a limited area in computer networks.

The term dynamic network means (in this paper) the use of dynamic routing protocols such as RIP [1], OSPF [2] or EIGRP [3]. The main objective of implementing dynamic network is to reduce failures, improve service levels, reduce dependability and improve the availability [4]. Initially to predict the behaviours, the analysis of reachability and the security properties are considered. In this paper, *reachability* refers to the feasibility of establishing a communication path between a given source and a destination. The term *security property* refers to the filtering rules implemented by Cisco Access Control Lists (ACLs) [5]. The formal model is named as Modified Topology Table (MTT) and it is capable of predicting the above properties for any given network state [definition 8]. Section II formally defines the required properties of dynamic networks. The detail implementation of the model is explained under the section III.

To eliminate the state space explosion during model checking process, the state space needs to be reduced both in the modelling and the analysis stages. This is achieved by grouping the network states which have similar behaviours and will be described in detail under computing general state section III-A.

Section IV explains the analysis process which will be automated in future with a model checking tool and section V discusses the results and a comparison of our approach to other commonly used approaches. Section VI draws the conclusions and the last part of the paper, section VII briefs on our future work and the directions.

A. Motivation

The effect of link up/down changes on a small network segment in a large dynamic network can propagate to other network segments. It is difficult to predict these propagating effects. Since there are many combinations of dependencies, generating test cases covering all possibilities and checking on the production network are not practically possible. The commonly used method to identify the paths after link failures is the on-demand method (computing the paths after link failures). On-demand method is efficient to analyse change of few network states and repeating this method many times

978-1-4244-7286-4/10/$26.00 ©2010 IEEE

to identify paths in different conditions is less efficient. So we are motivated to research on a complete solution to implement a global model with less state space and analyse the network properties.

B. State of the Art

The requirement of developing a system for predicting network behaviours in dynamic environment was addressed previously by Automated Network-wide Security Analysis (ANSA) project. The objective of ANSA project is to develop a system which reads Cisco configurations from networks configured with dynamic routing protocols and predict network behaviours under different conditions. The authors of ANSA project have outlined how formal modelling can be used to predict reachability and security properties. This has been addressed in the paper [6]. The ANSA project has shown that it is possible to develop a model with a combination of static and dynamic behaviour using techniques described in [7], [8] and [9].

In [7], authors show that a static model can be implemented to predict network behaviours and security properties. The detailed approach is not directly shown but the authors discuss some approaches which could be applied for modelling and analysis. The main goal of [7] is to develop a model for the full domain considering all the possible scenarios such as Inter-network Operating System (IOS) bugs of devices, configuration errors, static routing, networks with mixture of different protocols, dynamic routing, networks with Network Address Translations (NAT) and packet filtering. Since this has very broad area of dependencies which can lead to a huge state space. Approach in [7] has some differences to our approach. The main difference is that they compute all virtual paths according to the states of links, devices, applied filters, IOS bugs, states of routing tables, traffic etc. Then the computed virtual paths are compared with the available physical paths to check the reachability. This approach has less efficiency since it computes the virtual paths first which require complicated and time consuming algorithms as no available physical path can be found. In our approach, we first compute the available physical paths and then do the analysis to find the communication path based on the network state.

In pre-computed routing tables approach [6], constructing the routing tables for each router and for each state is a time consuming process. Routing algorithms and routing updates depend on the routing protocols used. To predict properties for each state, routing tables of each router has to be reconstructed. This is different from our approach and the comparison of both methods can be found under section V. Further paper [6] shows how formal modelling can be used to model behaviours of dynamic networks and describes in detail the filtering and packet matching procedures which have been used in our modelling.

All above articles assume that after the link failures, the network will converge to another topology and remain same. The research done by [10] considers further link failures due to loads after the network convergence from first link failures. They have considered the statistics of link failures and limited the number of consequent link failures. This reduces the state space. From the previous statistics, a link failure has the probability of 10^{-3}, hence only 3 link failures at a time is considered. One advantage of this method is that it eliminates working on extreme cases which have very low probability of occurrence in practise.

Another algorithmic framework based on probabilities to analyse the network failures and link overloads is discussed in [11]. This analysis also taken the link overloads and its risks after network failures. The main objective of the paper is to address the availability and the dis-connectivity between nodes.

The approaches described above are different from this paper. We use a unified model defined in [7], but employs different approach for analysis. Unlike [6], the analysis does not pre-compute all routing tables in order to verify network reachability. The probabilistic data are not used for our analysis as in [10] and [11]. These two papers are mainly considered to analyse the communication links after a network failure. Our work is mainly oriented to model a global network model (MTT) to confirm the validity of different properties as required in security auditing. Further our work is aimed to represent the network with less state space and an effective way to reduce the state space.

C. Contribution

This paper introduces a novel approach in formal modelling and analysing reachability and security properties of networks configured with dynamic routing protocols and an effective method to reduce the state space. This analysis has few iterations and hence quicker to predict network reachability and security properties under any given network state. We also show the steps to determine paths without constructing the routing tables. One major advantage of this approach is that it is independent of the dynamic routing protocol used in the network.

II. Preliminaries

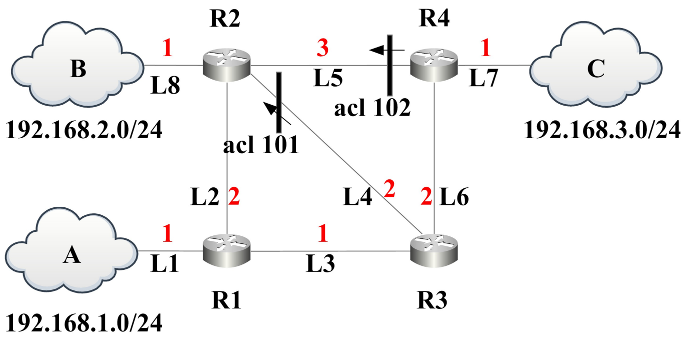
A. Formal model of the network

Consider the example network shown in Fig.1 with R–routers, A,B,C–networks, L–links and the link cost by the number associated with each link.

Definition 1 (Network): A network is a graph which defined as a tuple $N = \langle R, L \rangle$, where R is a finite set of network devices, and $L \subseteq R \times R$ is a finite set of physical connections between adjacent devices such that every physical link between two adjacent devices R_i and R_j is a pair of channels $l_{ij} = \langle R_i, R_j \rangle$ and $l_{ji} = \langle R_j, R_i \rangle$.

The model of two channels over one physical link enables modelling asymmetric communications. For simplicity the workout of the example network considers symmetric links and hence the notation has one suffix eg. l_1.

Definition 2 (Next-Hop): Function $NH(R, L) : R \times L \rightarrow R$ returns the connected adjacent device for a given device



acl 101 : deny ip 192.168.3.0/24 192.168.2.0/24
permit ip any any

acl 102 : deny ip any host 192.168.1.5 eq www
permit ip any any

Fig. 1. Dynamic network – devices, links and ACLs

and a link. Formally, $NH(R_i, l_{ij}) = R_j$ where $R_i \overset{l_{ij}}{\rightarrow} R_j$ and $R_i, R_j \in R$

Definition 3 (Cost Function): For a given link $l \in L$, $C(l) : L \rightarrow \mathcal{N}$. In network terminology it is called a metric.

Definition 4 (Packet-p): $< protocol : \{ip, tcp, udp\}$, $source - ip : IP,\ source - port : (0..65535)$, $destination - ip : IP,\ destination - port : (0..65535) >$, where IP $= \{a_1.a_2.a_3.a_4 : a_i \in (0..255),\ i \in \{1, 2, 3, 4\}\}$

Definition 5 (Filtering Function): Function $F_{l_{ij}}(p)$: $ACL \rightarrow boolean$ is the filtering function which evaluates packet p for the ACLs over link l_{ij}.

Formal representation of ACLs can be found in [6] and the cascaded ACLs along a path can be combined with ”AND” operation and each entry of an ACL will either permit a packet or deny a packet. Therefore the final result of the filtering functions can be evaluated as boolean value 1 (permit) or 0 (deny) [12].

B. Computing available paths

Different routing protocols use different algorithms to select the shortest path. Routing protocols cannot establish virtual paths without a physical connection. Therefore in our approach, as a first step the available physical paths are constructed then according to the routing protocol configured in the network, its path selection criteria is used to identify the best physical paths for communication.

Definition 6 (Path): Path π is a sequence of links and devices along the available physical connection between a source and a destination. Let R_0 be the source, and R_n be the destination of path π, then the k-th existing path between R_0 and R_n is defined as follows: $\pi^k_{<R_0,R_n>} = R_0 l_1 R_1 ....R_{i-1} l_i R_i........R_{n-1} l_n R_n$ such that $\forall i, l_i \in L, R_i \in R$ and $NH(R_{i-1}, l_i) = R_i$.

While identifying paths in the graph model, loops have to be eliminated. If $NH(R_{i-1}, l_i)$ is matching a device R_j which has been previously passed along a path π, then there is a loop on π. i.e., path $\pi = R_0 l_1 R_1 l_2 \ldots R_{n-1} l_n R_n$ has no loops if $\forall i, j : R_i = R_j$ or $l_i = l_j$ only when $i = j$. Cost over the path is $C(\pi) = C(l_1) + C(l_2) + \ldots + C(l_n)$, where $l_1, l_2, \ldots, l_n \in \pi$. Filtering function over a path for the packet p is $F_\pi(p) = F_{l_1}(p) \wedge F_{l_2}(p) \wedge \ldots \wedge F_{l_n}(p)$, where $l_1, l_2, \ldots, l_n \in \pi$. The result of the total filtering function is conjunction of filtering functions over links along the path π. The available paths and costs between two devices can be computed by repeating Dijkstra’s algorithm [2].

C. Cost function for RIP, OSPF and EIGRP

The proposed approach for modelling and analysis does not depend on the type of dynamic routing protocol in use. Only the computation of the cost function differs for each protocol as shown below. In our example we have considered OSPF as the configured protocol.

- *RIP* [1] – RIP has default link cost of value one and hence for any link l, $C(l) = 1$. Therefore the shortest path has the lowest hop count.
- *OSFP* [2] – OSPF requires the bandwidth function (BW) to compute the cost function, $BW(l_i) : L \rightarrow \mathcal{N}$, where $BW(l)$ is the bandwidth of the link l. The cost function for a link is defined by Cisco [13] as $C(l) = [10^8/BW(l)]$. The cost of the path $C(\pi)$ has the accumulated link costs along the path and least cost path will be used for the communication.
- *EIGRP* [3] – EIGRP uses delays and bandwidths of the whole path to calculate the cost function of the path. Delay function D is $D(l_i) : L \rightarrow \mathcal{N}$. Delay function for the path is defines as $D(\pi) = D(l_0) + D(l_1) + \ldots + D(l_n)$, where $l_0, l_1, \ldots, l_n \in \pi$. Bandwidth function BW is $BW(l_i) : L \rightarrow \mathcal{N}$. Bandwidth function for the path is defined as $BW(\pi) = BW(l_i)$ where $\forall l_i, l_j \in \pi, BW(l_i) \leq BW(l_j)$. Then the cost function for the path π is given by $C(\pi) = [10^8/BW(\pi) + D(\pi)]$.

III. Modified Topology Table

In general, the topology table keeps the physical interconnections of the network devices and links which will be used to build the routing tables. Using the routing tables we can check the communication properties such as reachability. The topology table is valid for a given network state and once the network state is changed due to link up/down, a new topology table needs to be computed. Routing table directs packets to the correct destination via the best route. Then routes receive updates from other routers and update the routing tables accordingly. This whole process can change communication paths. This is a time consuming process.

Therefore as a solution to this problem we introduce a modified topology table which is unique for a given network and invariant to the link changes unlike the routing tables. Using MTT, it is easy to derive topology for any network state, routing table for any network state, routing table for any source to any destination, available paths for any network state, available paths from any source to any destination, similar network states for a given topology, filters applied for any network state, filters applied for any source to any destination, costs of paths, critical links, refer Fig.2.

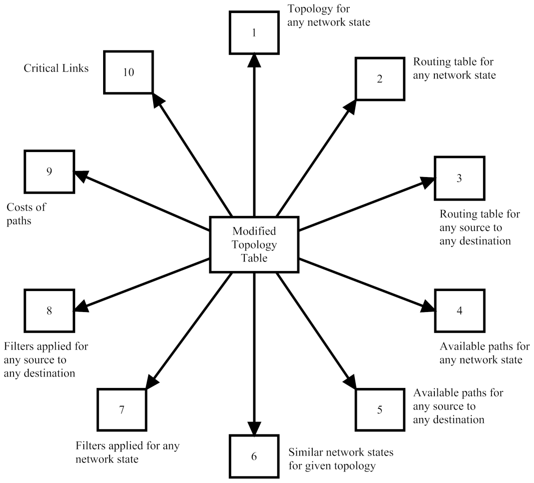

1
Topology for any network state
2
Routing table for any network state
3
Routing table for any source to any destination
4
Available paths for any network state
5
Available paths for any source to any destination
6
Similar network states for given topology
7
Filters applied for any network state
8
Filters applied for any source to any destination
9
Costs of paths
10
Critical Links
Modified Topology Table

Fig. 2. Features of Modified Topology Table

As described above, the MTT is capable of computing many useful information to predict dynamic behaviours. Another advantage of the MTT is that it does not need to recompute every time to analyse the properties under different network states.

A. Computing general state (GS)

This section briefs on computing the general state in MTT. General state is very important in terms of representing a network with few independent states as illustrated in Fig. 3 and eliminating the state space explosion during the model checking process. Further GS is used to check the reachability and security properties.

The *link state* for the i^{th} link is represented by c_i and it can have two values: 0 (link is down) or 1 (link is up).

Definition 7 (Function PrePath): Function $PrePath(\pi^k)$ is a function which returns path π^t between the same source and the destination which satisfies $C(\pi^t) \leq C(\pi^k)$ and $\nexists \pi^s \mid C(\pi^t) < C(\pi^s) < C(\pi^k)$ (no path between), where π^k is the k^{th} existing path between the source and the destination, and $k, s, t \in \mathcal{N}$.

When there are more paths with equal costs, further analysis are needed to *order* the paths. These algorithms are out of the scope of this paper and comparison of these methods are discussed in [14].

Definition 8 (Network State): Network state is a tuple $S =< c_1, c_2, \ldots,\ c_i, \ldots, c_m >$, where $\forall i, c_i$ represents the link state of $l_i \in L$, $i, m \in \mathcal{N}$, and m is the total number of links. In addition to the above link states one additional state X (link state is invariant to represent full state) is used to represent more link states by a single network state.

There can be many network states which satisfy a given path. So the representation of the r^{th} network state which satisfies path π^k between a source and a destination is $S_r^k =< c_1^r, c_2^r, \ldots, c_i^r, \ldots, c_m^r >, \forall\ l_i \in \pi^k,\ c_i^r = 1\ ,\ r \in \mathcal{N}$ and $\forall j, t,\ l_i \in \pi^j,\ \exists c_i^t = 0$ in $S_t^j =< c_1^t, c_2^t, \ldots, c_i^t, \ldots, c_m^t >$ where $t \in \mathcal{N}, \pi^j = PrePath(\pi^{j+1})$ and $0 \leq j < k$. In other words path π^k is used only when no other least cost path can be used.

Definition 9 (General State (GS)): This is defined for a specific path between a source and a destination. General state $\mathcal{S}_G^k = \{S_1^k, S_2^k, S_3^k, \ldots\}$ is a set of network states which satisfies k^{th} path π^k. The GS S_G^k is derived by processing the network states in S_G^{k-1} where k satisfies, $\pi^{k-1} = PrePath(\pi^k)$.

When there is a device or a link failure, the network topology and routing tables will be changed. By matching corresponding network state with general states, the reachability can be concluded easily.

One physical path is mapped to several link states, several link states are mapped to several network states and several network states are mapped to one general state. Fig.3 shows the mapping of path no.5 in the MTT (Table–I). Path no.5 is used for the communication between network A and C when path no.4 can not be used.

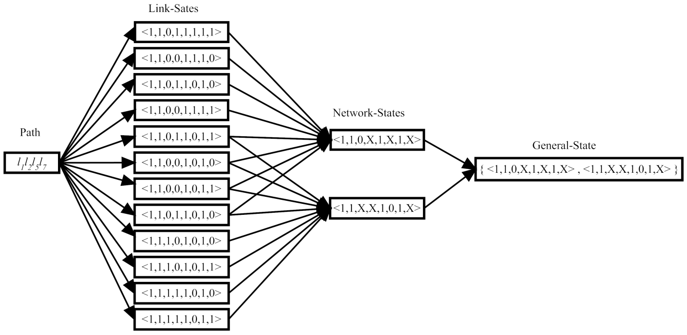


Fig. 3. Mapping between paths, link states, network states and general state

The following section explains state space reduction process by grouping the network states which have similar network behaviours.

1) **Computing general state S^0 for the first path π^0**
When all links are up path π^0 is used to send data over the network between a given source and a destination. The path π^0 is called the first path.
There can be many shortest paths in default configuration and considered they are ordered. To build the GS, we can order the equal cost paths in any order because it will not change the GS of other paths. When we change the order of the equal cost paths the GS will be interchanged among them. But order (in MTT) is required to identify the communication paths after the link failures. Since our main objective is to predict the validity of properties globally, the order of the equal cost paths are not significant for the analysis, e.g. checking a validity of security property in any network state.
Therefore in our analysis the general state S_G^0 for the first path is taken as a set with one element. $S_G^0 = \{< c_1^1, c_2^1, \ldots, c_i^1, \ldots, c_m^1 >\}$, where $\forall k,\ C(\pi^0) \leq C(\pi^k)$, and
    a) if $l_i \in \pi^0 : c_i := 1$

    b) Otherwise : $c_i := X$

2) **Computing general state S^{k+1} for path π^{k+1}**
   Assume the general state for the path π^k is $S_G^k = \{S_1^k, S_2^k, \ldots, S_r^k, \ldots, S_t^k\}$, where $t \in \mathcal{N}$, $1 \le r \le t$ and $\pi^k = PrePath(\pi^{k+1})$. Elements in S_G^{k+1}, $S_r^{k+1} =< c_1^{r'}, c_2^{r'}, \ldots, c_i^{r'}, \ldots, c_m^{r'} >$ are computed based on network state $S_r^k =< c_1^r, c_2^r, \ldots, c_i^r, \ldots, c_m^r >$ as follows: For $\forall i, l_i \in L$
    a) if $l_i \in \pi^{k+1}$ and $c_i^r = 0$ then ignore process of network state S_r^k and proceed with another r.
    b) if $l_i \in \pi^{k+1} : c_i^{r'} := 1$
    c) if $l_i \notin \pi^{k+1} and\ c_i^r = X : c_i^{r'} := X$
    d) if $l_i \notin \pi^{k+1} and\ c_i^r = 0 : c_i^{r'} := 0$
    e) if $l_i \notin \pi^{k+1} and\ c_i^r = 1$: Let $\{i \in< 0...m > |c_i^r = 1\ and\ l_i \notin \pi^{k+1}\} = \{r_1, r_2, ..., r_j, ..., r_q\}$ where $q \in \mathcal{N}$, $q < m$, m–total links and $1 \le j \le q$, then duplicate q times the state S_r^{k+1} as $S_{r_j}^{k+1} =< c_1^{r'_j}, c_2^{r'_j}, ..., c_{i-1}^{r'_j}, c_i^{r'_j}, c_{i+1}^{r'_j}, ..., c_m^{r'_j} >$, such that $c_i^{r'_j}$ is,
        i) if $c_i^{r'} = 0 : c_i^{r'_j} := 0$
        ii) if $c_i^{r'} = 1 : c_i^{r'_j} := 1$
        iii) if $c_i^{r'} = X : c_i^{r'_j} := X$
        iv) if $i = r'_j : c_i^{r'_j} := 0$
        v) Otherwise : $c_i^{r'_j} := X$
    f) Repeat (a) to (e) above for all r

Computing the general state for the first two paths between network A to C in our example, $\pi^0 = l_1 l_3 l_6 l_7$, $\pi^1 = l_1 l_2 l_5 l_7$ and $\pi^0 = PrePath(\pi^1)$ will be as below, (Refer the steps in section III-A–1 & 2)

1) Compute the general state for π^0
    a) $l_1, l_3, l_6, l_7 \in \pi^0$,
       $S_G^0 = \{S_1^0\} = \{< 1, , 1, , , 1, 1, >\}$
    b) $l_2, l_4, l_5, l_8 \notin \pi^0$,
       $S_G^0 = \{S_1^0\} = \{< 1, X, 1, X, X, 1, 1, X >\}$
2) Compute the general state for π^1
    a) no $c_i^1 = 0$ in S_G^0 , so no states to ignore in S_G^1
    b) $l_1, l_2, l_5, l_7 \in \pi^1$,
       $S_G^1 = \{S_1^1\} = \{< 1, 1, , , 1, , 1, >\}$
    c) $c_4^1 = X, c_8^1 = X$ in S_G^0, so $c_4^{1'} = X, c_8^{1'} = X$ in S_G^1, $S_G^1 = \{S_1^1\} = \{< 1, 1, , X, 1, , 1, X >\}$
    d) No $c_i^1 = 0$ in S_G^0 so no zero's copy to S_G^1
    e) $c_3^1 = 1, c_6^1 = 1$ in S_G^0 and $l_3, l_6 \notin \pi^1$, so $i = \{r_1, r_2\}$ where $r_1 = 3$ and $r_2 = 6$ duplicate S_1^1 as S_3^1 and S_6^1 where $S_G^1 = \{S_3^1, S_6^1\}$
        i) No $c_i^{1'} = 0$ in S_1^1 so no zero's copy to S_3^1 and S_6^1
        ii) $c_1^{1'} = 1, c_2^{1'} = 1, c_5^{1'} = 1, c_7^{1'} = 1$, in S_1^1 in S_1^1 so $c_1^{3'} = 1, c_2^{3'} = 1, c_5^{3'} = 1, c_7^{3'} = 1$, in S_3^1 and $c_1^{6'} = 1, c_2^{6'} = 1, c_5^{6'} = 1, c_7^{6'} = 1$, in S_6^1, $S_G^1 = \{S_3^1, S_6^1\} = \{< 1, 1, , , 1, , 1, >, < 1, 1, , , 1, , 1, >\}$
        iii) $c_4^{1'} = X, c_8^{1'} = X$, in S_1^1 so $c_4^{3'} = X, c_8^{3'} = X$, in S_3^1 and $c_4^{6'} = X, c_8^{6'} = X$, in S_6^1, $S_G^1 = \{S_3^1, S_6^1\} = \{< 1, 1, , X, 1, , 1, X >, < 1, 1, , X, 1, , 1, X >\}$
        iv) $i = \{r_1, r_2\}$, $r_1 = 3$ and $r_2 = 6$ so $c_3^{3'} = 0, c_6^{6'} = 0$, $S_G^1 = \{S_3^1, S_6^1\} = \{< 1, 1, 0, X, 1, , 1, X >, < 1, 1, , X, 1, 0, 1, X >\}$
        v) Mark remaining's with X,ie. $c_6^{3'} = X, c_3^{6'} = X$, $S_G^1 = \{S_3^1, S_6^1\} = \{< 1, 1, 0, X, 1, X, 1, X >, < 1, 1, X, X, 1, 0, 1, X >\}$

Like wise the general states for the other paths can be computed. Table–I contains the full MTT for our example network.

No	Source	Destination	General Sate	Cost	Path	Filter
1	192.168.1.0/24	192.168.2.0/24	1,1,X,X,X,X,X,1	4	$l_1 l_2 l_8$	permit ip any any
2	192.168.1.0/24	192.168.2.0/24	1,0,1,1,X,X,X,1	5	$l_1 l_3 l_4 l_8$	Deny ip 192.168.3.0/24 192.168.2.0/24
3	192.168.1.0/24	192.168.2.0/24	1,0,1,0,1,1,X,1	8	$l_1 l_3 l_6 l_5 l_8$	Deny ip any host 192.168.1.5 www
4	192.168.1.0/24	192.168.3.0/24	1,X,1,X,X,1,1,X	5	$l_1 l_3 l_6 l_7$	permit ip any any
5	192.168.1.0/24	192.168.3.0/24	1,1,0,X,1,X,1,X , 1,1,X,X,1,0,1,X	7	$l_1 l_2 l_5 l_7$	Deny ip any host 192.168.1.5 www
6	192.168.1.0/24	192.168.3.0/24	1,1,0,1,0,1,1,X	8	$l_1 l_2 l_4 l_6 l_7$	Deny ip 192.168.3.0/24 192.168.2.0/24
7	192.168.1.0/24	192.168.3.0/24	1,0,1,1,1,0,1,X	8	$l_1 l_3 l_4 l_5 l_7$	Deny ip host host 192.168.1.5 www OR Deny ip 192.168.3.0/24 192.168.2.0/2
8	192.168.2.0/24	192.168.1.0/24	1,1,X,X,X,X,X,1	4	$l_8 l_2 l_1$	permit ip any any
9	192.168.2.0/24	192.168.1.0/24	1,0,1,1,X,X,X,1	5	$l_8 l_4 l_3 l_1$	Deny ip 192.168.3.0/24 192.168.2.0/24
10	192.168.2.0/24	192.168.1.0/24	1,0,1,0,1,1,X,1	8	$l_8 l_5 l_6 l_3 l_1$	Deny ip any host 192.168.1.5 www
11	192.168.2.0/24	192.168.3.0/24	X,X,X,X,1,X,1,1	5	$l_8 l_5 l_7$	Deny ip anyhost 192.168.1.5 www
12	192.168.2.0/24	192.168.3.0/24	X,X,X,1,0,1,1,1	6	$l_8 l_4 l_6 l_7$	Deny ip 192.168.3.0/24 192.168.2.0/24
13	192.168.2.0/24	192.168.3.0/24	X,1,1,0,0,1,1,1	7	$l_8 l_2 l_3 l_6 l_7$	permit ip any any
14	192.168.3.0/24	192.168.1.0/24	1,X,1,X,X,1,1,X	5	$l_7 l_6 l_3 l_1$	permit ip any any
15	192.168.3.0/24	192.168.1.0/24	1,1,0,X,1,X,1,X , 1,1,X,X,1,0,1,X	7	$l_7 l_5 l_2 l_1$	Deny ip any host 192.168.1.5 www
16	192.168.3.0/24	192.168.1.0/24	1,1,0,1,0,1,1,X	8	$l_7 l_6 l_4 l_1$	Deny ip 192.168.3.0/24 192.168.2.0/24
17	192.168.3.0/24	192.168.1.0/24	1,0,1,1,1,0,1,X	8	$l_7 l_5 l_4 l_3 l_1$	Deny ip any host 192.168.1.5 www OR Deny ip 192.168.3.0/24 192.168.2.0/2
18	192.168.3.0/24	192.168.2.0/24	X,X,X,X,1,X,1,1	5	$l_7 l_5 l_8$	Deny ip host host 192.168.1.5 www
19	192.168.3.0/24	192.168.2.0/24	X,X,X,1,0,1,1,1	6	$l_7 l_6 l_4 l_8$	Deny ip 192.168.3.0/24 192.168.2.0/24
20	192.168.3.0/24	192.168.2.0/24	X,1,1,0,0,1,1,1	7	$l_7 l_6 l_3 l_2 l_8$	permit ip any any

TABLE I
MODIFIED TOPOLOGY TABLE

IV. REACHABILITY ANALYSIS

This section describes the process of reachability and security property analysis using the above defined formal model.

Now, we extend our definition of the path to an evaluated path. Let $P = \{\pi, C(\pi), F_\pi(p)\}$ be an *evaluated path*, where $C(\pi)$ is the cost function and $F_\pi(p)$ is the filtering function over path π and packet p for a given source and destination.

Definition 10 (Critical Points CP): CP is a subset of devices and links which present along every possible path. The links and devices of CP are essential for communication. CP is defined as follow. $CP = \{R^c, L^c\}$, where R^c represents critical devices and L^c represents the critical links such that $R^c = \{r \mid \forall \pi \in P : r \in \pi\}$, and $L^c = \{l \mid \forall \pi \in P : l \in \pi\}$.

Definition 11 (Universal Points UP): UP=$\{R^u, L^u\}$ is links and devices which have no effect on topology changes and hence no effect on communication. UP is defined using complementary sets R' and L'. Let R' and L' be sets such that $R' = \{r \mid \exists \pi \in P : r \in \pi\}$, and $L' = \{l \mid \exists \pi \in P : l \in \pi\}$. The universal devices and links are complement sets to R', and, L' i.e. $R^u = R - R'$, and $L^u = L - L'$.

Let us assume a set of failed devices R^f and failed links L^f. Using sets CP and UP, network reachability can be verified as,

1) If $\exists r \in R^f$ such that $r \in R^c$ or $\exists l \in L^f$ such that $l \in L^c$, then there is no available path, i.e., the destination is not reachable.
2) If $R^f \subseteq R^u$ and $L^f \subseteq L^u$, then topology is not changed.
3) Otherwise, MTT will be used to predict reachability and security properties.

First two border cases, (1) and (2) are easy to find and eliminate the obvious cases. The integration of the above two cases into a simulation tool and comparison of evaluated results were discussed in our previous paper [15]. The most interesting and the difficult case is (3) which uses MTT for prediction.

To predict reachability under a given network state, the format of the MTT and the reachability property should be formally defined.

MTT is a set which consists entries as elements, ie. MTT = $\{$entry1, entry2, entry3, ... $\}$ and an entry is a sequence of $< source - ip : IP, destination - ip : IP, GS, Cost, Path, ACL >$ (first column in MTT, No. is used for explanation purposes)

Reachability Property – is a combination of an ip packet and a network state which needs to be evaluated.

To predict reachability property from MTT, reachability property should be matched with the source–ip, destination–ip, network state (with GS) and the filtering rule in MTT. If no match is found, then there is no reachability under the given network state. We need to introduce one additional general states to MTT for representing all the unreachable network states and will be denoted by S_∞.

E.g. Predict the web service reachability from 192.168.3.80 to 192.168.168.1.5, when the links l_3 and l_4 are down.

As per the definition of the Reachability property – $< ip, 192.168.3.80, any, 192.168.168.1.5, www >$ should be checked under the Network State – $< X, X, 0, 0, X, X, X, X >$.

Matching MTT entry for predicting the reachability will be as below

1) First source–ip will be matched in MTT, it will start from entry No. 1 and stop at No. 14 to check next field of MTT (MTT is ordered by source–ip, destination–ip using *PrePath* fuction)
2) Then destination–ip will be matched in MTT, then it will start from No. 14 to check next field of MTT
3) After that the network state will be matched with the general state in MTT, it will select first match which is No. 15, then starts checking next fields of MTT
4) Finally the filter gives output of false, so even the server is reachable (there exists a communication path) the service is unreachable (service port is blocked by the filtering rules)

For matching the source–ip and the destination–ip in the MTT is complicated and required to use interval matching technique to improve the performance. To overcome this problem we will be using the method described in paper [6] which is the Interval Decision Diagrams (IDD's) [9] in our automation stage.

In general the MTT matching process is checking of a property (reachability property) over a model (MTT + given network state) therefore Model Checking techniques can be used to automate the above process. Our future work consists of building the transition model and incorporating model checking techniques for the above verification.

V. Results and discussions

The formal model, MTT is able to predict reachability under any given network state. We can compare our approach with on-demand approach mentioned in the state of the art which uses pre-computed routing tables to predict the reachability.

Method-1 : Pre-computing the routing table for each state and analyse the reachability, This will have following iterations,
(2^m link-states) × (r routers) × (n routing entries in each router (n networks)) × (Complexity of generating one routing entry) $= 2^m \times r \times n \times O(building\ routing\ entry)$
Iteration(Method-1)= $2^m \times r \times n$

Method-2 : Computing the available paths from each source to each destination (which is described by this paper). This will have following iterations,
n routing entries (n networks) × (n-1 destinations) × (q average physical paths) × (Complexity of generating one path) $= n \times (n-1) \times q \times O(building\ a\ path)$
Iteration(Method-2)= $n \times (n-1) \times q$

The complexity of both methods are dependent on the degree of the mesh network. In a full mesh network, there is no significant difference on the iterations. But there is a huge improvement on method-2 with the decrease of the degree of the mesh network. Proper theoretical evaluation of the complexity will be done in our future work.

To check the complexity we have applied both methods to the topology of Czech Academic Network (CESNET). For CESNET, r=23, m=26, n=23 and q=20 [15]. Iteration(method-1) = $2^{26} \times 23 \times 23 = 35500589056$, Iteration(Method-2) = $23 \times 22 \times 20 = 10120$. For our example network in Fig.1, n=3, m=8, r=4 and q=4; Iteration(Method-1) = $2^8 \times 8 \times 3 = 6144$, and Iteration(Method-2) = $3 \times 2 \times 4 = 24$ considering average path as four, but for our topology actual value is 20 iterations (See Table–I). Generally $O(building\ a\ routing\ entry) > O(building\ a\ path)$, it can be seen that method-2 is much more efficient than method-1.

Scalability of the above approach was examined by using our university network. Our model predicted the reachability and security properties which are matching with the expected results. Also a ring lattice type network was simulated by increasing the number of nodes and measured the generated paths and the time consumption. Testing was done in 2xDual Xeon Core2/3GHz with 8 GB RAM machine and the performance results are in Table–II.

Nodes	1–16	17	18	19	20	21	22	23	24
Paths	5–7502	13776	25314	46534	85562	157344	289370	532202	978838
Time(Sec)	01	02	03	04	08	15	31	55	95

TABLE II
TIME CONSUMPTION FOR BUILDING MTT

One weakness of this approach is that this model does not give any information on the transitional behaviours. We are planning to introduce a transition model to overcome this drawback. Also the model does not contain any second level link failures due to loads after the network convergence from the first set of links failures.

VI. CONCLUSION

The described approach is very effective on analysing network reachability and security properties in dynamic networks without computing routing tables for each network states. It's shown that the network reachability and security properties under any network state can be easily predicted by using MTT. The method used to reduce the state space by grouping the states which have similar network behaviours is very effective than working on all available link states. In comparison with the other formal models used for the model checking process this model has key features such as capability of predicting behaviours of dynamic networks and security properties without rebuilding the routing tables for each network state.

We have shown how MTT is used to identify the communication paths after link failures. The reachability analysis is one feature of the MTT. We have outlined on the evaluation the security property from MTT. To evaluate validity of the security properties globally, the analysis process should be integrated into a model checking tool. MTT contains complete state space and communication paths a network can have, therefore MTT enables to predict validity of properties globally. This concludes that MTT is a global representation of a network and can be used as an input for many other analysis.

VII. FUTURE WORK

Our future work is focussed on modelling and analysing of complex networks which include different combination of routing protocols. In modelling, we are planning to implement a framework to model the device configurations and to improve the transition model. Then to use a model checking tool to check the soundness of the Cisco configuration files and automate the verification process.

Another area is to optimise the algorithms to improve the efficiency. This should be done in the model checking process. Once we have the optimised algorithm, we need to evaluate the computational complexity of our method.

Finally the proposed technique will be integrated into the simulation tool OMNeT++. Our goal is to automatically convert Cisco router configuration files into our model and analyse the security and reachability properties by using verification techniques.

VIII. ACKNOWLEDGEMENT

The research is supported by the Czech Ministry of Education in frame of the Research Intention MSM 0021630528: Security-Oriented Research in Information Technology.

REFERENCES

[1] C.L.Hedrick, "*Routing Information Protocol*," RFC 1058, June 1988.

[2] J. Moy, "*OSPF Version 2*," RFC 2328, April 1998.

[3] J.Doyle, *CCIE Professional Development Routing TCP/IP*. Cisco Systems, Inc., 2006, vol. 1.

[4] A. Avizienis, J. Laprie, B. Randell, and C. Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing," *IEEE Transactions on Dependable and secure Computing*, vol. 1, no. 1, pp. 11–33, 2004.

[5] *Configuring IP Access Lists*, White papers 23602 ed., Cisco, July 2007.

[6] P. Matoušek, J. Ráb, O. Ryšavý, and M. Švéda, "A formal model for network-wide security analysis," in *15th IEEE Symposium and Workshop on ECBS*, 2008.

[7] G. G. X. et., "On static reachability analysis of ip networks," in *INFOCOM*, 2005, pp. 2170–2183.

[8] D. Antoš, "Hardware-constrained Packet Classification," Ph.D. dissertation, Masaryk University, 2006.

[9] M. Christiansen and E. Fleury, "An Interval Decision Diagram Based Firewall," in *3rd International Conference on Networking (ICN'04)*. IEEE, Feb. 2004.

[10] Q. Gan, Bjarne, and E. Helvik, "Dependability modelling and analysis of networks as taking routing and traffic into account," 2006.

[11] R. M. Michael Menth, Michael Duelli and J. Milbrandt, "Resilience analysis of packet-switched communication networks," *IEEE/ACM Transactions on Networking*, vol. 17, no. 6, 2009.

[12] J. Guttman, "Filtering Postures: Local Enforcement for Global Policies," in *IEEE Symposium on Security and Privacy*, 1997, pp. 120–129.

[13] "Ospf design guide," *Cisco Systems*, 2006, available at
URL: http://www.cisco.com/warp/public/104/1.pdf.

[14] M. M. David Hock, Matthias Hartmann and C. Schwartz, "Optimizing unique shortest paths for resilient routing and fast reroute in ip-based networks," in *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS)*, 2010.

[15] P. Matouek, O. Ryav, G. de Silva, and M. Danko, "Combination of simulation and formal methods to analyse network survivability," in *Proceedings of the IEEE 3rd International ICST Conference on Simulation Tools and Techniques*. International Communication Sciences and Technology Association, 2010, p. 6.