Journal of Computer Science 10 (9): 1712-1724, 2014
ISSN: 1549-3636

doi:10.3844/jcssp.2014.1712.1724 Published Online 10 (9) 2014 (http://www.thescipub.com/jcs.toc)

COGNITIVE AGENTS BASED SECURITY SCHEME TO HANDLE ROUTING LOOPS IN WIRELESS NETWORKS

[1]Kumar, R. and [2]G. Kousalya

[1]Department of Information Technology, Ramakrishna Institute of Technology, Coimbatore, Tamil Nadu, India
[2]Department of Computer Science and Engineering, Coimbatore Institute of Technology Coimbatore, Tamil Nadu, India

Received 2014-02-18; Revised 2014-03-18; Accepted 2014-04-19

ABSTRACT

Routers in wireless networks are often prone to variety of attacks like a man in the middle, distributed denial of service, smurf, ping of death, routing loops, counting to infinity, . Among all these attacks routing loop is the most common one and it have a harmful effect on network performance. In this study, we have proposed a novel cognitive agents based security scheme to handle routing loops in wireless networks. The proposed scheme uses Cognitive Agents (CAs) on every routers with Observation-Belief (O-B) model, which detect and handle routing loops efficiently. As a result, network performance improves with respect to various performance metrics like delay, packet loss ratio, bandwidth consumption, throughput, latency, queue length and so on.

Keywords: Wireless Networks, Cognitive Agents (CAs), Routing Loops, Security, Time To Leave (TTL), O-B Model

1. INTRODUCTION

1.1. Wireless Network Security

In recent years, wireless networks are gaining more popularity because they have become cheaper and more effective source of information (Kanawat and Parihar, 2011; Fan *et al*., 2005; Barman *et al*., 2007). However risks are inherent in wireless networks due to a variety of attacks, which have devastating impact on the network performance (Chou *et al*., 2009; Lashkari *et al*., 2009). Therefore security plays a vital role in wireless network.

1.2. Routing Loops

Routing is the process of forwarding the packets from source to destination through the shortest and secure possible path. Routing loops is a common problem in wireless networks. In a general wireless routing scenario, the source sends the packets containing destination address to the network of routers. It is the job of the routers to intercept the packet and forward it to the intended destination. If the router is witnessing routing loop attack then the packets continue to be in a loop forever (Chakrabarti and Manimaran, 2003; Yeung and Fung, 2004). Consider an example (**Fig. 1**), where there exists a source node 'S' and destination node 'D', along with a set of routers (R1,R2, R3 and R4). S first forwards the packets to R1, then to R2, which in turn forwards it to R3. In the usual case R3 was supposed to forward the packets to destination 'D'. Suppose R3 and R4 are under routing loop, then R3 forwards packets to R4 and R4 forwards it back to R1. This forces the packets to circulate in the loop until there Time To Live (TTL) value expires.

Routing loops are classified into two types i.e., transient routing loops and persistent routing loops. In transient routing loop, packets get trapped in a loop for a short period of time. Factors that cause transient routing loop are propagation delay, uneven routing table updation, changes in network topology and so on. In persistent routing loop, packets gets trapped in a loop for a prolonged period of time. Factors that causes persistent routing loop are routing table poisoning, miscon-figuration of router, duplication of router control information, overloading of shared links and so on. Compared to transient loop, persistent loop causes devastating impact on network performance (Xia *et al*., 2005; Saini and Khari, 2011).

Correspondent Author: Kumar, R., Department of Information Technology, Ramakrishna Institute of Technology, Coimbatore, Tamil Nadu, India

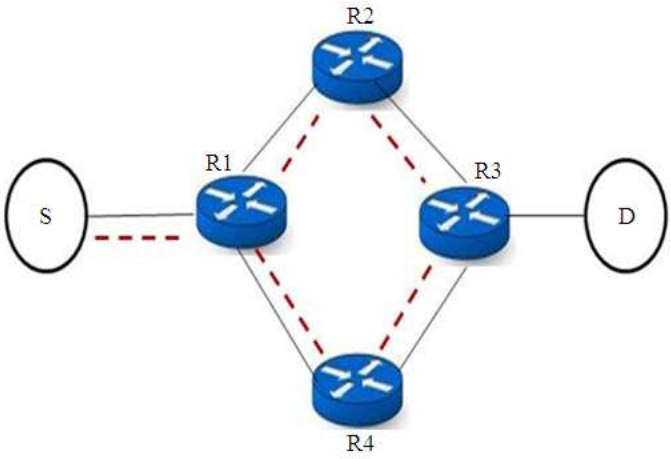


Fig. 1. Routing loop example

1.3. Impact of Routing Loops on Network Performance

Usually lack of consistency in routing table data causes routing loops. Among all routing attacks, routing loop attacks are the most common one and has a harmful effect on net-work performance, which includes unbounded delay, increased packet loss, out of order delivery of packets, excessive band-width consumption, degrading the quality of service, booming jitter, decreased throughput, packets get distracted and swing into wrong location, routing cache poisoning, count to infinity problem. Hence there is a need to detect routing loops in its early stages and develop counter measures to handle it properly (Waichal and Meshram, 2013).

1.4. Cognitive Agents

CA is a software entity which functions continuously and autonomously in a particular environment, able to carry out activities in a flexible and intelligent manner. CAs are good at handling routing problems in wireless network (Minar *et al*., 1999; Muraleedharan *et al*., 2007; Hengartner *et al*., 2002a). The following features of CAs helps in handling routing loops problems:

CAs observes the behavior of surrounding routers and makes decision accordingly. Suppose if the router is exhibiting suspicious behavior (i.e., frequently dropping packets, excessive link bandwidth consumption and so on) then CAs suspects that the packets routed over that path is in routing loop.

CAs learns the traffic pattern on the ongoing link, then compares the current traffic pattern with the earlier one. If the deviation among them is high then it senses that the link is getting over utilized and suspects the packets passing over that link to be in indeterminate loop.

The proactive nature of CAs helps in detecting the packets that are mistreated i.e., those packets will be prevented from further broadcasting and sending speed of the packets will also be reduced. Thereby the CA successfully determines the compromised router with malicious intent.

If the routing table is taking much longer time to con-center, then the CA with opportunistic nature helps in spotting the count to infinity problem.

CA diagnoses the network topology for artificial partitions. If it finds any artificial partition then the routing table is considered to be poisoned. Because artificial partitions are created by the wrong entries present in the routing table.

Since the wireless routers are restrictive in resources, having CAs based scheme will efficiently make use of resources to propose required security.

1.5. Proposed Cognitive Agents Based Adaptive Security Scheme

The proposed security scheme for routers in wireless networks, mainly consists of two functional components i.e., Action-Taker and Belief-Analyzer. The presence of routing loops are detected by incorporating O-B model in CA on every router. The Action-Taker with two sub components Observation-Identifier and Belief-Generator generate beliefs over the connections and then takes security actions based on the generated beliefs. If the generated belief is No-Routing-Loop, then the connection is considered as trustworthy and it will be continued as earlier; if the belief is Suspect-Routing-Loop, then the connection is considered for further analysis, suspected belief will be sent

to Belief-Analyzer component, which either confirms or ignores the suspected belief; if the generated belief is Confirm-Routing-Loop, then based on the deviation, the connection is declared as malicious or not.

1.6. Organization of Paper

The rest of the paper is organized as follows, section 2 gives some of the related works, section 3 provides some of the terminologies used in the study, section 4 explains CA with O-B model, section 5 discusses the proposed security scheme in the detail, section 6 gives a sample packet flow diagram for various formulated beliefs, section 7 discuss the results obtained, finally section 8 draws the conclusion.

2. RELATED WORKS

Routing loops are caused by inconsistencies in routing table. Hengartner *et al*. (2002b), routing loops are classified based on loop sizes and loop durations. Here, routing loops causes are identified and then an analysis is carried out to determine its impact on packet loss, delay incurred, link utilization and jitter. The results obtained shows that the routing loops have a profound impact on network performance.

Detection and analysis of routing loops in (Garcia-Lunes-Aceves, 1993), discusses routing loops manifestation in packet traces. Here an algorithm is presented that detects the presence of routing loops based on the packet replica streams. The algorithm first detects the packet replicas, validates the replicas and then merge the replica streams. The merged replicas are considered as routing loops and all the packets in the merged replicas are trapped in the routing loop. The algorithm is applied on packet traces of sprint IP backbone network and packet replicas is analyzed with respect to TTL value and several other factors. The paper does not consider per connection analysis for routing loops.

Loop free routing algorithms i.e., diffusing update algorithms are designed in (Francois and Bonaventure, 2005). These algorithms treat the distributed shortest path routing as diffusing computations problem and converges in finite time after any topological changes and link failures. It performs better than the existing loop free routing algorithms which involves message and storage ambiguities. But it lacks practical implementation and results for the newly proposed diffusing update algorithms.

In routing loops (Francois and Bonaventure, 2005) various topological changes that occur in large networks are discussed. Then it proves that by ordering the updates of the routing tables, transient loops can be avoided during interior gateway protocol convergence period. A protocol is also proposed for updating the routing table content, which in turn avoid the transient loop with less computation overhead. But the problem of updating consistent forwarding information base is not addressed.

3. DEFINITIONS

In this section, we provide definitions for some of the terminologies used in the study.

3.1. Looping Parameters

The networking parameters that causes a routing loop are referred as Looping Parameters. e.g.,: TTL value, Internet Protocol (IP) header checksum, link utilization rate, packet transmission rate.

3.2. Observation

Primarily, Observation means becoming aware of connections behavior based on their looping parameters value. Example: Conventional packet format, regular traffic, endangered integrity.

An observation is obtained from the collection of various looping parameter. E.g., an observation called Conventional packet format is obtained by a set of looping parameters like {TTL value of the packets passing through the router are unique, Packets are passing through a particular router only once, IP header checksum matches with the checksum of every hop along the path}.

3.3. Belief

A strongly held notion about routing loop existence or non-existence is known as belief. Example: No-routing loop, Suspect-routing-loop and Confirm-routing-loop.

A belief is deduced from various observations. E.g., a belief called Suspect-Routing-Loop is derived from a set of observations like {Unconventional packet format, Irregular traffic and Preserved integrity}. A detailed belief generation model is shown in **Fig. 2**.

3.4. Time Window

Time window is the measure of the number of packets that are transmitted in a specified period of time over a connection.

The Time Window Size (TWS) is determined as follows Equation 1:

$$TWS = CB * CRTT \quad (1)$$

where, CB is the Connection Bandwidth and CRTT is the Connection Round Trip Time.

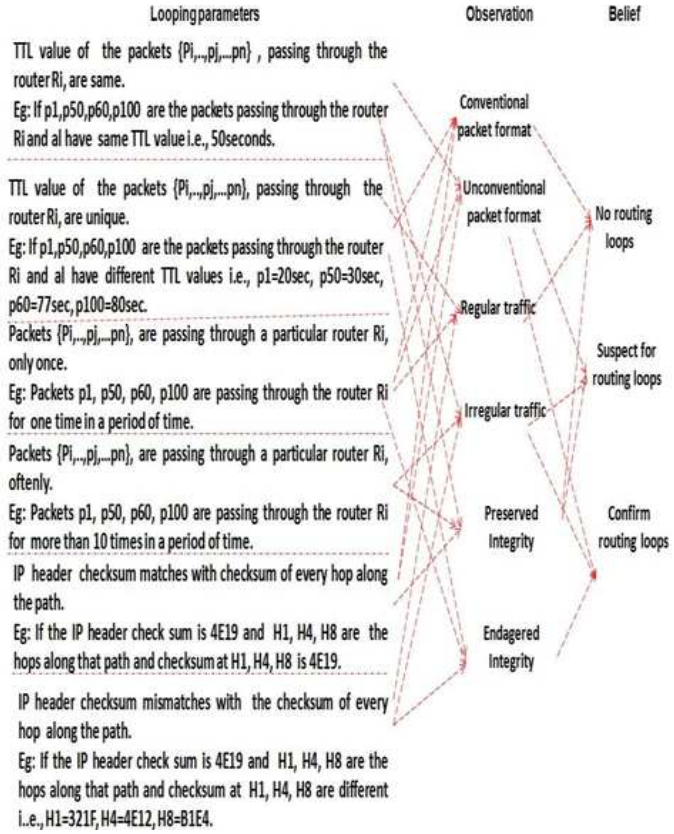


Fig. 2. A simple observation-belief model

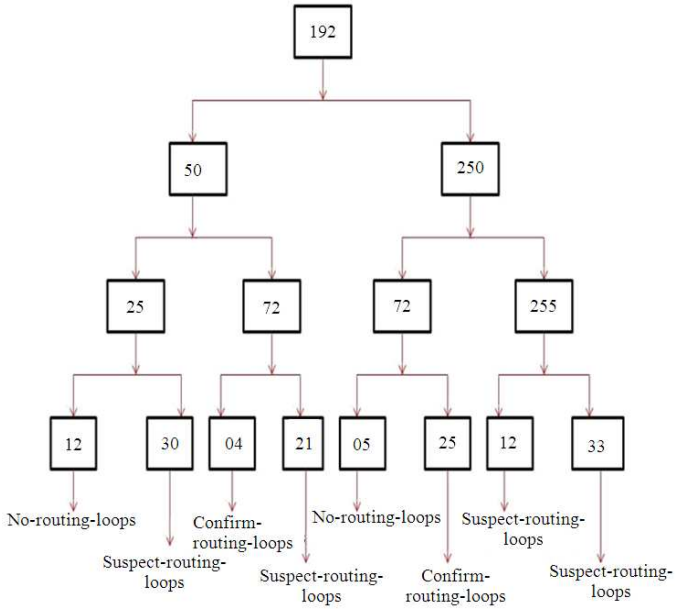


Fig. 3. Belief database tree structure

Table 1. Examples for class c IPV4 addresses

IP address	Network part	Host part
192.50.25.12	192.50.25	0.12
192.50.25.30	192.50.25	0.30
192.50.71.04	192.50.71	0.04
192.50.72.21	192.250.72	0.21
192.250.72.05	192.250.72	0.05
192.250.72.25	192.250.72	0.25
192.250.255.12	192.250.255	0.12
192.250.255.33	192.250.255	0.33

3.5. Belief Database

This database is available at Belief-Analyser for maintaining history of beliefs generated over the connections. The entries in Belief database are represented in tree form i.e., root node and intermediate node identify the network to which the connections belong to and leaf nodes stores the beliefs generated over the connections. A sample belief database tree structure is shown in **Fig. 3**, for some of the class C IPV4 addresses mentioned in **Table 1**.

4. COGNITIVE AGENTS BASED SECURITY SCHEME TO HANDLE ROUTING LOOPS

In this section, we first explain a wireless networking model considered for the proposed scheme and then discuss the functioning of CA along with its components and corresponding algorithms.

4.1. Network Model

Consider a wireless networking environment i.e., comprised of N nodes (connections) distributed over a wide geographical area (**Fig. 4**). Packet streams are forwarded from source nodes to destination nodes through several routers along the path. CA with O-B model is placed on every router, receives packet streams from various source nodes then intercepts looping parameters from each stream. After intercepting the looping parameter values, it generates a belief over that particular packet stream using O-B model. Then the generated belief will be analyzed further to determine the existence or non-existence of routing loops.

4.2. CA on Router

CA is placed on every router, mainly consists of two important components i.e., Action-Taker and Belief-Analyser. Action-Taker receives all the incoming connections and generates beliefs over the connection. Based on the generated belief, actions will be taken on the connections. Action-Taker will make use of Belief-Analyser while diagnosing any suspicious connections. CA along with its components is pictorially depicted in **Fig. 5**.

Action-Taker: Action-Taker with O-B model is one of the important functional components in the proposed architecture. The O-B model has two sub components i.e., Observation-Identifier (OI) and Belief-Generator (BG).

Three kinds of beliefs are generated over a connection i.e., *No-Routing-Loop, Suspect-routing-loop* and *Confirm-Routing-Loop*. In case of No-Routing-Loop, the connection is genuine without any malicious intent. In case of Suspect-Routing-Loop, connection is suspected to be malicious, Belief-Analyzer component is used for further analysis. In case of Confirm-Routing-Loop, connection immediately starts exhibiting steep looping parameters and decision cannot be taken based only on that. So if the number of times Confirm-Routing-Loop beliefs generated exceeds the Confirm-Routing-Loop threshold i.e., Th_{crl} then it is considered as malicious and will be terminated permanently else the connection is prone to be malicious, as a proactive measure its time window size will be shrinked for a CRTT period. The functioning of Action-Taker is given in algorithm 1.

The process of computing Th_{crl} is given as follows.

First, the weighted mean of Confirm-Routing-Loop belief i.e., WM_{crl} is computed Equation 2:

$$WM_{crl} = \left(Wn * N_{nrl} + Ws * N_{srl} + Wc * N_{crl}\right) / T_{rl} \quad (2)$$

Then the threshold for Confirm-Routing-Loop belief i.e., Th_{crl} is calculated based on the WM_{crl}:

$$Th_{crl} = WM_{crl}^{(1/Trl)} \quad (3)$$

Where:

Wn, Ws, Wc = The priority based weights assigned to *No-Routing-Loop, Suspect-Routing-Loop and Confirm-Routing-Loop* belief. Wc has higher priority, Ws has moderate priority and Wn have least priority

N_{nrl} = The number of times No-Routing-Loop belief is generated over the connection

N_{srl} = The number of times Suspect-Routing-Loop belief is generated over the connection

N_{crl} = The number of times Confirm-Routing-Loop belief is generated over the connection

T_{rl} = The total number of routing loops generated over the connection, in the given time window

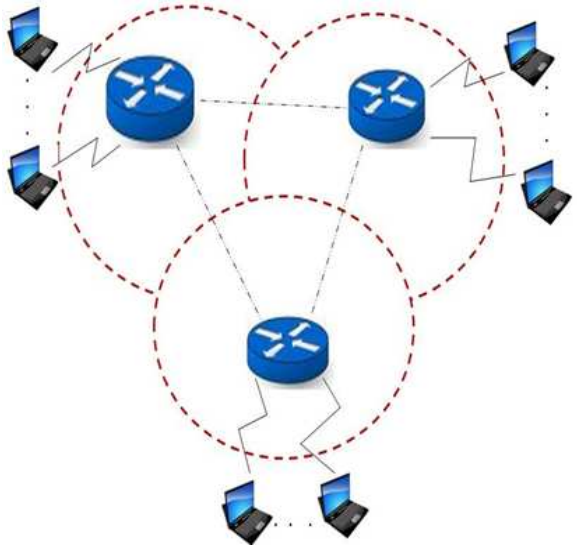


Fig. 4. Network model

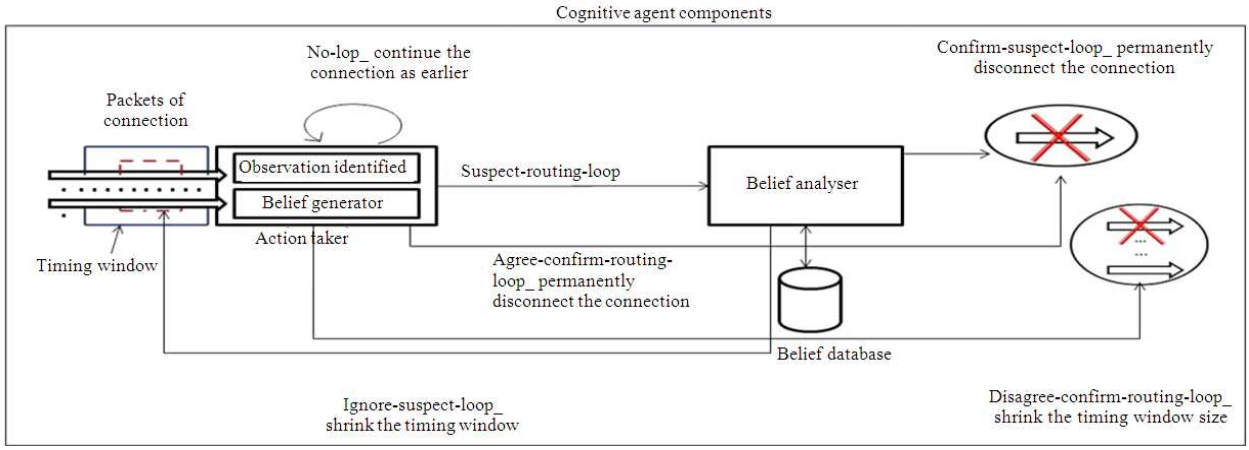


Fig. 5. CA on wireless router

4.3. Observation-Identifier

The OI helps in identifying observations for connection based on its current looping parameters value. The looping parameters are logically combined to form observations. A sample working of OI is given in algorithm 2:

Example1

TTL-Value^Packets-per-Router^ PChecksum = Conventional-packet-format.

Example2

TTL-Value^Packets-per-Router^IPChecksum = Regular-Traffic.

Example3

TTL-Value^Packets-per-Router^IPChecksum = Endangered-Integrity.

4.4. Belief-Generator

The BG generates beliefs based on the observations that are identified over each connection. Here, logical AND. operation is applied on the identified observations, based on the resultant value belief will be generated over the connection. A sample working of BG is given in algorithm 3.

Algorithm 1 Working of Action-Taker

1. Begin

2. Input: Packet streams from various connections.
3. Output: Beliefs i.e., *Bl* will be formulated for every input connections.
4. Initialize Number of confirm-routing-loop belief counter i.e., N_{crl} to zero.
5. Computer Th_{crl}.
6. for every connection do
7. Calculate the time window size for every new connection.
8. Accepts various looping parameters from connection C.
9. Observation set OB⇐Observation-identifier (looping parameter).
10. Belief i.e., Bl⇐ Belief-Generator (OB).
11. if Bl is no-routing-loop then
12. Continue the connection as earlier.
13. else if Bl is suspect-routing-loop then.
14. send Bl to Belief-Analyzer.
15. else if BI is confirm-routing-loop then.
16. Increment N_{crl}.
17. If $N_{crl} > Th_{crl}$ then
18. Agree confirm routing loop-Permanently disconnect the connection.
19. else
20. Disagree confirm routing loop-Shrink the time window size.
21. end if
22. end if
23. end for
24. End

Example 1

Conventional-packet-format^Regular-Traffic^Preserved-Integrity= No-Routing-Loop.

Example 2

Conventional-packet-format^Regular-Traffic^Preserved-Integrity = Suspect-Routing-Loop.

Example 3

Conventional-packet-format^Regular-Traffic^Preserved-Integrity = Confirm-Routing-Loop.

Algorithm 2 Logic of Observation-Identifier

1. Begin
2. Input: Looping parameters.
3. Output: Identified observations.
4. Accept various looping parameter values, say LVi, LVk and LVm.
5. LVi⇐TTL-Value.
6. LVk⇐Packets-per-Router value.
7. LVm⇐IPChecksum value.
8. if (LVi∧¬LVk∧¬LVm)
9. The conventional-packet-format observation generated.
10. else
11. The unconventional-packet-format observation generated.
12. end if
13. if (¬LVi∧¬ LVk∧LVm)
14. The Regular-Traffic observation generated.
15. else
16. The Irregular-Traffic observation generated.
17. end if
18. if (¬LVi∧LVk∧¬LVm)
19. The Preserved-Integrity observation generated.
20. else
21. The Endangered-Integrity observation generated.
22. end if
23. Return the identified observations set i.e., OB.
24. End

Algorithm 3 Logic of Belief-Generator

1. Begin
2. Input: Identified observations set.
3. Output: New belief generated over the connection.
4. Accept identified observations set.
5. Let OVi, OVk, OVm, be the values of identified observations in observations set.
6. OVi⇐Conventional-Packet-Format value.
7. OVk⇐Regular-Traffic value.
8. OVm⇐Preserved-Integrity value.
9. if (OVi∧OVk∧OVm) then
10. The No-Routing-Loop belief generated.
11. else if (¬OVi∧¬OVk∧OVm) then
12. The Suspect-Routing-Loop belief generated.
13. else if (¬OVi∧¬OVk∧¬OVm) then
14. The Confirm-Routing-Loop belief generated.
15. end if
16. Let Bl be the selected belief.
17. Return BI.
18. End

Example 1

Conventional-packet-format^Regular-Traffic^Preserved-Integrity = No-Routing-Loop.

Example 2

Conventional-packet-format^Regular-Traffic^Preserved-Integrity = Suspect-Routing-Loop.

Example 3

Conventional-packet-format^Regular-Traffic^Preserved-Integrity = Confirm-Routing-Loop.

4.5. Belief-Analyzer

Belief-Analyzer finds the Cumulative Deviation Factor (CDF) between the received suspected belief of a connection and that connections beliefs in belief database. It establishes a threshold for suspect-routing-loop beliefs i.e., Th_{srl} based on the history of beliefs in Beliefs database. If the CDF is within the Th_{srl} then the suspect-routing-loop belief will be ignored, but as looping parameters exhibited by the connection are little more than the normal range, as a proactive measure that connection will temporarily disconnected for a CRTT period else the connection is confirmed to be malicious and it will be terminated permanently. The logic of Belief Analyzer is shown in algorithm Equation 4 and 5:

$$CDF=\left(\left(Bl-Bl_{srl1}\right)+....+\left(Bl-Bl_{srln}\right)\right)/T_{srl} \quad (4)$$

Where:

Bl = The generated belief over the connection

$Bl_{srl1},....Bl_{srln}$ = The ignored and confirmed suspect-routing-loop beliefs in beliefs database

T_{srl} = The total number of suspect-routing-loop beliefs in beliefs database:

$$Th_{srl}=CDF/n \quad (5)$$

Where:

CDF = The Cumulative Deviation Factor of the suspect-routing-loop belief

n = The number of time the threshold is computed so far

Algorithm 4 Working of Action-Taker

1. Begin
2. Input: suspect routing loop belief.
3. Output: Ignore or confirm the suspect routing loop belief.
4. Initialize Belief database⇐NULL.
5. Initialize the threshold i.e., Th_{srl}.
6. Initialize CDF to zero
7. Initialize DF to zero
8. for every connection do
9. for every suspect routing loop belief do.
10. Compute DF between received suspected belief and the beliefs existing in the beliefs database.
11. CDF = CDF+DF
12. end for
13. if CDF>Th_{sri} then
14. Confirm the suspect belief: Terminate the connection permanently.
15. else
16. Ignore the suspect belief: Temporarily disconnect the connection.
17. end if
18. end for
19. Refresh the belief database periodically.
20. End

5. PACKET FLOW DIAGRAM FOR VARIOUS BELIEFS FORMULATED OVER THE CONNECTIONS

In this section, we discuss the general structure of the packet and routing table then sample packet flow diagram is drawn for every different kinds of beliefs (i.e., No-Routing-Loop, Suspect-Routing-Loop and Confirm-Routing-Loop) generated over the connections. **Figure 6** shows the general structure of the packet and routing table.

Figure 7 shows a sample packet flow diagram for No-Routing-Loop belief. Here, we can observe that TTL value of the packets are unique, header checksum of the packet and checksum at every hop are same and packet passes through the router exactly once. By seeing all these features CA on router formulates a belief called No-Routing-Loop over the connection.

Figure 8 shows a sample packet flow diagram for Suspect-Routing-Loop belief. Here, we can observe that TTL Value of the packets are unique, header checksum of the packet and checksum at every hop are not same and packet passes through the router twice. By seeing all these features CA on router formulates a belief called Suspect-Routing-Loop over the connection.

Figure 9 shows a sample packet flow diagram for Confirm-Routing-Loop belief. Here, we can observe that TTL value of the packets are same, header checksum of the packet and checksum at every hop are not same and packet passes through the same router very often. By seeing all these features CA on router formulates a belief called Confirm-Routing-Loop over the connection.

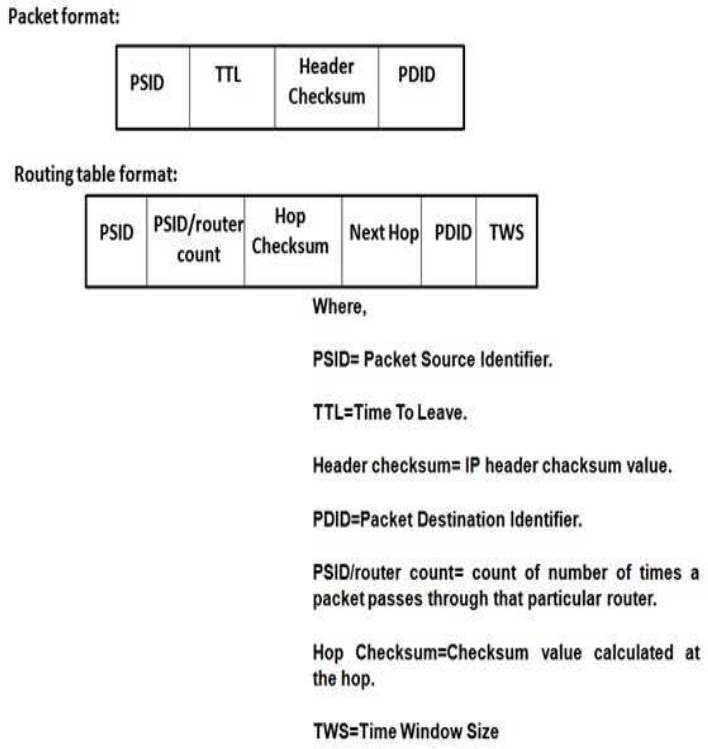
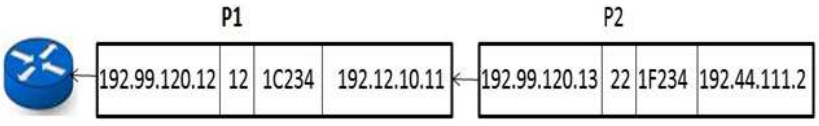


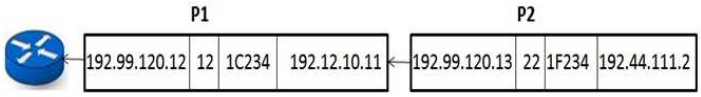
Fig. 6. General structure of packet and routing table



Routing Table

PSID	PSID/router count	Hop Checksum	Next Hop	PDID	TWS
192.99.120.12	1	1C234	192.99.121.22	192.12.10.11	122ms
192.99.120.13	1	1F234	192.99.12.2	192.44.111.2	90ms
.....	.....	.....	.....	.....	.....

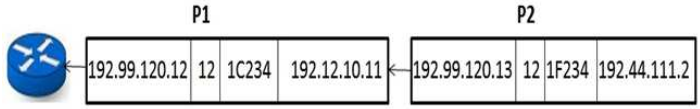
Fig. 7. Sample packet flow for No-Routing-Loop



Routing Table

PSID	PSID/router count	Hop Checksum	Next Hop	PDID	TWS
192.99.120.12	1	1C212	192.99.121.22	192.12.10.11	100ms
192.99.120.13	2	1EEE1	192.99.12.2	192.44.111.2	122ms
.....	.....	.....	.....	.....	.....

Fig. 8. Sample packet flow for Suspect-Routing-Loop



Routing Table

PSID	PSID/router count	Hop Checksum	Next Hop	PDID	TWS
192.99.120.12	5	1C212	192.99.121.22	192.12.10.11	99ms
192.99.120.13	6	1EEE1	192.99.12.2	192.44.111.2	110ms
.....	.....	.....	.....	.....	.....

Fig. 9. Sample packet flow for Confirm-Routing-Loop

6. RESULTS

In this section, we discuss the performance of the pro-posed cognitive agent based security with respect to various networking parameters like packet drops, latency, packet re-transmission rate.

Figure 10 shows a plot on beliefs over the connections Vs average delay incurred. Here, beliefs are classified into three types i.e., no routing loop, suspect routing loop and confirm the routing loop. In case of No-Routing-Loop (NRL), looping parameter values are within the normal range, so Action-Taker immediately formulates the belief and the amount of computation involved is less. As a result, the packet experiences zero incurred delay; In case of Suspect-Routing-Loop (SRL), the looping parameter values are slightly above the normal range so Action-Taker consults Belief-Analyser to ignore or confirm the belief. Here, the amount of computation involved may be slightly more so the packets may experience increased delay; In case of Confirm-Routing-Loop (CRL), exceeds the normal range at a sudden, but will be handled efficiently by the Action-Taker. As a result, the delay experienced by the packets will be less.

Figure 11 shows a plot on transmission time (sec) Vs throughput (bps). Here, CAs are built with O-B model so they are intelligent enough in tracking the routing loops. So the chances of packets getting trapped in a routing loop and wasting the bandwidth decreases. As a result, network throughput increases over time. Meanwhile, its history database will be updating so it can accurately detect the presence or absence of routing loop.

Figure 12 shows a plot on transmission time Vs packet loss ratio. Here as the packet transmission time proceeds the packet loss ratio experienced will be reduced because of the two factors i.e., belief generation and use of the time window. The CA on every router generates beliefs over the connections for every time window period. Time window helps in choosing an optimal rate for packet transmission over the connection and belief generation helps in selecting reliable path i.e., free from all kinds of routing loops for packet transmission. As a result the packets sent will be successfully delivered to the destination and packet loss experienced will be reduced over time.

Figure 13 shows a plot on the number of connections Vs efficiency in detecting the routing loops. As the number of connection increases the efficiency in detecting the routing loop also increases. CA diagnoses many connections, it gains more knowledge about the connections and its packet streams.

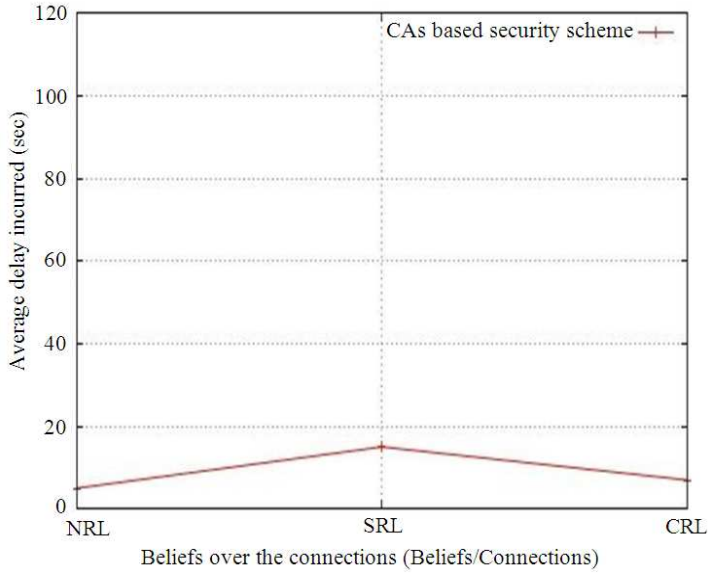


Fig. 10. Various beliefs over the connections Vs average delay incurred

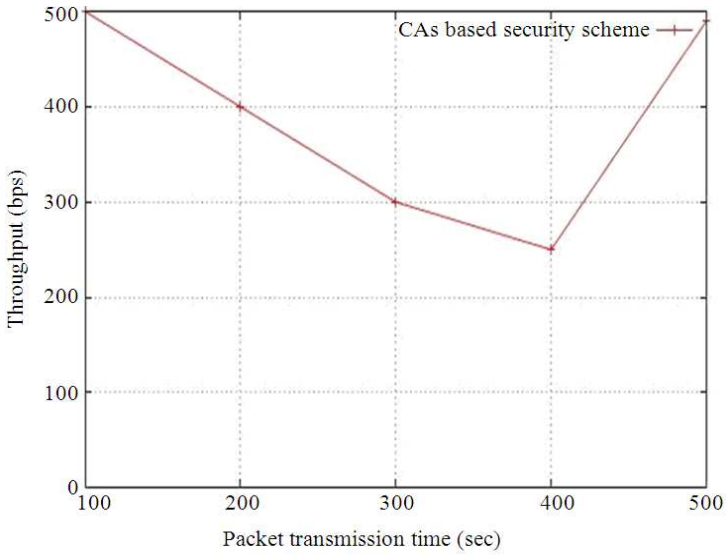


Fig. 11. Packet transmission time Vs throughput

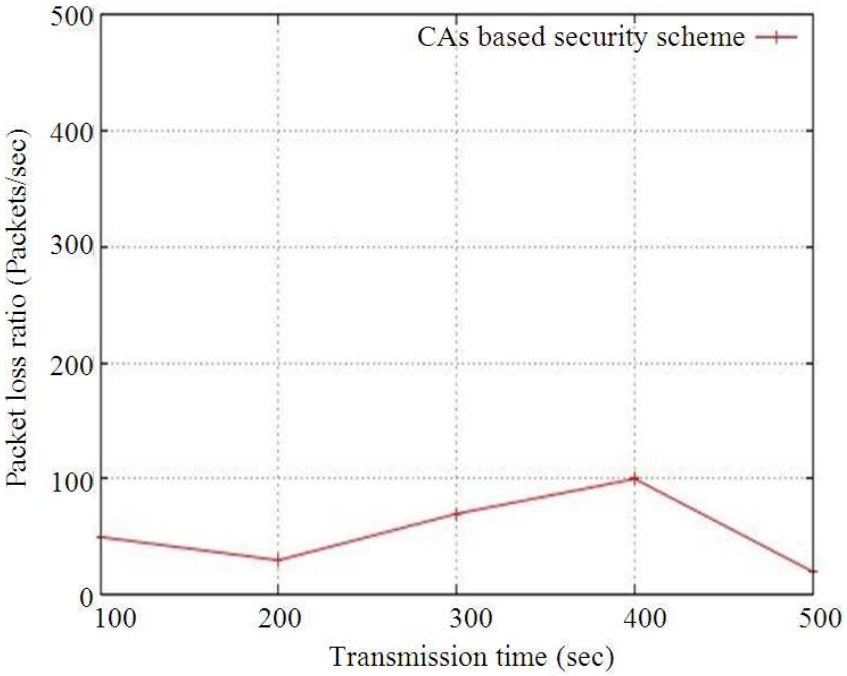


Fig. 12. Packet transmission time Vs Packet loss ratio

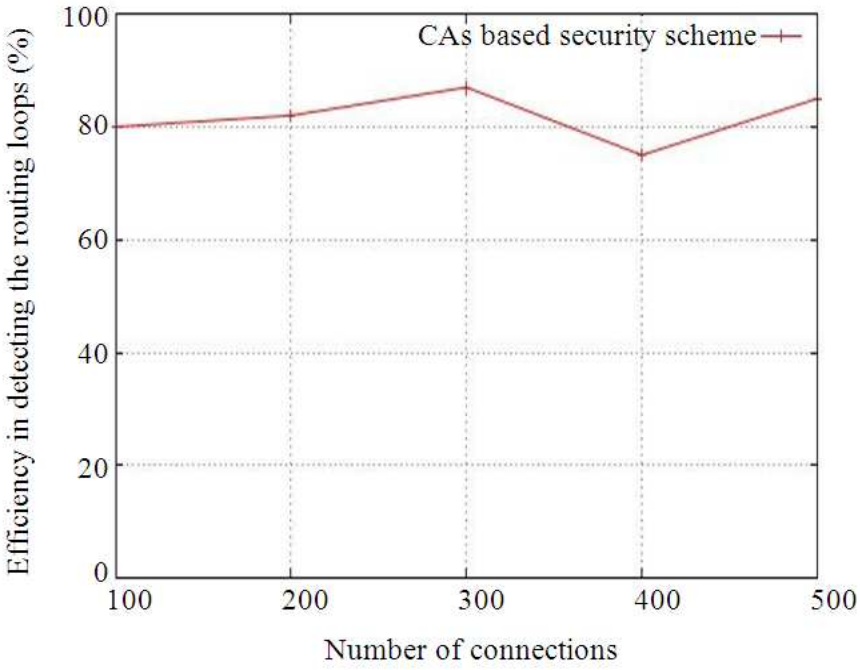


Fig. 13. Number of connections Vs efficiency in detecting the routing loops

7. CONCLUSION

In this study, we have presented a novel cognitive agents based adaptive security scheme for routers in wireless net-works. Cognitive thinking is employed in the router, which makes it more proactive and opportunistic in nature. Routers with cognitive agents can effectively detect the presence of routing loops and handle it efficiently. It also makes sure that the packets get transmitted only through the secure path that is free from all kinds of routing loops. The proposed security scheme enhances the network performance with respect to various performance metrics such as routing delay, latency, packet drop ratio, throughput, bandwidth consumption, queuing length.

8. REFERENCES

Barman, D., P. Satapathy and G. Ciardo, 2007. Detecting attacks in routers using sketches. Proceedings of the 7th Workshop on High Performance Switching and Routing, May 30-Jun. 1, IEEE Xplore Press, Brooklyn, NY, pp: 1-6. DOI: 10.1109/HPSR.2007.4281248

Chakrabarti, A. and G. Manimaran, 2003. A Scalable method for router attack detection and location in link state routing. Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks, Oct. 20-24, IEEE Xplore Press, pp: 293-294. DOI: 10.1109/LCN.2003.1243145

Chou, J.C.Y., B. Lin, S. Sen and O. Spatscheck, 2009. Proactive surge protection: A defense mechanism for bandwidth-based attacks. IEEE/ACM Trans. Netw., 17: 1711-1723. DOI: 10.1109/TNET.2009.2017199

Fan, Y., H. Hassanein and P. Martin, 2005. Proactive control of distributed denial of service attacks with source router preferential dropping. Proceedings of the 3rd ACS/IEEE International Conference on Computer Systems and Applications, (CSA' 05), IEEE Xplore Press, DOI: 10.1109/AICCSA.2005.1387064

Francois, P. and O. Bonaventure, 2005. Avoiding transient loops during IGP convergence in IP networks. Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies, Mar. 13-17, IEEE Xplore Press, pp: 237-247. DOI: 10.1109/INFCOM.2005.1497895

Garcia-Lunes-Aceves, J.J., 1993. Loop-free routing using diffusing computations. IEEE/ACM Trans. Netw., 1: 130-141. DOI: 10.1109/90.222913

Hengartner, U., S. Moon, R. Mortier and C. Diot, 2002a. Routing loops: Detection and analysis of their impact on loss and delay. Proceedings of the Proceedings of ACM/USENIX Internet Measurement Workshop, (IMS' 02).

Hengartner, U., S. Moon, R. Mortier and C. Diot, 2002b. Detection and analysis of routing loops in packet traces. Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurment-ACM, Nov. 06-08, ACM Press, New York, pp: 107-112. DOI: 10.1145/637201.637217

Kanawat, S.D. and P.S. Parihar, 2011. Attacks in wireless networks. Int. J. Smart Sensor Ad Hoc Netw.

Lashkari, A.H., O.B. Zakaria, S. Farmand and R. Saleh, 2009. Shoulder surfing attack in graphical password authentication. Int. J. Comput. Sci. Inform. Security, 6: 145-154.

Minar, N., K.H. Kramer and P. Maes, 1999. Cooperating mobile agents for dynamic network routing. Software Agents Future Commun. Syst. DOI: 10.1007/978-3-642-58418-3_12

Muraleedharan, R., Y. Yan and L.A. Osadciw, 2007. Detecting sybil attacks in image sensor network using cognitive intelligence. Proceedings of the 1st ACM Workshop on SENSOR and Actor Networks, Sep. 09-14, Montreal, Canada, pp: 59-60. DOI: 10.1145/1287731.1287746

Saini, R. and M. Khari, 2011. Defining malicious behavior of a node and its defensive methods in Ad Hoc network. Int. J. Comput. Applic., 20:18-21. DOI: 10.5120/2422-3251

Waichal, S. and B.B. Meshram, 2013. Router attacks-detection and defense mechanisms. Int. J. Sci. Technol. Res., 2: 145-149.

Xia, J., L. Gao and T. Fei, 2005. Flooding attacks by exploiting persistent forwarding loops. Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement, (CIM' 05), Association Berkeley, pp: 36-36.

Yeung, K.H. and W.K. Fung, 2004. Attacking routers by packet misrouting. WSEAS Trans. Commun.