Defence Science Journal, Vol. 71, No. 4, July 2021, pp. 507-514, DOI : 10.14429/dsj.71.15623

Uninterrupted VPN Connection-Service with Mobility Management and Dead Peer Detection

Shreeram Hudda

Society for Electronic Transactions and Security, Chennai - 600 113, India
E-mail: hudda.shhudda@gmail.com

ABSTRACT

The VPN technology is used to connect client devices securely over unsecured public networks. The Connection handover and Dead Peer Detection (DPD) are the most challenging and important tasks in VPN. In this present article, a solution for connection handover is proposed which covers the pre-authentication with new credentials for a mobile device prior to connection handover. In VPN, a case where such connection handover is failed due to incomplete connection handover or several unusual events which make a VPN client to become offline or dead. To address DPD issues a solution is proposed which includes a Keep Alive Timer (KAT) server at the VPN server side. This proposed solution for DPD is far better than an already existing solution that uses a DNS server for DPD since the DNS server faces several attacks consistently. Thus, the proposed solution for DPD is more secure against such vulnerable attacks.

Keywords: Connection handover; DPD; Mobility management; VPN

1. INTRODUCTION

In the current scenario of this competitive world, an organisation has one central office and several branch offices. These branch offices should be connected securely, in some way, to the central office. This connection can happen in two ways. One way is by using the leased lines – since every branch office is geographically separated from its central office (i.e. the branch office can be in a different country or continent from its central office) so connecting such a branch office to the central office by using a leased line is inconvenient or even impossible. In another technique, the VPN service can be used to connect these geographically separated branch offices. Since VPN is a virtual link, and such link is dedicated to it's VPN client (i.e. branch office). Thus, Every VPN client is connected to its central office by using a separate and dedicated virtual link.

In the above mentioned case, the client is fixed or permanent to one physical location. In another case, an organisation has a moving employee; and that employee wants to connect securely with its central office or office. In such a case, the connection establishment by using a leased line is not convenient or even impossible since the employee is usually in a moving or volatile nature. He/she is moving from first physical location to second, and second to third, and so on. In such cases, the only VPN option remains. By using VPN, the moving employee gets connected with the central office or office.

Alshalan[1], *et al.*, in research study, presented several widely available issues in mobile VPN development. These issues are: a) an open application connection doesn't remain active while network connection gets interrupted or client device goes in sleep mode; b) resuming a TLS session by using lightweight handshake instead of full handshake upon network reconnection; c) handover of VPN session from one device, say tablet, to another device, say laptop; d) detect the unavailability of remote client; e) a lot of power consumption, even when the VPN tunnel is idle, requires to keep tunnel alive.

These commonly available issues are being investigated by many researchers in order to develop a mobile VPN. Some of them have already been solved but much research works still have to be done for this mobile-heavy industry. In this paper, the author is taking two difficulties of VPN technologies into consideration. The First difficulty/issue is about "Mobility management". When a mobile device is moving from one physical location or even one network to another physical location or another network then certain parameters (will be studied in the following sections) will get changed. Furthermore, these changed parameters will affect the already existing VPN connection. So there is also a need to resolve such an issue.

The second issue, that the author is discussing here, is "Dead Peer Detection" (DPD). When a mobile device gets connected with a VPN server, it becomes offline or unreachable after some time. The author also introduced a technique to identify or recognize such dead peers in VPN connection.

These both proposed solutions (for Mobility management, and for DPD) can be applied in part or in whole to existing network architectures (i.e. wireless network, GSM, CDMA, LTE, UMTS etc.), communication devices (any device which is capable for data communication such mobile phone, laptop, tablet, computer etc.), data storage centres (suchas a file storage

Received : 24 March 2020, Revised : 08 March 2021
Accepted : 22 March 2021, Online published : 01 July 2021

that enables multiple users to access data from a centralised system over the network, accessing files from HDDs or SSDs which are connected through Ethernet or Fiber channels, a pool of one and more storage sub-systems or systems which are connected through a network), and security systems (i.e. something is secured through a system of networking devices and components).

2. RELATED BACKGROUND WORK

The authors Alshalan[1], *et al.*, in research study, surveyed some Mobile VPN technologies, and the authors categorised them into three types: - (i) Network mobility-based VPN, (ii) Application mobility-based VPN, and (iii) HIP based VPN. In addition, they have presented some open issues after a study of some available mobile VPN products.

In work[2], the authors have presented a system and method for connection handover in VPN. The authors used SIM based pre-authentication prior to handing over the connection. In the work[3], the authors concluded that the mobile agent pre-authentication system provides better performance compared to the original mobile IP system. In study[4], the authors placed the DNS server behind the VPN server to check VPN tunnel status.

A method considered several choices; such as when a VPN client can establish one session at a time, multiple sessions are allowed for one peer; for establishing/re-establishing a session between the VPN client and the server[5]. The study[6] claimed that the existing methods for dead peer detection (DPD) can detect the liveliness of the tunnel only in the first Phase of IPsec (i.e. IKE SA). The patent[7] provided some techniques for uninterrupted VPN connection service with dynamic policy enforcement.

In patent[8], the author demonstrated several techniques for the accurate determination of network topology utilising multicast groups. In the patent[9], the authors mentioned several methods for implementing Prefix Discovery Server (PDS). These PDS servers assist VPN gateways to determine routes for remote secure private networks. In the patent[10], the authors made an arrangement in which a router, part of a local network, includes a security gateway module and a keep-alive module.

The authors analysed some well-known tunnelling protocols like GRE, IPsec, PPTP, and L2TP/IPsec based on several criteria like throughput, Jitter, Round Trip Time (RTT), and security; and they concluded the GRE is preferable for delay and bandwidth sensitive application in the context of site-to-site VPN whereas L2TP/IPsec more effective for remote access VPN[11]. Moreover, the multi-phase encryption algorithm may improve the security of the VPN network[12].

In the paper[13], the authors performed several passive and active measurements to investigate some security and privacy features on 283 android apps using VPN permission. They found some issues such as a) Third-party user tracking and access to sensitive android permissions, b) Malware presence, c) Traffic interception modes, d) Lack of encryption and traffic leaks, e) In-path proxies and traffic manipulation, and f) TLS interception. In study[14], the authors proposed a SIM-based authentication method, named as EAP-ESIM, to authenticate the mobile device with an access point.

From study[15], the author of this article can say that with the rapid growth in the number of mobile subscribers, and convergence of the Internet; the mobility management is one of the most challenging and important tasks. Therefore, MOBIKE[16], as described in RFC 4555, is best suited for situations where at least one peer (either VPN client or VPN server) does not move (i.e. one peer has fixed a IP address).

Mobile IP for IPv4 with IPsec suffers from Triangle routing problem[17]. The triangle problem occurs when a mobile device knows the fixed host address but the fixed host address does not know the mobile device's current address. When an open application connection remains active while the network connection gets interrupted or the client device goes in sleep mode then the application session remains persisted. The application session persistence is not assured in Mobile IP for IPv6 with IPsec[18].

The Host Identity protocol[19] (HIP), as mentioned in RFC 5201, needs all HIP enabled devices in the network. On the other hand, the framework of Media-Independent Pre-Authentication[20] (MAP), as described in RFC 6252, suffers from deployment issues such as ping-pong problem. The ping-pong problem arises when a mobile device is at the boundary of a network, and handover happens immediately.

After a detailed literature overview as mentioned above, the author of this present article can outline this work as: a) discussion about several issues that may be or may not be part of study1 in substantial limited or extended form, b) use of EAP-SIM based pre-authentication prior to connection handover[14], and c) use of Keep Alive Timer (KAT) server for DPD. The authors of study[4,8,10] used DNS servers, ping packets, and ping messages respectively for DPD. For every network certain varied policies, rules, and security parameters are defined. To apply such policies dynamically, new credentials are used to re-authenticate the mobile device[7]. A solution is also proposed to handle multiple session[5].

3. PROPOSED WORK

In this section, the author is proposing two separate solutions to resolve two different issues. These two issues are: i) Volatile or Vehicular nature of the mobile device or Mobility management, and ii) DPD. Firstly, the author wants to present a proposed solution for the volatile nature of a mobile device, and then one more solution for DPD.

3.1 Volatile or Vehicular Nature of Mobile Device

As stated earlier, it became more clear from study[15] that with the rapid growth in the number of mobile subscribers, and convergence of Internet and wireless mobile communications; volatile nature of mobile (i.e. mobility management) is one of the challenging and most important tasks. Mobility management is further divided into two sub-tasks a) Location management and b) Handover management. Location management enables the networks to track the locations of mobile devices throughout the process. Handover management enables the mobile device to keep its current connection alive when it moves from one network to another. With the increasing demands of real time services, the various wireless networks should satisfy different needs of mobile users with different characteristics and

quality of service. These networks or – even its better to say – heterogeneous networks are complementary to each other in terms of capability and suitability for different applications[15]. To seamlessly switch between different networks, an efficient mechanism is required for mobility management under such heterogeneous environments.

In study[2], the authors presented a method for VPN connection handover. They used SIM based pre-authentication prior to connection handover. Nowadays, it is a common thing for a mobile device to frequently switch between several different networks with different SSID like mobile data to Wi-Fi or vice versa and something likewise. So a mobile user may frequently move from one network connection to other network connections. For every network connection, the network administrator may want to install some additional policies or new rules[7]. To dynamically adapt these network policies, network resetting may be happening there. Hence, in such circumstances, the existing VPN session is got terminated[7] so it is necessary to re-authenticate the mobile device to a different network. Such network changes should be implemented during an existing VPN connection. The authors of study[2] did not consider pre-authentication with the dynamic installation of new rules prior to connection handover. They didn't consider session termination due to network resetting (i.e. new rules and policies). Thus, to resolve such an issue there is a need for a new mechanism.

While a mobile device is in vehicular nature then it will definitely enter into another network or wireless network. Then a new IP address is assigned to the mobile device since the mobile changed it's current network. To authenticate a mobile device with a new IP address, a handover process takes place. The delay in authenticating the device with a new IP address is critical. It may cause packet loss, and such delay is called handover delay.

In this present article, the author is assuming that a mobile device is already authenticated in a network and has a VPN connection. No handover delay occurs since EAP-SIM based pre-authentication[14] will take place prior to connection handover. So throughout the VPN connection, the IP address will remain the same whether the mobile device is changing network frequently. As stated above, when a mobile device is in volatile nature then new policies and rules are employed such as some user services are accessible in one network but not in another network, and some are accessible partly in one network whereas another network provides complete access to them. These new policies require dynamic, automatic, and real time re-authentication to the existing VPN session. To re-authenticate the mobile device, the new credentials are used. When re-authentication, by using new credentials, becomes successful then the VPN server will not allow the connection by old credentials[7]. Now the handover takes place.

In this paper, the author is categorising the complete handover process into four phases as follows:

(a) Phase 1 – Pre-Authentication Prior to Connection Handover
(b) Phase 2 – Connection Handover with New Credentials
(c) Phase 3 – Rejecting Connection with Old Credentials
(d) Phase 4 – End of Connection Handover

3.1.1 Phase 1

Pre-Authentication Prior to Connection Handover: - In this phase, a mobile device initiates SIM based pre-authentication (as shown in Figure 1), prior to the handover, to the authentication server via VPN server[14]. Since the mobile device is already authenticated in its current network (as per assumption – the mobile device is already authenticated and has a VPN connection) so re-authentication of the EAP-ESIM[14] algorithm is used for pre-authentication prior to connection handover in foreign/neighbour network.

When a mobile device is going out from coverage of one network and coming into coverage of another network then the radio signal strength of the first network goes down gradually. In its current network, the mobile device is already getting some radio signals from various networks. Then it tries to find out about neighbouring networks with different SSID which have radio signal strength higher than a certain predefined value. For radio signal strength, two different threshold values can be fixed. So that when a signal strength goes below than the first threshold value then the mobile device multicasts a control packet to every neighbouring network, which has radio signal strength higher than the second threshold value, and to VPN server. Each neighbouring network sends such a control packet to the VPN server only for pre-authentication purposes. The mobile device can recommend a network of high radio signal strength for pre-authentication. These control packets carry the subscriber's identity (i.e. a unique number that identifies every mobile device uniquely) since pre-authentication will take place based on subscriber's identity only[14]. International Mobile Subscriber Identity, and a randomly generated TMSI (Temporary Mobile Subscriber Identity) can be used for subscriber's identity. Then the VPN server forwards all these control packets to the authentication server. The VPN server makes authentication successfully with the help of an authentication server. The authentication server authenticates

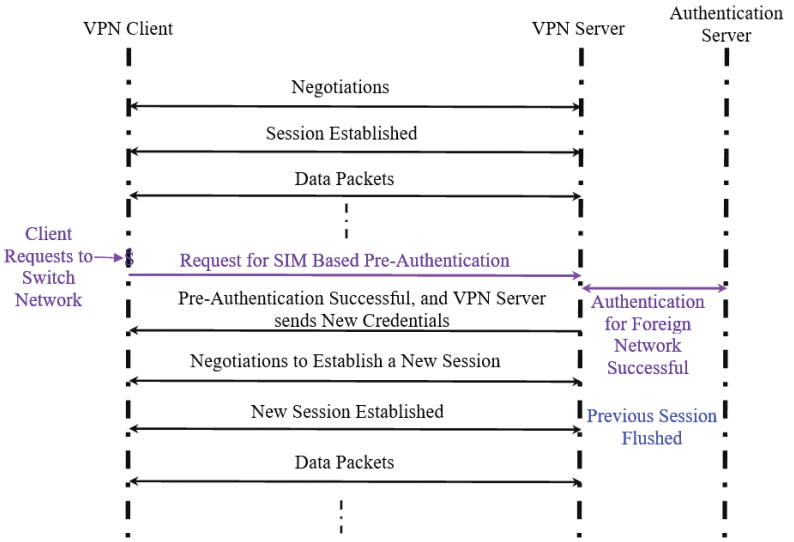

Figure 1. Flow Diagram for SIM-Based Pre-Authentication

the network based on the request, and sends a reply back to the VPN server. It's up to the authentication server that it can pre-authenticate the recommended network or any other network. When the first reply comes back to the VPN server then it immediately forwards the reply to the neighbouring network for which it got a reply first from the authentication server, and authenticates it. The VPN server will not wait for other replies from the authentication server. In such a manner, pre-authentication takes place automatically prior to connection handover.

As stated earlier, the dynamic network changes should be implemented during an existing uninterrupted VPN connection. To do so the VPN server also sends new credentials in the reply to the VPN client (i.e. mobile device). To verify the new credentials, the mobile device sends an acknowledgement to the VPN server. After that, some updates will take place in the lookup table for new credentials and the new foreign network's SSID. At the VPN server side, the lookup table contains the information about the mobile device which is connected to the VPN server, device's current network. It may also contain an entry to ensure that previously issued credentials for an existing VPN session cannot be used for re-authentication purposes. In the pre-authentication process, to improve the efficiency, multicasting is used to send or receive control packets to or from the VPN server and neighbouring network[2].

3.1.2 Phase 2

Connection Handover with new credentials: - Connection Handover with new credentials will happen only when a neighbouringnetwork gets a reply with new credentials, and those received credentials get verified. Thus, after establishing a successfully EAP-SIM based pre-authentication the handover takes place[2,14]. A neighbouring network (or foreign network) that which gets a reply first, will only allow the mobile device to send data packets with new credentials. Earlier, the mobile device was sending data packets with the previous network by using old credentials. In the new network, the device IP address remains the same since SIM based pre-authentication took place. So throughout the VPN tunnelling process the IP address of mobile remains the same even if the mobile device is changing it's network frequently. When a VPN connection with new credentials is established successfully then the VPN server replaces old credentials with new credentials in the lookup table. Thus, now the mobile device cannot establish the VPN connection with old credentials. When, somehow, an attacker is able to get old credentials, and tries to establish a VPN connection then the VPN server will not allow such connection since old credentials are not valid anymore. So an unauthorised connection is also not possible even with some valid form of credentials. After a long period, if a network will not get a reply for its control packet from the VPN server then such control packet will expire.

3.1.3 Phase 3

Rejecting connection with old credentials: - In the third phase, when a mobile device, with new credentials, established a VPN connection then the VPN connection for this mobile device with old credentials will not work anymore. The VPN server will reject such requests. Moreover, a mobile device will not be able to connect even with new credentials also, if another device is already connected with the same new credentials. It means that even with new credentials only one connection is allowed by the VPN server. So no duplicacy will take place. Thus, already authenticated devices with the latest credentials are allowed to establish a VPN connection.

3.1.4 Phase 4

End of Connection Handover: - In the last phase, after establishing a successful VPN connection the handover process will be ended. When the handover is ended then the respective control packet will be deleted, and after that an update will take place to update linking information in the lookup table at the authentication server.

The complete process, with all phases, is explained in Fig. 2. In this Figure, the author is representing each communication between any two ends (i.e. mobile and VPN server, Mobile and network, network and VPN server, VPN server and authentication server) by a numbered arrow-line. Every arrow-line is represented by a different number. A number is assigned according to phase such as for Phase 1 – it starts from 1000 and goes on in sequence till 1013 – completion of the phase, for Phase 2 – it starts from 2000 and continues in sequencing till 2004 – end of the phase, and similarly for remaining two phases as well. A set of numbers

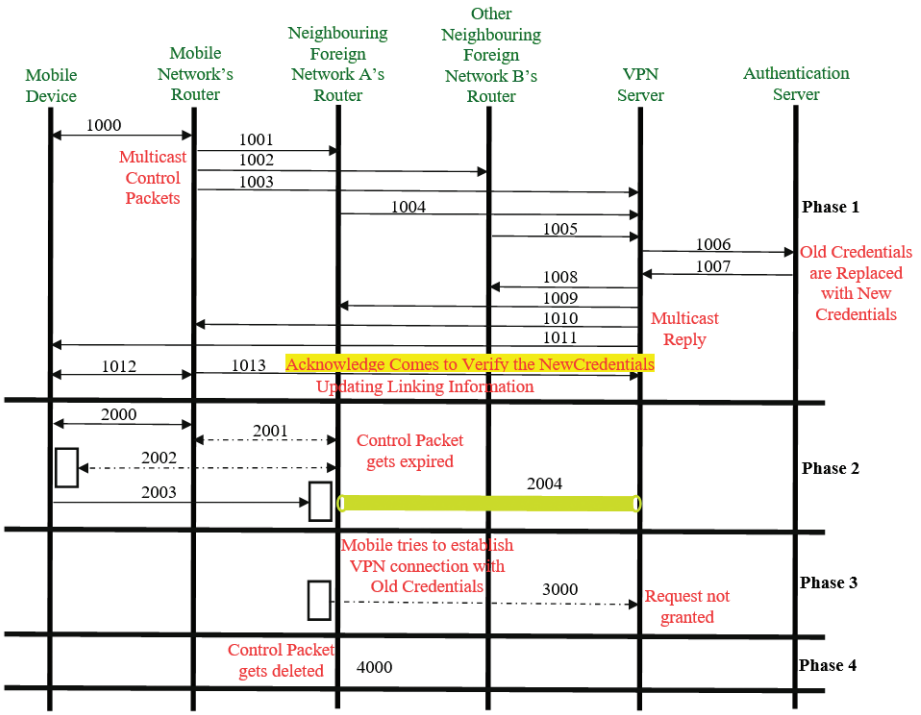

Figure 2. Flowchart for Connection Handover process.

in a phase also represents the sequence of communication such as the communication is represented by 1000 happens before communications represented by 1001 through 1013, and the communication is represented by 1001 happens after 1000 and before 1002 through 1013.

In the Figure 2, the markings of every sub-phase (i.e. 1000, 1001, etc.),in the text explaining each of 4 phases, has the following contexts -

Sub-phases of Phase 1 –

1000 – Mobile Device (MD) is associating with nearest Mobile Network's Router (MNR)

1001 – MNR is trying to make a connection with a Neighbouring Foreign Network (NFN) A's Router

1002 – MNR is trying to make a connection with an NFN B's Router

1003 – MNR forwards the MD information to VPN Server (VS)

1004 – NFN 'A' forwards the MD information to VS

1005 – NFN 'B' forwards the MD information to VS

1006 – VS forwards the MD information to the Authentication Server (AS)

1007 –AS sends reply to VS

1008 – VS forwards reply to NFN 'B'

1009 – VS forwards reply to NFN 'A'

1010 – VS forwards reply to MNR

1011 – VS forwards reply with new credentials to MD

1012 –MD communicates with MNR

1013 –MD acknowledges the reply with new credentials to VS and updates take place in the lookup table for these new credentials and the new foreign network's SSID

Sub-phases of Phase 2 –

2000 –Same as 1012 (it drawn here again to show the communication already happened between MD and MNR)

2001 – MNR handovers MD to NFN 'A'

2002 – MD is associating with NFN 'A'

2003 – MNR handovered MD to NFN 'A'

2004 – VPN Tunnel (VT) established between NFN 'A' and VS

Sub-phase of Phase 3 –

3000 – MD tries to establish a VT with old credentials. Hence, the VS rejects such a request

Sub-phase of Phase 4 –

4000–Connection handover ended. Hence, the control packet deleted

The first method is about the connection handover with pre-authentication. This applies only when a mobile device moves from one network to another network. However, due to some technical errors, routing problems, incomplete handover, incomplete roaming procedure[5], and some other network errors; it can be possible that such handover may not happen successfully. Thus, due to incomplete handover, a mobile device becomes dead (or offline or inactive). So in such a case, another mechanism, called as DPD, is needed to detect a dead mobile device.

3.2 Dead Peer Detection

Now, the author is proposing a method to detect dead peers. When a mobile device gets connected with a VPN server, after some time due to various unusual events (as mentioned above) the VPN tunnel or mobile may become inactive or offline. One or more such event(s) lead(s) the unavailability of connection between VPN client and server. In such cases, one or both peer(s) may not detect such unavailability, and keep believing that connection is still alive. So both peers are connected by a dead connection. Such a dead connection can cause more critical problems when there are a large number of VPN clients in the live environment. Moreover, the clients are connected with dead connections, and no one knows about it. So when a peer goes offline without properly informing another peer then initially another peer does not know about such a dead peer. For that a mechanism (as shown in Fig. 3) is used to detect such dead peers called dead peer detection (DPD). Thus, there should be a DPD to check the live status of VPN tunnels periodically.

As discussed in section 2, in the study[4], the DNS server is placed after and very close to the VPN server to check the liveness of the VPN tunnel. A DNS request is made to detect the tunnel's live status. If a reply comes before the expiration of the timer for such a DNS request then it indicates the tunnel is alive otherwise it is dead. Since DNS creation, it has become the most critical internet service. Most of the services are relying on DNS to work 24*7. The DNS is like an internet's phonebook, it translates user familiar names into corresponding specific numbers or vice-versa. As we know the DNS remains under continuous attacks since it is designed for usability not for security.

There are some DNS methods for DPD, they are

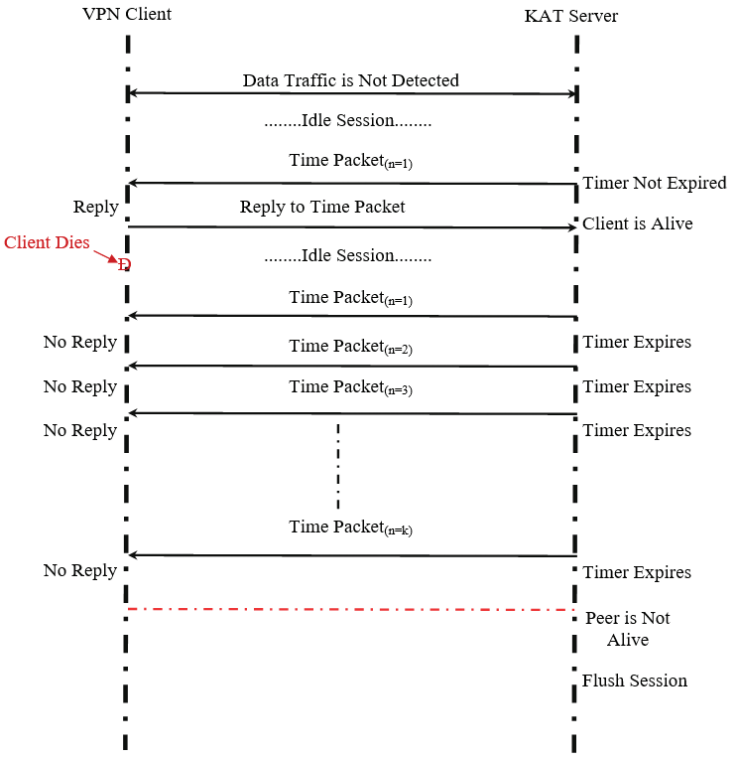

Figure 3. Flow Diagram for Dead Peer Detection.

implemented successfully in the VPN technology. However, they are attack prone because several Multinational Corporations (MNCs) are facing a lot of DNS attacks constantly. Some of these DNS attacks are DNS Hijacking, DNS Spoofing, DoS, DDoS, DNS Reflected, DNS amplification, and many more.

Since in study[4], the DNS server is used to detect the liveliness of a tunnel. So these attacks can harm such a DNS server as well. An attacker can trick a client and DNS server by believing bogus information. Many clients and servers do not check appropriately whether the reply comes from that DNS server to which they have made requests, and the answers are related to their query.

By using DNS spoofing, an attacker can change the IP address of an actual DNS server by his own malicious DNS server's IP address. Thus, all the legitimate DNS requests will come to this malicious server only. Moreover, it can be intentionally possible that the attacker's server does not want to respond to such DNS requests. Thus, after a certain number of iterations the tunnel becomes dead, but actually it is alive. The tunnel is dead since the client did not get a reply for the DNS request. So such issue can be faced if a DNS server is used to the detect live status of a tunnel.

The time interval of the attacker's malicious server is higher than the actual DNS server. In such a case, the DPD process will take more time than usual. Thus, a client gets a reply late or even very late for his DNS request.

An attacker can make a DNS server unavailable or unreachable to clients by using DoS, DDoS, DNS reflected, DNS amplification. Such attacks make the DNS server busy by replying to illegitimate DNS traffic. So when a client does not get a reply for a DNS request then it declares that the tunnel is not alive, but actually it is alive. Such things can be repeated by the attacker whenever a client sends a DNS request. So every time, the client gets dead live status of the VPN tunnel.

The time interval value for the attacker's malicious server may differ from the actual DNS server. Moreover, it may also be possible that an actual DNS server allows multiple sessions for a single device but a fake or malicious server does not. A dead peer can revive automatically as shown in Fig. 4. Thus, for such a peer the current DPD process is not completed but since it revives automatically so a new session needs to start. This new session may not be allowed by the malicious DNS server. Moreover, the DPD process is carried on the attacker's server so it will take more time. Thus, a client gets a reply late or even very late for a DNS request.

An attacker cannot reveal more confidential and personal information from a DNS request when traffic is passed via a VPN tunnel since it is in secured form. Although, the traffic is secured or encrypted from client to the VPN server, it is unencrypted from the VPN server to DNS server. Thus, from such an unencrypted part of the connection an attacker may get the client's information by using DNS attacks (as discussed previously). A malicious or fake DNS server can break the VPN connection or force re-establish the VPN connection between client and server every time after some duration.

The current paper presents a method for DPD. This proposed method is more secure against previously discussed

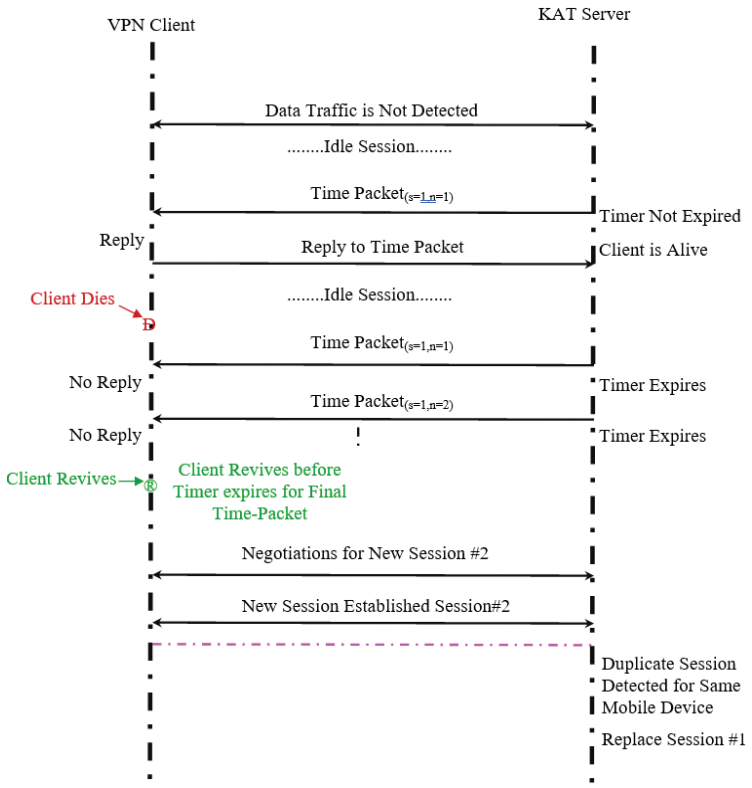

Figure 4. Flow Diagram for Session Replacement.

attacks. In this method, VPN tunnel status is checked using the "Keep Alive Timer" (KAT) server which is placed at the VPN server side. The KAT server works as: when a mobile device got connected with a VPN server, and it started to transmit data packets. However, after some time, it became offline i.e. the VPN tunnel is still there between client and server but the data packets are not coming/going from/to client anymore. It may be due to the unavailability of the client or several other reasons like the client is not interested, at least for now, to transmit the data packets. Then KAT server sends a time-packet that does not include any information about data rather it is an alert message for a client that if client will not respond to this time-packet then KAT server wishes to close the VPN connection forcefully. Such time-packets are sent by the KAT server only when a VPN tunnel is not used i.e. either side does not transmit the data. The VPN server counts the number of time-packets that have been sent for a respective tunnel. Thus, in addition to the timer facility, the VPN server should have a sequence of steps (will be discussed in the following paragraphs) to detect the data (i.e. traffic) in the tunnel.

Thus, to detect a dead peer, firstly, the system determines whether the tunnel has been established or not. If the tunnel is not established then there is a need to establish a tunnel between client and server rather than to send time-packets, and for such a tunnel the count value of time-packets is set to zero (0). If a tunnel is already established between client and server then the system tries to detect data traffic. If data traffic is detected then no need to send time-packets, and it is declared that the tunnel is alive. If a tunnel is already established but data traffic is not detected then time-packets are sent. The count value for time-packets is started by one (1), and a fixed timer is also started.

These time-packets have been sent till a certain predefined value (i.e. maximum threshold count value of time-packets). The maximum threshold value is the maximum number of time-packets. For such a time-packet, if a reply comes before the expiring fixed timer then the tunnel is reachable or alive. If a reply does not come then a fixed timer will expire, so once again the KAT server will resend a time-packet for the same fixed timer or for some additive increase of the previous timer. Once again, if a reply does not come before expiring such timer then the KAT server will resend a time packet again, and this sequence continues by the KAT server till either the reply will come or the threshold value will expire. If it is required to detect the live status of more than one tunnel then the sequence of these steps is repeated for every tunnel. The Fig. 5 explains the flowchart for sending a time packet.

Now, at another end of the tunnel the encrypted time-packet is received by the peer. Since the time-packet is received by a VPN client so it wants to respond. A client can only respond when the tunnel and itself are alive. Thus, the client records the time when the time-packet is received, and sets the count value to zero (0) since the peer is alive and going to respond. Now, the client sends the reply back to the KAT server. When the KAT server receives the reply successfully then it declares that the peer is reachable and alive. Now, for this peer, the KAT server resets the count value to zero (0). The Fig. 6 explains the flowchart for replying to a received time-packet.

The flow diagram for the complete process of DPD is shown in Fig. 7. The Fig. 7 is a combination of Fig. 5 and Fig. 6. In Fig. 5, the block named as “Reply comes” represents the complete Fig. 6, and it is shown in Fig. 7 by a special block.

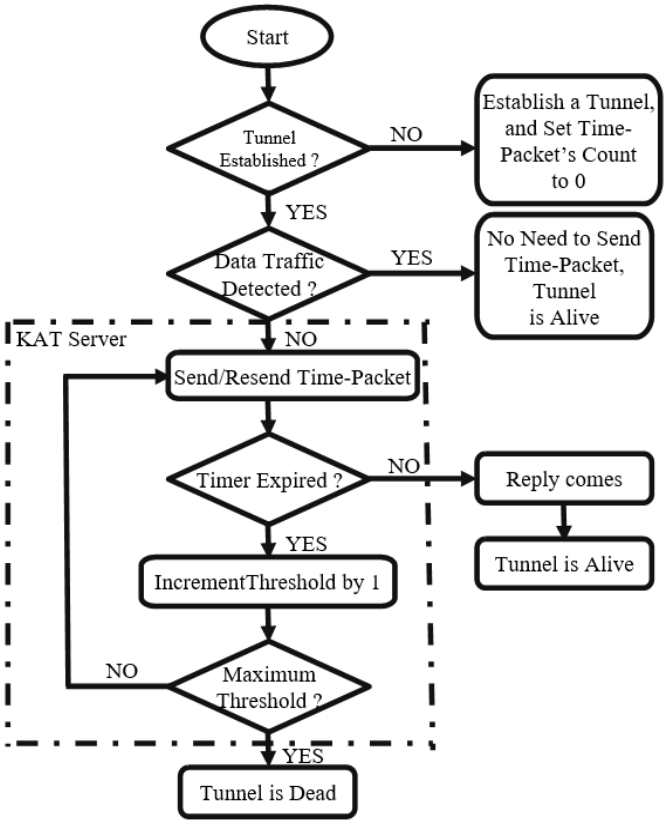

Figure 5. Flowchart for sending a Time-Packet.

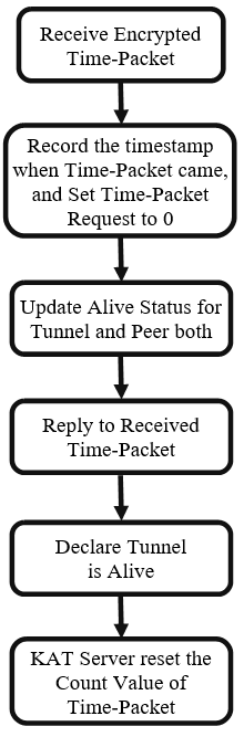

Figure 6. Flowchart for replying to a received Time-Packet.

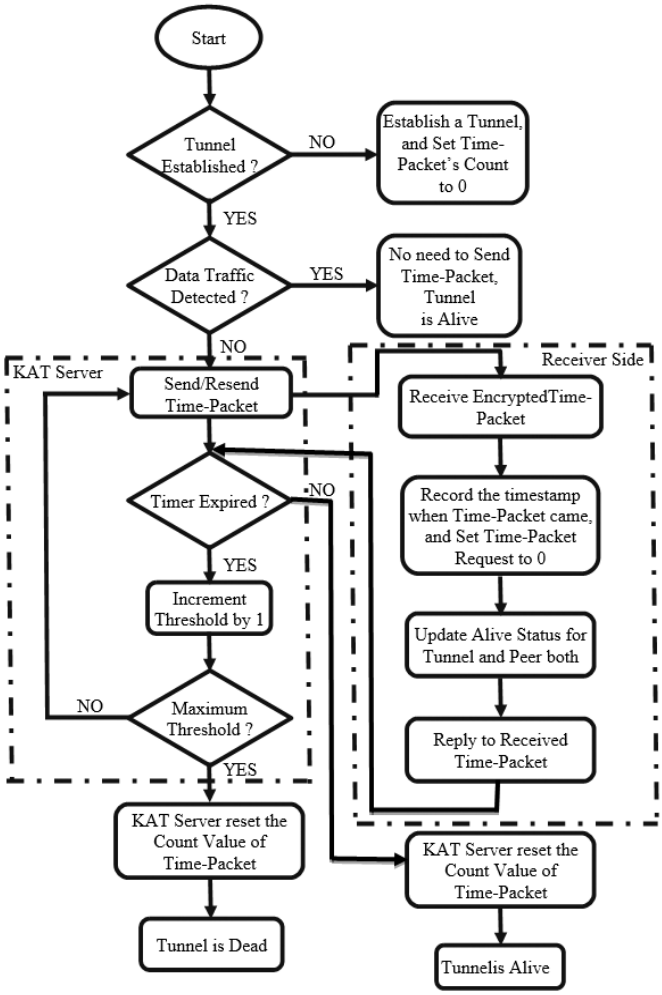

Figure 7. Flowchart for complete process of DPD.

4. CONCLUSION AND FUTURE SCOPE

In this paper, the author has presented two separate solutions for the two most difficult, challenging, and important tasks – connection handover and DPD – respectively. Several other research works also considered the same issues. However, in this paper, the connection handover method includes a single phase for pre-authentication with new credentials prior to connection handover. The mobile device switches or moves among several networks frequently. Some networks may support various new or additional policies and rules that others may not. Thus, a new session needs to be established under such a heterogeneous environment since the previous session is terminated due to network resetting. Hence, to establish a new session during the pre-authentication process every time new credentials are sent back to the client, and the client verifies the same. Due to incomplete connection handover or several other reasons, the client (i.e. mobile device) becomes unavailable or dead. So there is a need for another method/solution to detect such unavailable or dead peers. In the second method, a KAT server is placed at the VPN server side. Whenever the DPD request comes from the KAT server, the client sends a reply if it is alive otherwise it does not send a reply since it is dead. The author also compared the proposed DPD method with one already existing method (that uses a DNS server for DPD), and concluded that the proposed method is more secure than DNS based methods. However, these proposed methods still may face some issues such as speed of the mobile device, pre-authentication management – in case back-and-forth movement of a mobile device between current network and neighbour network, and between the candidate neighbour networks. Therefore, the author would work on these issues in the future study.

REFERENCES

1. Alshalan, A.; Pisharody, S. & Huang D. A Survey of Mobile VPN Technologies. *IEEE Communications Surveys and Tutorials*, 2015, 18(2), 1177-1196. doi:10.1109/COMST.2015.2496624
2. Wu, K.C.; Yang, J.S. & Lin, C.H. System and Method for Connection Handover in a Virtual Private Network. US patent 2006/0176852 A1, 2006.
3. Lin, C.H.; Yang, J.S. & Wu, K.C. Mobile Intelligent Agent Technologies to Support Intelligent Handover Strategy. *In Proceedings of the Workshop on Wireless, Ad-hoc, and Sensor Networks*, 2005, 521-528.
4. Son, G.; Tse, C. & Fedotenko. System and Method for Enabling VPN Tunnel Status Checking. US patent 8458248 B2, 2013.
5. Skraparlis, D. & Kondylis, K. Peer Revival Detection. US patent 9736244 B2, 2017.
6. Chen, Z.; Thangaveolu, A.; Xiang, D. & Yang, Y. Virtual Private Network Dead Peer Detection. US patent 9294461 B2, 2016.
7. Babula, A.; Attur, V.G. & Ananda, G.C. Uninterrupted Virtual Private Network (VPN) Connection Service with Dynamic Policy Enforcement. US patent 8209749 B2, 2012.
8. Ballanyne, A.J. Connectivity Outage Detection Based on a Multicast Management MPLS-VPN Group. US patent 7969908 B2, 2011.
9. Sax, W.C.; Wollman, W. & Jegers, E.H. VPN Discovery Server. US patent 8296839 B2, 2012.
10. Donzis, L.T.; Hughes, E.E.; Matelske, R.M. & Baron, P.W. Detecting if a Secure Link is Alive. US patent 6976071 B1, 2005.
11. Jahan, S.; Rahman, M.S. & Saha, S. Application Specific Tunneling Protocol Selection for Virtual Provate Networks. *IEEE International Conference on Networking Systems and Security (NsysS),* 2017. doi:10.1109/NSysS.2017.7885799
12. Singh, K.K.V.V. & Gupta, H. A new Approach for the Security VPN. *ACM Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies (ICTCS),* 2016. doi: 10.1145/2905055.2905219
13. Ikram, M.; Rodriguez, N.V.; Seneviratne, S.; Kaafar, M.A. & Paxson, V. An Analysis of the Privacy and Security Risks of Android VPN Permission-Enabled Apps. *ACM Proceedings of the 2016 Internet Measurement Conference (IMC),* 2016, 349-364. doi:10.1145/2987443.2987471
14. Liu, X. & Fapojuwo, A.O. An Efficient SIM-Based Authentication and Key Distribution Method for Wireless LANs. *Canadian Conference on Electrical and Computer Engineering,* 2005. doi:10.1109/CCECE.2005.1557185
15. Sen, J. Mobility and Handoff management in Wireless Networks: Trends in Telecommunications Technologies, Christos J. Bouras (Eds.), InTech, 2010, ISBN: 978-953-307-072-8. https://arxiv.org/abs/1011.1956 [Last Accessed on 19 January 2020].
16. https://www.ietf.org/rfc/rfc4555.txt [Accessed on 28 January 2021].
17. https://tools.ietf.org/rfc/rfc5944.txt [Accessed on 29 January 2021].
18. https://tools.ietf.org/rfc/rfc3775.txt [Accessed on 29 January 2021].
19. https://tools.ietf.org/rfc/rfc5201.txt [Accessed on 30 January 2021].
20. https://tools.ietf.org/rfc/rfc6252.txt [Accessed on 30 January 2021].

CONTRIBUTORS

Mr Shreeram Hudda, received MTech (Computer Science and Engineering) from Lovely Professional University, Jalandhar in 2016. Now, he is pursuing his PhD from Birla Institute of Technology and Science, Pilani, Pilani Campus. He has worked as researcher in Agriculture Knowledge Management Unit, Indian Agriculture Research Institute, New Delhi, and Society for Electronic Transactions and Security (Under Principal Scientific Advisor, Govt. of India), Chennai. His research areas are Cryptography, Network Security, Agile Development Methodologies, and Bioinformatics.