Proceedings of the 2014 Federated Conference on Computer Science and Information Systems pp. 947–954
DOI: 10.15439/2014F159
ACSIS, Vol. 2

The procedure for monitoring and maintaining a network of distributed resources

Tomasz Malinowski
Military University of Technology,
Faculty of Cybernetics
ul. Kaliskiego 2, 00-902 Warsaw,
Poland,
Email: tmalin@ita.wat.edu.pl

Artur Arciuch
Military University of Technology,
Faculty of Cybernetics
ul. Kaliskiego 2, 00-902 Warsaw,
Poland,
Email: a.arciuch@ita.wat.edu.pl

Abstract—**In this article, we propose an algorithm for the management of the structure of the tunnel connections in large, distributed nature, corporate network. Management is here understood as monitoring the availability and health of key elements and dynamic reconfiguring the system in situations of failure symptoms. It was assumed that the algorithm can be used in cases where the correct functioning of the network (providing the appropriate level of quality of service) means an access to distributed resources, usually redundant, which may be subject to reallocation. It was shown that developed algorithm may be used for maintenance of VPN network with dynamic tunnels (DMVPN).**

I. Introduction

In present-day's communication networks, which are networks of different services integration and networks giving access to a distributed resources, the efforts of the engineers, protocols and network hardware designers, and in particular, the information and communication systems administrators, focus on ensuring high-reliability of the system, continuous availability of network resources and the desired level of quality of services. Efficient and effective diagnosis of inappropriate states (incompatible with expected) of system functioning is unquestionably the most important challenge for administrators of complex ICT environments.

The algorithm used to oversee the operation and dynamic reconfiguration of the system of dynamic tunnels, raised between border routers of company's branches networks was proposed. It's not too hard to indicate many examples of systems, in which the continuous and reliable access to distributed network resources (for example resources within the cloud computing system, within distributed data centres, etc.) seems to be critical (especially important) and for which different optimizing algorithms of structure of connections and reallocating resources of servers and network nodes, with the requirement of achieving state of convergence as soon as possible are proposed [8], [9], [10].

The proposed algorithm for system of dynamic tunnels is based on some elements of system-level diagnostics and self-diagnosable systems [1] - [7], [13], [14].

Corporate network of dynamic tunnels (DMVPN - Dynamic Multipoint Virtual Private Network) [11] is seen by authors as a client-server system in which a significant problem is to provide reliable client-server communication, where server plays role of the tunnel broker, and reliable client-client communication through dynamically created tunnels.

The DMVPN network, from the point of view of the authors, contains certain number of clients, DMVP servers and, introduced by authors, the management station. Transmission links (tunnels) between clients and servers form a logical structure of connection links and the primary function of tunnel brokers (servers) is to provide communication between network clients by informing clients about tunnel parameters. Tunnels between clients are rise dynamically on transmission time only, so the logical structure is subject to continuous modification.

It is assumed that the client can forward the query about parameters of tunnel leading to another client to one of the many assigned (known) servers (it's system with tunnel broker redundancy).There are the primary server and backup servers in the group of servers. In the absence of the ability to provide client-server communication, or in case of communication parameters deterioration, the client should have the possibility to appeal (to send service order) to the other server from a server group. In addition, the client should send inaccessibility notification to the management stations (suspicion of primary or/and backup server failure notification). Quick server failure detection will be possible through periodic availability testing of the servers, that are assigned to the client.

Management station should have the possibility of reconfiguring logical network connections, which means the ability to assign clients to servers (primary and backup).It is assumed that the servers will have the opportunity to test the quality of the connections to clients (source to destination latency, packet loss, jitter, etc.), and the result of testing will have an impact on the allocation of clients to the server.

Quickly responding to the unavailability of tunnel brokers (detection of unfit brokers) will be carried out using IP SLA probes [12], available within operating systems of network devices, automating the login process (telnet/ssh) from the

978-83-60810-58-3/$25.00

management station to network nodes (clients and servers) and changing their configuration.

Authors propose the procedure of dynamic tunnels network management based on linking the utility functions of network devices with diagnostic functions. Methods of connecting utility functions with diagnostic functions belong to the system diagnostic methods (system-level diagnosis), and systems capable to indicate unsuitable items and reconfiguration (auto-repair) belong to class of self-diagnosable systems.

II. Characteristics and Structures of DMVPN Network

DMVPN is a technology that allows to build scalable VPN, combining the advantages of GRE (Multipoint GRE), IPSec and NHRP (Next Hop Resolution Protocol). DMVPN connects branch offices through the Internet (WAN) infrastructure based on tunneling and assuming that the tunnels will be "lifted" dynamically, as needed, and so will not be required to create persistent connections and, worse still, arranging them in a structure of type "full-mesh". The technology is based on a logical topology of a star, with the highlighted nodes called Hub and Spoke. In the basic configuration, DMVPN offers communication between branches through Central Branch (Spoke-Hub-Spoke connections) and in the enlarged configuration gives the ability to directly connect branch office networks with Spoke routers (Spoke-to-Spoke connections).

The general shape of the corporate network with dynamic tunnels was shown in figure 2.

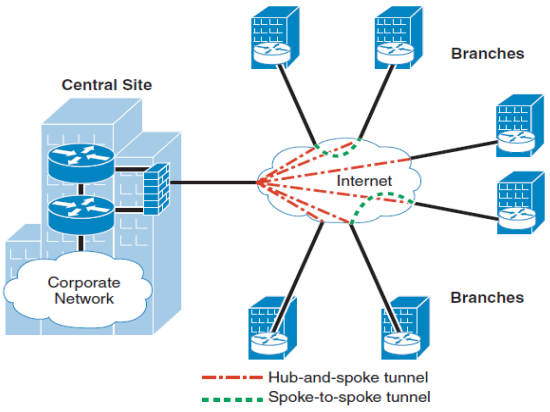


Fig. 2 Overall shape of DMVPN network[11]

Implementations of Dynamic Multipoint VPN networks

A particularly important feature of DMVPN is the ability to dynamic "lifting" tunnels, which encapsulate packets of any type (unicast, multicast, broadcast), IPv4, IPv6, and others. It follows directly from the application of GRE (Generic Routing Encapsulation) protocol. So, for example, DMVP can be seen as one of the transitional mechanisms for connecting IPv6-capable networks through IPv4-based network infrastructure. Importantly, packets transmission in dynamic GRE tunnels can be secured by IPSec, and one tunnel interface supports multiple IPSec sessions. It is also worth noting that due to the possibility of organizing tunnels between branch offices, although a little bit difficult, configuration of QoS policies, for example in system with VoIP calls, becomes more transparent.

DMVPN network in basic configuration was presented in figure 3.

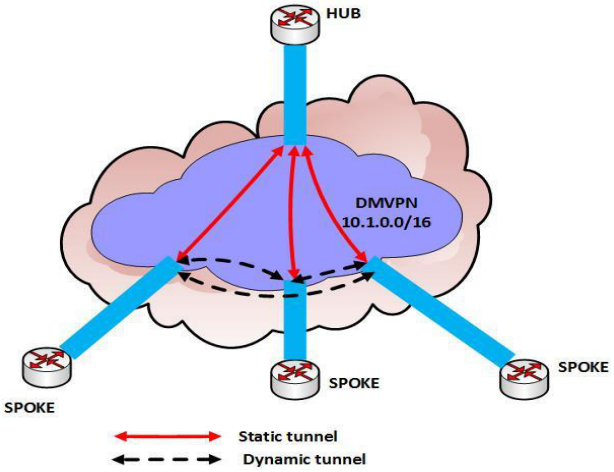


Fig. 3 DMVP Network with one Hub router

The router called Hub operates within the Central Branch network, while in remote branches act Spoke routers (Hub's clients). Remote branches have a persistent tunnel connection to Central Branch and access to Central Branch's network resources (static tunnel between a Spoke-Hub pair). In the case of realizing transmission between remote branch offices, transmission is preceded by sending an NHRP (Next Hop Resolution Protocol) initiator Spoke's request for tunnel address of Spoke router in a distant location. The Hub in a central location answers to NHRP query and is called NHS Server (Next-Hop Server), and NHRP protocol allows to get information about actual addresses of Spoke's interfaces. Tunnel connection between the branch offices can be implemented via the Hub, or it can be made a direct tunnel between Spokes. The main requirement of tunnel creation between the branches (direct or via the Hub) is Spoke's registration in NHS server and cyclic refreshing of registration. Hub's and Spoke's configuration details are discussed in the technical documentation [11] and will not be discussed here.

In the extended configuration (as in figures 4 and 5), to increase the degree of reliability of the network, additional Hub nodes are implemented or another DMVPN network is created.

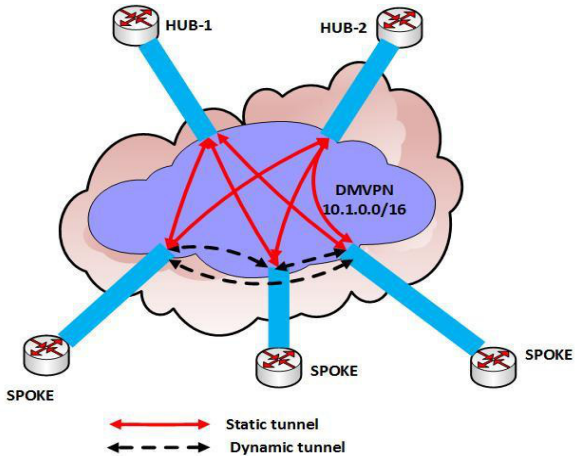


Fig. 4 DMVPN network with backup NHS server

In the case of single DMVPN cloud (figure 4), each Spoke uses a single mGRE tunnel leading to two or more Hubs, acting as the NHS servers.

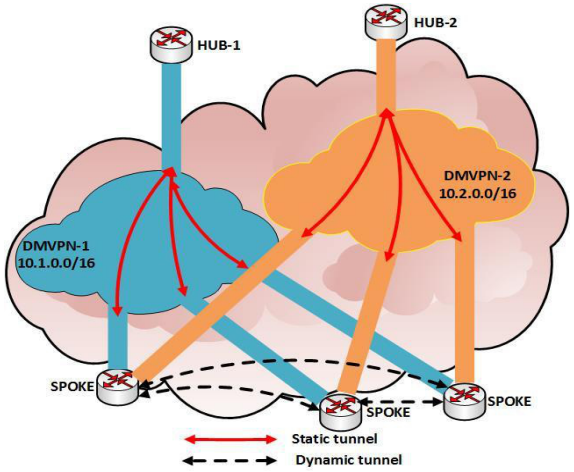


Fig. 5 Dual-cloud DMVPN

In dual-cloud DMVPN network (figure 5), each Spoke uses two tunnels, leading to two different Hubs.

Sample problems in DMVPN networks

As mentioned, the basic condition of successful tunneling between the different locations is proper functioning of the Spoke in Hub signing up mechanism, the polling mechanism about physical addresses of the final points of tunnels, but also disconnecting calls in case of failure (for example in case of demise Hub's network interface, its temporary loss, too much load of the Hub, too long a response time, etc.).

In some cases, such as realization of tunnel connections protected by IPSec, it is necessary to continuous monitoring Hub's availability and removing IPSec session after a period of temporary unavailability of the Hub (cleaning of IPSec Security Associations). In case of unavailability or even reduce the effectiveness of primary Hub, handling the entire NHRP process should be taken over by another (secondary) Hub.

DMVPN technology is refined through the years, but it's still possible to indicate some scenarios in which an administrator's intervention is necessary.

An illustration of problems appearing in the network DMVPN (besides the obvious problem of the physical unavailability of critical nodes) may be the following example with IPSec tunnels. In a simulation environment, DMVPN network with structure shown in Figure 6, has been configured.

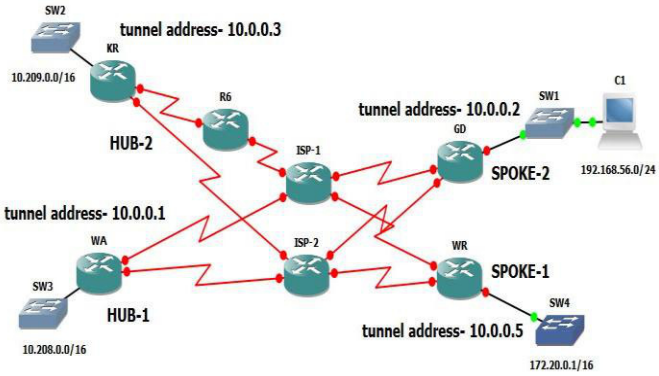


Fig. 6 Simulation environment of DMVPN network

In accordance with the assumption of DMVPN network, SPOKE-1 and SPOKE-2 maintain IPSec tunnels with theirs primary HUB-1. The sample configuration of Spoke's tunnel is as follows (for SPOKE-1):

```
interface Tunnel0
 bandwidth 1024
 ip address 10.0.0.5 255.255.255.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication dmvpn
 ip nhrp map 10.0.0.1 1.0.0.1
 ip nhrp map multicast 1.0.0.1
 ip nhrp network-id 99
 ip nhrp holdtime 300
 ip nhrp nhs 10.0.0.1
 ip tcp adjust-mss 1360
 delay 1000
 tunnel source Serial0/0
 tunnel mode gre multipoint
 tunnel key 232323
 tunnel protection ipsec profile dmvpn
```

Active IPSec connection can be listed after the commands: show dmvpn and show crypto session, as below (for SPOKE-1).

```
WR#show dmvpn
Legend: Attrb --> S - Static, D - Dynamic, I - Incompletea
      N - NATed, L - Local, X - No Socket
      # Ent --> Number of NHRP entries with same NBMA peer

Tunnel0, Type:Spoke, NHRP Peers:1,
 # Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb
 ----- --------------- --------------- ----- -------- -----
    1         1.0.0.1         10.0.0.1    UP 00:01:54 S
```

```
WR#show crypto session
Crypto session current status

Interface: Tunnel0
Session status: UP-ACTIVE
Peer: 1.0.0.1 port 500
  IKE SA: local 3.0.0.1/500 remote 1.0.0.1/500 Active
  IPSEC FLOW: permit 47 host 3.0.0.1 host 1.0.0.1
   Active SAs: 2, origin: crypto map
```

It's easy to demonstrate that a temporary loss of connection between Spoke and Hub, causes transmission rupture between two Spokes.

The following sequence of events is an illustration of this problem:

```
WR#ping 192.168.56.2 repeat 10000000

Type escape sequence to abort.
Sending 10000000, 100-byte ICMP Echos to 192.168.56.2, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!.....
```

In this moment, the temporary collapse of the HUB-*1's interface* occurs

```
*Mar  1 00:07:23.679: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor 10.0.0.1 (Tunnel0) is down: holding time expired.....................................................
.....................................................................
Success rate is 37 percent (79/209), round-trip min/avg/max = 156/365/844 ms
```

When we observe messages on HUB-1's console, just after lifting the interface, we see syslog messages, announcing incorrect SPI IPSec session ID.

```
WA#
*Mar  1 00:08:31.703: %CRYPTO-4-RECVD_PKT_INV_SPI: decaps: rec'd IPSEC packet has invalid spi for destaddr=1.0.0.1, prot=50, spi=0x741CDD33(1948048691), srcaddr =3.0.0.1

*Mar  1 00:09:32.139: %CRYPTO-4-RECVD_PKT_INV_SPI: decaps: rec'd IPSEC packet has invalid spi for destaddr=1.0.0.1, prot=50, spi=0x1085029(17322025), srcaddr=2.0.0.1
```

Unfortunately, the remedy in this case is "manual" breaking IPSec session, as illustrated below.

```
WR#clear crypto session
*Mar  1 00:11:59.223: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor 10.0.0.1 (Tunnel0) is up: new adjacency

WR#ping 192.168.56.2 repeat 10000

Type escape sequence to abort.
Sending 10000, 100-byte ICMP Echos to 192.168.56.2, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!.
Success rate is 94 percent (18/19), round-trip min/avg/max = 224/411/624 ms
```

The case discussed above helped the authors to test the procedure of monitoring and reconfiguration of the nodes (routers, Hubs and Spokes) and it should be noted that in multiple IPSec implementation, including Cisco device's implementation, a mechanism for detecting unreachability of nodes communicating by cryptographic tunnels and resolving the problem of "dead" IPSec session, known as the Dead Peer Detection, was introduced.

Beside the mentioned case, problems caused by the failure of nodes and network links were identified. Scenarios in which reconfiguring the nodes was necessary (because of the failure of the primary route leading to the HUB) were considered. The necessary reconfiguration included changes of static routing paths in the routing table and ensuring availability of one of two NHS servers by the Spoke node.

III. General Model of Self-Diagnosable Client-Server System

The procedure for oversee a network of dynamic tunnels, presented in chapter IV, allows to treat a managed DMVPN network as a client-server and a self-diagnosable system ([3], [4], [7]).

The self-diagnosable system determines the structure of mutual testing of elements, the method of using tests results, and the model of inference, based on the results of tests, about the reliability state of the system [3]. The structure of self-diagnosable system is described using a testing graph. Depending on the method for interpreting the results of tests, distributed systems and centralized systems can be distinguished. Inference about the state of the distributed system take place on the basis of results of parts of all the results of tests. Inference about the state of the centralized system take place on the basis of results of all the results of tests. In addition, stands out heterogeneous and homogeneous systems. Due to the reasoning model, which defines the relationship between results of tests and the reliability state of system, in centralized systems stands out PMC model [4] and BGM model [5], in distributed systems stands out MM and MM* models [1],[2].

In the self-diagnosable systems inference about the state of the system is implemented under certain conditions and on the basis of the results of tests obtained by fault-free and faulty elements. One of the specific conditions necessary for the reasoning about the reliability state of the system is the requirement for the maximum number of faulty elements of the system (called diagnosability) within a given number of all elements of the system. The diagnosability of the system is defined as the maximum number t, such that the system is t-diagnosable as long as the number of the faulty elements is not greater than t.

If a testing graph of a self-diagnosable system is a such edge induced subgraph of the system, which describes the t-diagnosable system, has minimal number of tests, then is called t-optimal testing graph of the t-diagnosable system. The t-diagnosable system has an irreducible testing graph if none of its edge induced subgraph does not describe a testing graph of t-diagnosable system. An irreducible testing graph that is not t-optimal is a t-quasi-optimal testing graph.

Sometimes, the system design must take into account that the costs of mutual testing of elements are not the same. If the arcs of the testing graph have assigned a generalized cost of testing, then such a graph can be called economic testing graph. The edge induced subgraph, that describes t-diagnosable system for which the generalized cost of testing shall give the minimum value, describes the cheapest t-diagnosable system. The t-diagnosable system, in the general case, may has several (many) t-optimal testing graphs. The cheapest testing graph is generally one of the t-optimal or t-quasi-optimal testing graphs.

General model of self-diagnosable client-server system

Assume that an organization has a computer network O, $O = \langle V, E\rangle$, of a certain logical structure, in which V denotes a set of computers ($v \in V$), E – a set of communication links ($v', v'') \in E \wedge v'' \in V(v') \wedge v' \in V(v'')$, where V(v) denotes set of nodes adjacent to node v. In the network O stands out computers of client type $k \in K$, computers of server type $s \in S$ and a computer of manager type z. It is assumed, for simplicity, that the manager is reliable. Also $(V = K \cup S \cup \{z\}) \wedge ((K \cup S) \cap \{z\} = \emptyset)$. A client k has assigned an ordered pair of servers $k(S) = (s', s'') \in S$, where s' is a primary server and s'' is a backup server. A client communicates with the server to invoke a given service on a server. A client also sends diagnostic messages (so-called traps) to the manager z. Similarly, each server sends traps to the manager z. The manager z stores information about logical structure of network O and about the status of clients and servers. Each server s stores information about adjacent clients: $s(K) = \{k \mid k \in S(s)\}$.

Set S of servers is a such subgraph of O, $\langle S\rangle_V$, that it is a testing graph of the PMC model.

It is known that if the system with |V| nodes is t-diagnosable for the PMC model, then [4]:

$$(|V| \geq 2t + 1) \wedge (\forall(v \in V) \mid \mu^-(v) \geq t), \qquad (1)$$

where $\mu^-(v)$ is indegree of node v.

A system with |V| nodes that satisfies the formula (1) is t-diagnosable if and only if [7]:

$$((\forall(0 \leq p \leq t-1) \wedge \forall(V' \subset V) \mid |V'| = |V| - 2t + p)) \Rightarrow |\Gamma(V')| > p, \qquad (2)$$

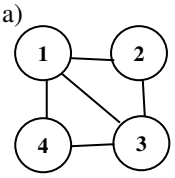
where $\Gamma(V') = \{v \mid \exists(v' \in V')\ v \in V'(v') \wedge v \notin V'$ means set of successors of elements of a set V', and $v \notin V'$.

Assume that subgraph $\langle S\rangle_V$ is described by a testing graph for 1-diagnosable system for the PMC model - it is assumed that the probability of damage of more than one server at the same time is low. From the formula (1) follows that $|S| \geq 3$. It is known that strongly connected graph, which has at least 3 nodes, is a testing graph for 1-diagnosable system of the PMC type.

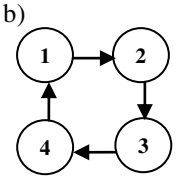
The figure 1 shows an example of a testing graph (1b) for subnet of servers, which has a logical structure like in figure 1a. A such pattern of test d_{st} where server s is testing server t was shown in figure 1c. Symbols: n(e), $n_0(e)$ and N = [n(4), n(3), n(2), n(1)], n(i) = x, i = 1,...,4, $x \in \{0,1\}$ denote: a reliability state of node e, state where node e is fault-free and vector that describes a reliability states of system, wherein, if n(i) = 0, then the i-th node is fault-free. If N = [0001], then result of test [d_{12}, d_{23}, d_{34}, d_{41}] can be [0001] else [1001], which corresponds to the pattern [x001].

In PMC model all tests are performed between two adjacent nodes, and it was assumed that a test result is reliable (respectively, unreliable) if the node that initiates the test is fault-free (respectively, faulty).

a)



b)



c)

$$[n(e_s) = n_0(e_s)] \rightarrow \left[d_{st} = \begin{cases} 0 \text{ dla } n(e_t) = n_0(e_t) \\ 1 \text{ dla } n(e_t) \neq n_0(e_t) \end{cases} \right]$$

$$[n(e_s) \neq n_0(e_s)] \rightarrow [d_{st} = x],\ x \in \{0,1\},$$

d)

N \ d_{st}	⟨1,2⟩	⟨2,3⟩	⟨3,4⟩	⟨4,1⟩
[0000]	0	0	0	0
[0001]	x	0	0	1
[0010]	1	x	0	0
[0100]	0	1	x	0
[1000]	0	0	1	0

Fig. 1 The PMC model illustration

IV. Algorithm for Monitoring and Maintaining DMVPN Networks

This chapter presents a general algorithm for oversee client-server networks, applied in the construction of self-diagnosable DMVPN network.

The fundamental objectives of proposed algorithm are as follows:

a) ensuring high reliability of the system on the basis of the continuous servers availability. We assume that the client should be able to communicate all the time with one/two servers. At the entrance, each client has assigned two servers, from a pool of servers (initial pool consists at least 3 servers to provide 1-diagnosability using the PMC model). In case of failure (unavailability) of one of the server, reconfiguration of the client is needed. Reconfiguration involves interfering in the configuration file and changes the settings of tunnel interfaces, which should indicate a new set of servers. So, also reconfiguration of the set of servers is needed.

b) shortening the convergence time of the system, which can be, in the opinion of the authors, achieved (in dynamic routing based and timers controlled DMVPN network) by forcing fast reconfiguration of nodes (servers and clients).

c) servers load balancing, which means that servers support nearly equal (comparable) number of clients.

We assume that the DMVPN network of an organization has a logical structure (figure 7) described by a graph $O = \langle V, E\rangle \mid (V = K \cup S \cup \{z\}) \wedge ((K \cup S) \cap \{z\} = \emptyset)$, where S denotes a set of servers (Hubs), K denotes a set of clients (Spokes) and z is reliable manager of the network. Servers $s \in S$ form a testing graph for the PMC 1-diagnosable system, The testing graph has a logical structure which is the Hamiltonian cycle, which has a directed Hamiltonian cycle. As mentioned earlier, a client k communicates with its servers $k(S) = (s', s'')$, $s', s'' \in S$, where *s'* is a primary server and *s''* is a backup server. A client maintains constant tunnel with its servers and asks servers (primary at first turn) about tunnel parameters leading to other clients. Each client has the ability to send diagnostic messages about fault detection (inability to communicate with server) to the manager z. Similarly, each server can inform about problems within the set of servers by sending traps to the manager z. The manager z knows logical structure of network O and the role played by each network node (client or server). Each server s has a specific set of assigned clients $s(K) = \{k \mid k \in S(s)\}$.

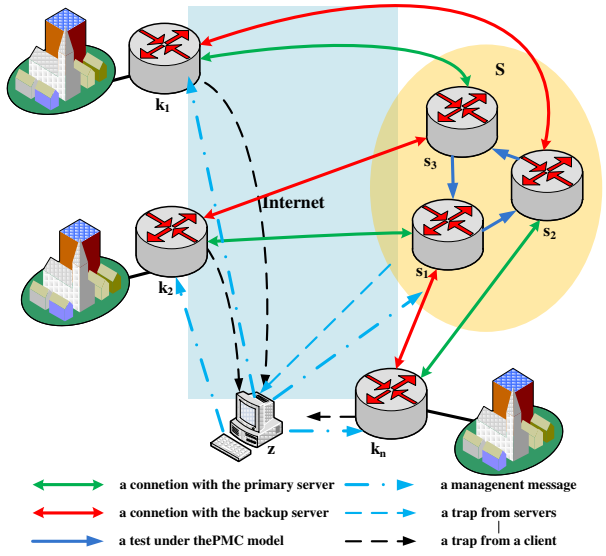


Fig. 7 Logical structure of connections and relations within DMVPN network

We assume that DMVPN system is running and has initial configuration, which means that each server supports some set of clients and the set of servers contains at least 3 servers. The set of potential servers (subset of all nodes) is also known.

The algorithm for monitoring and maintaining consists of the following steps:

1) The client k to perform a task (establish communication with the other client/server), invokes a service on a primary server s'. If the server s' can not do the job then:
   a) the client k invokes a service on a backup server *s''*,
   b) the client k generates the trap informing about server *s'* unavailability, which is sent to the management station (manager) z.
2) The manager z checks whether other traps, concerning server *s'*, from others clients k '| k' ≠ k were received, which would be a confirmation of the server's crash:
   a) if not, it is assumed that a failure of connections only between s' and k is occurred,
   b) otherwise, the manager z modifies a set of servers S, by removing the server s' and adding to this set a new server k * (from the pool of other network nodes) and by promoting it to the role of the server *s'*.
3) The manager z instructs the servers of S to execute functional tasks. Functional tasks are:
   a) the server s' instructs servers S(s') to perform the task of checking the connection parameters S(s')(K), where (S(s')(K) denotes set of clients adjacent to servers of S(*s'*)),
   b) servers S(*s'*), after checking of connections parameters with clients S(*s'*)(K), send these parameters to s'; sending responses to s' confirms the absence of failure of servers S(s'),
   c) if there is such $s \in S(s')$, which did not respond, the server s' sends to the manager z a trap, which inform about the not-responding server; a trap service is performed like in step 2b),
4) The manager z instructs servers $s \in S$ to send back testing results, which are parameters values of connections between specific clients and servers.
5) After receipt of the test results from servers, the manager z allocate to clients k pairs of servers k(S). The method of allocation should result in an equal servers load and assuming that servers should support customers with the best connection parameter values.

In a DMVPN network, in relation to the general model of client-server, Hubs play a role of servers, Spokes play the role of clients and a manager manages a reconfiguration of Hubs and Spokes.

V. TEST ENVIRONMENT AND THE TECHNICAL FACILITIES USED IN THE COURSE OF EVALUATING THE FEASIBILITY OF THE ALGORITHM

Evaluating the feasibility of the algorithm was done in testing environment shown in Figure 8.

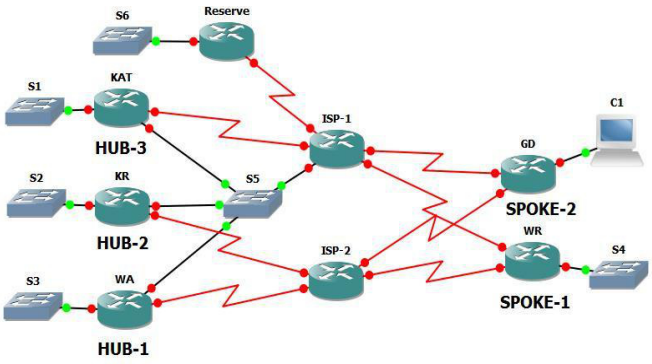


Fig. 8 Test environment of self-diagnosable DMVPN network

The set of potential Hub routers includes HUB-1, HUB-2, HUB-3 and Reserve. Routers called SPOKE-1 and SPOKE-2 register in the primary and backup server NHS (HUB-1 and HUB-2 routers). HUB-3 router is dynamically configured NHS server when the primary and/or backup server is unavailable.

Hub called Reserve was dynamically included to the set of hubs in case of failure of one of the hubs, which causes that testing in accordance with the PMC model assumptions was possible.

Simulated scenarios with symptoms of failure were as follows:

- loss of one NHS servers, caused by the failure of the primary link (leading to a primary NHS server) or simply by switching off NHS server,
- momentary loss of the primary server within IPSec connections.

Test environment has been built on the basis of Cisco equipment. Thus, offered by Cisco IOS system technical facilities for testing the quality of connections between nodes, with the possibility of sending a trap messages, generated by the device in the case of the unavailability of another tested device, were used. The logging from management station (C1 in Figure 8) on devices which require reconfiguration and the necessary changes in the configuration file was also used. At the moment, the procedure is not yet fully automated. Full automation of our system still requires writing shell scripts, woven into the configuration file of network devices (in the case of the Cisco, Tcl scripts).

The principal mechanisms of detecting the unavailability of network nodes and the state of the route's transmission parameters deterioration were IP SLA probes [12].

Sample configuration of two IP SLA probes is given below.

```
ip sla 1
 icmp-echo 10.0.0.1 source-interface Serial0/0
 timeout 1000
 threshold 1000
 frequency 30
ip sla schedule 1 life forever start-time now

ip sla 10
 udp-jitter 192.168.56.2 5000 codec g729a
 frequency 30
 ip sla schedule 10 life forever start-time now
```

The probe No. 1 was used to test reachability of a selected node (HUB router with the address 10.0.0.1), while probe No. 10 was used to verify the quality of the connection (transmission delay, packet loss, jitter).

In addition to the probes, the ability to track the status of the connection between nodes (up/down), with sending the syslog trap message to the management station, was used. A simple example of node's reachability monitoring, along with the illustration of sending messages to the management station when the Spoke loses connection with the Hub and the connection is restored, is given below.

```
track 1 rtr 1 reachability
 delay down 90 up 90

event manager applet track_SLA_1
 event track 1 state any
action 1.0 syslog msg "IPSLA collector 1 time out"


Dec 30 11:28:34.181: %TRACKING-5-STATE: 1 rtr 1 reachability Up->Down
Dec 30 11:28:34.217: %HA_EM-6-LOG: track_SLA_1: IPSLA collector 1 time
out

Dec 30 11:31:34.185: %TRACKING-5-STATE: 1 rtr 1 reachability Down->Up
Dec 30 11:31:34.245: %HA_EM-6-LOG: track_SLA_1: IPSLA collector 1 time
out
```

The result of the measurement of the quality of the connection between the network nodes is a series of statistics, as shown in the listing below. The most important are marked in bold.

```
WR#show ip sla statistics

Round Trip Time (RTT) for      Index 1
Latest RTT: 52 milliseconds
Latest operation start time: 14:09:33.018 MyZone Mon Dec 30 2013
Latest operation return code: OK
Number of successes: 23
Number of failures: 1
Operation time to live: Forever


Round Trip Time (RTT) for      Index 10
Latest RTT: 195 milliseconds
Latest operation start time: 14:09:03.222 MyZone Mon Dec 30 2013
Latest operation return code: OK
RTT Values:
Number Of RTT: 990    RTT Min/Avg/Max: 20/195/635 milliseconds
Latency one-way time:
  Number of Latency one-way Samples: 409
Source to Destination Latency  Min/Avg/Max: 2/72/240 milliseconds
Destination to Source Latency  Min/Avg/Max: 2/224/453 milliseconds
Jitter Time:
      Number of SD Jitter Samples: 987
      Number of DS Jitter Samples: 982
Source to Destination Jitter Min/Avg/Max: 0/17/211 milliseconds
Destination to Source Jitter Min/Avg/Max: 0/25/324 milliseconds
Packet Loss Values:
      Loss Source to Destination: 0          Loss Destination to Source: 0
      Out Of Sequence: 0     Tail Drop: 9
      Packet Late Arrival: 0  Packet Skipped: 1
Voice Score Values:
      Calculated Planning Impairment Factor (ICPIF): 13
MOS score: 4.00
Number of successes: 21
Number of failures: 2
Operation time to live: Forever
```

The availability of NHS servers was supervised (in accordance with the proposed algorithm) by Spoke nodes with reachability testing probes. In the case of the unavailability of the NHS servers, the management station, on the basis of received messages-traps, decides to make Spokes and Hubs reconfiguration (firstly, resets the

connection between the Hub and Spoke—*clear dmvpn session*, secondly, indicates a new set of primary and secondary NHS server) or/and to start the procedure of mutual testing Hub nodes to the designation of a new set of NHS servers and the new allocation of clients to servers.

In the case of removing one of the Hub from the set of Hubs, Hub named Reserve was included to the set of Hubs. Test procedure, consistent with the PMC model, was initiated by the manager and the Reserve's job was to initiate testing of all Hubs. It was assumed that Reserve is a reliable node (reachable by all the other nodes).

Functional test performed by a single Hub relied on the use of probe like No. 10, although only the reachability test was taken into account. Hub was treated as capable of realize its function if it was able to communicate with all of its Spokes (Spoke-1 and Spoke-2).

Specifying a new allocation can be carried out on the basis of analysis of the results of testing the quality of the connection between potential NHS servers and Spoke nodes (probe like No. 10). It should be noted that the test results can be used to determine the new allocation taking into account the response times of nodes, packet loss, jitter, but also current load related to the number of supported Spokes, CPU utilization, memory consumption, etc.

Currently, the new allocation was implemented on the basis of assigning Spokes to the Hub that hosts the least Hubs.

In the case of temporary unavailability of the NHS server, with tunnels protected by IPSec, the management station, on the basis of received message-trap from the Spoke, has decided to log on the Spoke and to break active IPSec session (clear crypto session).

VI. Conclusion

The experiment had the hallmarks of a “manually” controlled experiment (supervised). The development work on fully automate the procedure of DMVPN network management is underway. The authors have a preconception about the effectiveness of the proposed solution. The tests (manual inspection and reconfiguration of the system) did not allow to assess the impact of the procedure on convergence time of large DMVPN network. It seems that a good means of verification and comparison proposed algorithm with system without modification or with other solution are simulation studies, which the authors intend to accomplish in the near future. Also, an interesting issue seems to be develop effective procedure for load balancing of Hubs, which will be based on the results of testing the communication parameters in the DMVPN network.

References

[1] J. Maeng, , M. Malek,”A Comparison Connection Assignment for Self-Diagnosis of Multiprocessor Systems”. Digest Int′l Symp.FTC, 1981, pp. 173–175.

[2] M. Malek,”A Comparison Connection Assignment for Self-Diagnosis of Multiprocessors”, Systems, Proc. Seventh Int′l Symp. Computer Architecture, 1980, pp. 31–35, http://dx.doi.org/10.1145/800053.801906

[3] A., Sengupta, A. T. Dahbura,”On Self-Diagnosable Multiprocessors Systems: Diagnosis by the Comparison Approach”, IEEE Trans. Comput., 41, 11, 1992, pp. 1386–1396, http://dx.doi.org/10.1109/12.177309

[4] F. P. Preparata, G. Metze, R. T. Chien, R.T,”On the Connection Assignment Problem of Diagnosable Systems” IEEE Transactions on Computers Vol. EC-16 No. 6, 1967, pp. 848–854, http://dx.doi.org/10.1109/PGEC.1967.264748

[5] F. Barsi, F. Grandoni, P. Maestrini, ”A Theory of Diagnosability of Digital Systems”, IEEE Transactions on Computers, Vol. C-24, Np. 6, 1976, pp. 585–593, http://dx.doi.org/10.1109/TC.1976.1674658

[6] A. Arciuch, “Reliability state of connections in a microprocessor network with binary hypercube structure”, Electrical Review, R.86 No. 9/2010, pp. 154–156.

[7] S. L. Hakimi, A.T.Amin, “Characterization of Connection Assigmanet of Diagnosable Systems”, IEEE Transactions on Computers 1, 1974, pp.86-88, http://dx.doi.org/10.1109/T-C.1974.223782

[8] L. Zang, D. Ardagna, “Sla based profit optimization in autonomic computing systems”, Proceedings of ICSOC’ 04, 2004, http://dx.doi.org/10.1145/1035167.1035193

[9] Y. Hamadi, “Continuous resources allocation in Internet data centers”, CCGrid 2005. IEEE International Symposium on, 2005, pp. 566-573, http://dx.doi.org/10.1109/CCGRID.2005.1558604

[10] K. Lu, R. Yahyapour, P. Wieder, C. Kotsokalis, E. Yaqub, A. I Jehangiri, Y. Hamadi, “QoS – Based Resource Allocation Framework for Multi-Domain SLA Management in Clouds”, International Journal of Cloud Computing, Vol. 1, No. 1, 2013.

[11] „Dynamic Multipoint VPN (DMVPN) Design Guide”, Cisco Systems, Inc. 2006.

[12] „Cisco IOS IP SLAs Configuration Guide”, Cisco Systems, Inc. 2008.

[13] J. Chudzikiewicz, K. Murawski, ”Wyznaczanie bezkolizyjnych dróg przesyłania danych w sieci teleinformatycznej o strukturze typu hipersześcianu”, Diagnostyka 3(39), pp. 131-136 (in Polish).

[14] J. Chudzikiewicz, Z. Zieliński, “Resources placement in the 4-dimensional fault-tolerant hypercube processors network”, Studia Informatica Universalis, Volume 11 (2013), Number 1, pp.1-20