Synchronization and Time Slot-Based Method for Vampire Attacks Detection in Wireless Sensor Networks

[I] N. Keerthikaa MCA., [II] K. Devika M.Sc., MCA., M.Phil.,

[I] Research Scholar, Bharathiar University, Coimbatore, [II] Assistant Professor, CS,
[I, II] Dept. of Computer Science, Maharaja Co-Education Arts and Science College,
Perundurai, Erode – 638052.
[I] Email id: keerthinagarajan@gmail.com
[II] Email id: devika_tarun@yahoo.co.in

ABSTRACT: Ad hoc Wireless Sensor Networks (WSNs) promise exciting new applications in the near future, such as ubiquitous on-demand computing power, continuous connectivity, and instantly deployable communication for military and first responders. Such networks already monitor environmental conditions, factory performance, and troop deployment, to name a few applications. As WSNs become more and more crucial to the everyday functioning of people and organizations, availability faults become less tolerable—lack of availability can make the difference between businesses as usual and lost productivity. Ad hoc low-power wireless networks are an exciting research direction in sensing computing. Prior security work in this area has focused primarily on denial of communication at the routing or medium access control levels. This paper **explores resource depletion attacks** at the routing protocol layer, **which permanently disable networks by quickly draining nodes' battery power**. These "Vampire" attacks are not specific to any specific protocol, but rather rely on the properties of many popular classes of routing protocols. It is found that all examined protocols are susceptible to Vampire attacks, which are devastating, difficult to detect, and are **easy to carry out using as few as one malicious insider sending only protocol-compliant messages**. **Carousel attack** and **Stretch attack** are the possible scenarios occurred. To avoid such attacks **attestation based forwarding scheme** and **loose source routing** is proposed. The thesis proposes the concepts using Content and Presence Multicast Protocol (CPMP) using the FPD (Future Peak Detection) and RFPD (Randomized Future Peak Detection) algorithms which nodes use to send updates to their neighbors. The updates contain the relative time of their sender's next transmission. To address the energy efficiency problem the algorithms propose by synchronizing the transmission times of all the nodes in the system. Transmission synchronization presents energy saving opportunities through dynamic power management of the network interface. That is, nodes can switch off their wireless interfaces between transmissions. However, in uncontrolled ad hoc environments, a single malicious user can easily disrupt network stability and synchronization, affecting either the nodes' power savings or their ability to receive updates from their neighbors.

Keywords: Carousel attack, Stretch attack, Future Peak Detection Content, Presence Multicast Protocol, Randomized Future Peak Detection, Sensor.

1. INTRODUCTION

1.1. Ad Hoc Wireless Sensor Network

Ad-hoc wireless sensor networks (WSNs) promise exciting new applications in the near future, such as ubiquitous on-demand computing power, continuous connectivity, and instantly deploy able communication for military and first responders. Such networks already monitor environmental conditions, factory performance, and troop deployment, to name a few applications. As WSNs become more and more crucial to the everyday functioning of people and organizations, availability faults become less tolerable—lack of availability can make the difference between business as usual and lost productivity, power outages, environmental disasters, and even lost lives; thus high availability of these networks is a critical property, and should hold even under malicious conditions. Due to their ad hoc organization, wireless ad hoc networks are particularly vulnerable to denial of service (DoS) attacks, and a great deal of research has been done to enhance survivability While these schemes can prevent attacks on the short-term availability of a network, they do not address attacks that affect long – term availability— the most permanent denial of service attack is to entirely deplete nodes' batteries.

These attacks are distinct from previously studied DOS, reduction of quality (ROQ), and routing infrastructure attacks as they do not disrupt immediate availability, but rather work over time to entirely disable a network. While some of the individual attacks are simple, and power draining and resource exhaustion attacks have been discussed before prior work has been mostly confined to other levels of the protocol stack, e.g., medium access control (MAC) or application layers, and to our knowledge there is little discussion, and no thorough analysis or mitigation, of routing-layer resource exhaustion attacks .

Vampire attacks are not protocol-specific, in that they do not rely on design properties or implementation faults of particular routing protocols, but rather exploit general properties of protocol classes such as link-state, distance-vector, source routing and geographic and beacon routing. Neither do these attacks rely on flooding the network with large amounts of data, but rather try to transmit as little data as possible to achieve the largest energy drain, preventing a rate limiting solution. Since Vampires use protocol-compliant messages, these attacks are very difficult to detect and prevent.

1.2. Denial of Service Attack

Adversary injecting malicious information or altering legitimate routing setup messages, or can prevent the routing protocol from functioning correctly. For example, an attacker can forge messages to convince legitimate nodes to route packets in a way from the correct destination. Vampire attack is one of the resource depletion attacks. The resource depletion attack focuses the node's batteries life. Vampire attacks affect any protocol and utilize the properties of routing protocols classes such as source routing, distance vector and link state and geographic and beacon routing.

Dynamic Source Routing (DSR) Protocol is a stateless protocol do not store or maintain any routing information at the nodes. The source node specifies the entire route to a destination within the packet header, so intermediaries do not make independent forwarding decisions, relying rather on a route specified by the source. An adversary arranges packets with knowingly establish routing loops sends packets in circles targets source routing protocols by take advantage of the limited verification of message headers at forwarding nodes, allowing a single packet to repetitively traverse the same set of nodes that is called Carousel attack.

1.3. Contributions

In this thesis makes three primary contributions. First, one is thoroughly evaluate the vulnerabilities of existing protocols to routing layer battery depletion attacks. They are observe that security measures to prevent vampire attacks are orthogonal to those used to protect routing infrastructure, and so existing secure routing protocols and do not protect against Vampire attacks. Existing work on secure routing attempts to ensure that adversaries cannot cause path discovery to return an invalid network path, but vampires do not disrupt or alter discovered paths, instead using existing valid network path s and protocol- compliant messages.

Protocols that maximize power efficiency are also inappropriate, since they rely on cooperative node behavior and cannot optimize out malicious action. Second, one is show the simulation results quantifying the performance of several representative protocols in the presence of a single Vampire (insider adversary). Third, one is modify an existing sensor network routing protocol to provably bind the damage from Vampire attacks during packet forwarding.

Vampire attacks, a new class of resource consumption attacks that use routing protocols to permanently disable ad hoc wireless sensor networks by depleting no des' battery power. These attacks do not depend on particular protocols or implementations, but rather expose vulnerabilities in a number of popular protocol classes. They are showed a number of proofs of concept attacks against representative examples of existing routing protocols using a small number of weak adversaries, and measured their attack success on a randomly generated topology of 30 nodes.

The proposed technique routing protocol are provably bounds damage from Vampire attacks by verifying that packets consistently make progress toward their destinations and reduce the reimbursement. Derivation is damage bounds and defenses for topology discovery, as well as handling mobile networks

2. PROBLEM FORMULATION

2.1. Problem Definition

The first challenge in addressing Vampire attacks is defining DOS attacks in wired networks are frequently characterized by amplification of adversary can amplify the resources it spends on the attack, e.g., use 1 minute of its own CPU time to cause the victim to use 10 minutes. However, consider the process of routing a packet in any multi-hop network: a source composes and transmits it to the next hop toward the destination, which transmits it further, until the destination is reached, consuming re-sources not only at the source node but also at every node the message moves through.

If they are consider the cumulative energy of an entire network, amplification attacks are always possible, given that an adversary can compose and send messages which are processed by each node along the message path. So, the act of sending a message is in itself an act of amplification, leading to resource exhaustion, as long as the aggregate cost of routing a message (at the intermediate nodes) is lower than the cost to the source to compose and transmit it. So, we must drop amplification as our definition of maliciousness and instead focus on the cumulative energy consumption increase that a malicious node can cause while sending the same number of messages as an honest node.

They are define a Vampire attack as the composition and transmission of a message that causes more energy to be consumed by the network than if an honest node transmitted a message of identical size to the same destination, although using different packet headers. They are measure the strength of the attack by the ratio of network energy used in the benign case to the energy used in the malicious case, i.e., the ratio of network-wide power utilization with malicious nodes present to energy usage with only honest nodes when the number and size of packets sent remains constant. Safety from Vampire attacks implies that this ratio is 1. Energy use by malicious nodes is not considered, since they can always unilaterally drain their own batteries.

The stretch attack is more challenging to prevent, its success rests on the forwarding node not checking for optimality of the route. If we call the no-optimization case "strict" source routing, since the route is followed exactly as specified in the header, loose source routing is defined, where intermediate nodes may replace part or all of the route in the packet header if they know of a better route to the destination. This makes it necessary for nodes to discover and cache optimal routes to at least some fraction of other nodes, partially defeating the as-needed discovery advantage.

2.2. Main Objective

1. To extend the generic algorithm and implement the Weight Based Synchronization algorithm to find the winner slot to store the packet data.

2. To extend the Weight Based Synchronization algorithm and implement the Future Peak Detection algorithm to avoid the inflation attack which is made by sending false maximum weight among the nodes.
3. To extend the Future Peak Detection algorithm and implement the Randomized Future Peak Detection algorithm to synchronize all the neighbor nodes by using all the slots.

2.3. System Methodology

In this thesis, they existing present a series of increasingly damaging Vampire attacks, evaluate the vulnerability of several example protocols, and suggest how to improve resilience. In source routing protocols, we show how a malicious packet source can specify paths through the network which are far longer than optimal, wasting energy at intermediate nodes that forward the packet based on the included source route.

2.3.1 Carousel Attack and Stretch Attack

In routing schemes, where forwarding decisions are made independently by each node (as opposed to specified by the source), they are suggest how directional antenna and worm-hole attacks can be used to deliver packets to multiple remote network positions, forcing packet processing at nodes that would not normally receive that packet at all, and thus increasing network-wide energy expenditure. Lastly, they are show how an adversary can target not only packet forwarding but also route and topology discovery phases—if discovery messages are flooded, an adversary can, for the cost of a single packet, consume energy at every node in the network.

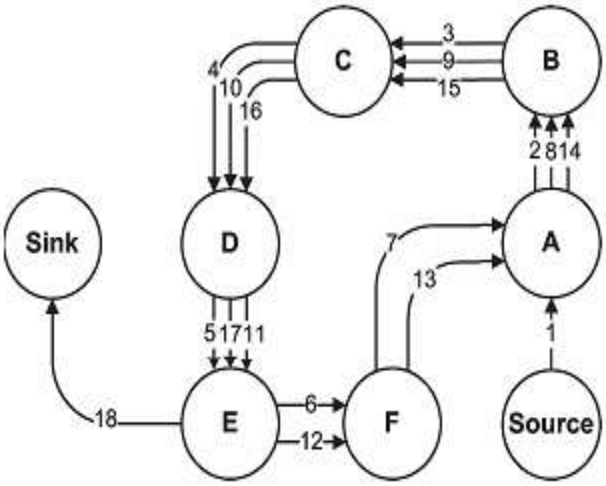


Fig 1.1 Malicious Route Carousel Attacks On Source Routing

In this first attack, an adversary composes packets with purposely introduced routing loops. They are calling it the carousel attack, since it sends packets in circles as shown in **Fig. 1.1.** It targets source routing protocols by exploiting the limited verification of message headers at forwarding nodes, allowing a single packet to repeatedly traverse the same set of nodes. Brief mentions of this attack can be found in other literature, but no intuition for defense or any evaluation is provided.

In this second attack, also targeting source routing, an adversary constructs artificially long routes, potentially traversing every node in the net-work. We call this the stretch attack, since it increases packet path lengths, causing packets to be processed by a number of nodes that is independent of hop count along the shortest path between the adversary and packet destination.

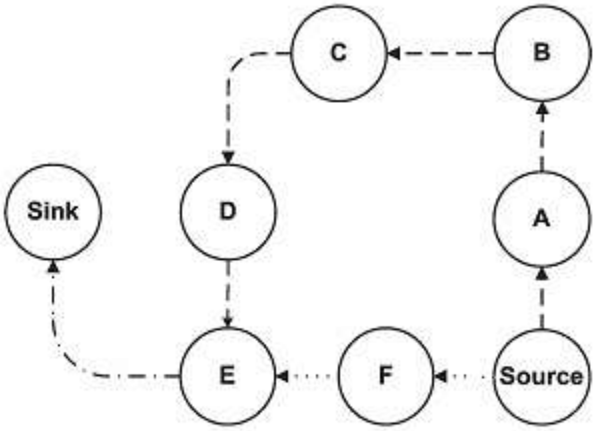


Fig. 1.2 Malicious Route Stretch Attack Attacks On Source Routing

An example is illustrated in **Fig. 1.2**. Results show that in a randomly generated topology, a single attacker can use a carousel attack to increase energy consumption by as much as a factor of 4, while stretch attacks increase energy usage by up to an or der of magnitude, depending on the position of the malicious node. The impact of these attacks can be further increased by combining t hem, increasing the number of adversarial nodes in the network, or simply sending more packets. Although in networks that do not employ authentication or only use end-to-end authentication, adversaries are free to replace routes in any overheard packets, we assume that only messages originated by adversaries may have maliciously composed routes

2.3.2 CPMP Overview

The CPMP (Content and Presence Multicast Protocol) is designed to support social content consumption experiences. CPMP provides a framework for periodically communicating information about what content is currently being consumed and what content is being sought for future consumption at each participating node.

The goal is to make the protocol efficient and scalable while including features intended to support synchronization of presence message transmissions. CPMP messages are transmitted periodically to inform nearby devices of updated content presence information using IP multicast. CPMP headers have the following format CPMP; device identifier; TX; containing a TX field specifying the number of seconds in which to expect a new CPMP message from the node specified by device identifier.

2.3.3 WBS: Weight Based Synchronization Algorithm

An algorithm is described first that uses the size of synchronization clusters as a catalyst for synchronization. The algorithm is called WBS—weight based synchronization. As mentioned previously, at the end of each active interval, a node uses the slotArray structure to decide its next transmission time. The slotArray structure has s entries, one for each slot of the next (sleep) interval.

WBS requires each node to locally maintain a variable monitoring the size of the cluster of synchronization which contains the node. The variable is called the weight of the node/cluster. Initially, the weight of each node is 1. Each node includes its weight in all its CPMP updates. Certainly, nodes cannot maintain globally accurate weights. Instead, each node needs to use only local knowledge—extractedfrom packets received from neighbors—to update the value of this variable.

```
Object implementation WBS extends GENERIC;
  maxW :int; #max weight over active interval
  weight :int; #weight advertised in CPMP packets
  OperationinitState()
    for (i:= 0; i < s; i ++) do
      slotArray[i] : = new pkt[]; od
  end
  OperationsetTX()
    #compute the maxW value
    maxW := 0;
    for (i:= 0; i < s; i ++) do
      for (j:= 0; j < slotArray[i]:size();j ++) do
        if (slotArray[i][j]:weight >maxW) then
          winnerSlot := i;
          maxW := slotArray[i][j]:weight; fi
    odod
    #determine new TX and weight values
    if (winnerSlot!= nextSendCPMP % ta) then
      TX:=winnerSlot;
      nextSendCPMP := tcurr þ TX;
      weight := maxW + 1;
    else
      weight := maxW;
    fi
  end
  OperationprocessPackets($t_{curr}$: int)
    pktList := inq:getAllPackets(slotLen);
    for (i:= 0; i <pktList:size();i ++) do
      index :=(($t_{curr}$ + pktList[i]:TX) mod $t_a$)/ts);
      slotArray[index]:add(pktList[i]);
    od
  end
```

2.3.4 Future Peak Detection Algorithm

The future peak detection algorithm is proposed to address the inflation attack. Instead of relying on subjective information (the weight value contained in CPMP updates), FPD allows nodes to build a local approximation of this metric, using only objective information derived from observation—the time of update receptions. FPD works by counting the number of packets that are stored in each slot of the current active interval.

```
Object implementation FPD extends WBS;
maxC :int; #max nr: of packets per slot
Operation setTX()
#compute the maxC value
maxC :¼ 0;
for (i := 0; i < s; i++) do
    if (slotArray[i]:size() >maxC) then
        maxC :¼ slotArray[i]:size();
        winnerSlot := i; fi
od
#update the TX value
if (winnerSlot! = nextSendCPMP % ta) then
    TX:=winnerSlot;
    nextSendCPMP := tcurr + TX;
fi
end
```

3. SYSTEM DESIGN

3.1 Module Design

The thesis contains the following modules.

1. Network Nodes Creation

Nodes details are added using this module. The node contains details such as Node Id, periodical updates sending information, next transmission time information and the time which it could be in sleeping mode. In addition, neighbor node details are also added in which start node id, end node id and distance are saved.

2. Carousel Attack

a. Scenario

In this module, the path is created such that the intermediate nodes modify the path information. One of the node is set as malicious and it alters the path information such that the packet again travels in the partial visited path. So cycle/loop occurs in the transmission.

b. Prevention

In this module, after the loop occurs, attested based scheme algorithm is worked out so that the packet contains signature which avoids the loop in the transmission.

3. Stretch Attack

a. Scenario

In this module, the path is created such that the intermediate nodes modify the path information. One of the nodes is set as malicious and it alters the path information such that it creates a new multi-hop partial path to reach its neighbor, so that the packets hop distance to destination is increased.

b. Prevention

In this module, after the stretch path scenario is found out, the loose source routing is applied so that all the nodes are made to find the alternate shortest available path to reach the destination. So stretch path attack is prevented.

4. Transmission Schedule Fixing

Here we use Content and Presence Multicast Protocol (CPMP) in which nodes use to send updates to their neighbors. The updates contain the relative time of their sender's next transmission.

5. Node Synchronization

In this module Future Peak Detection (FPD) algorithm is proposed. Nodes running FPD use CPMP updates to sync with their largest set of already synced neighbors, counting the number of packets received within a given interval and setting the node's next transmission to be in sync with the slot where most packets have been received. While lightweight and efficient, FPD's greedy strategy clusters the network: nodes reach a stable state without being synchronized with all their neighbors.

We address this issue using randomization. The Randomized Future Peak Detection (RFPD) algorithm is similar to FPD but uses a weighted probabilistic strategy to decide a node's next transmission time based on the packets received.

6. Suspicious Node Detection Based On Neighbor Nodes Behavior During Previous Transmissions

This module proposes the future peak detection algorithm to address the inflation attack. Instead of relying on subjective information (the weight value contained in CPMP updates), FPD allows nodes to build a local approximation of this metric, using only objective information derived from observation—the time of update receptions. FPD works by counting the number of packets that are stored in each slot of the current active interval. Note that each packet received during the current active interval is stored in the slot corresponding to the packet sender's next transmission time. The RFPD nodes do not propagate information (e.g., cluster sizes) thus preventing nodes from spreading inaccurate data and so suspicious node is avoided.

4. RESULT AND DISCUSSION

4.1. Experimental Analysis for Pr0posed

The following **Table 4.1** describes experimental result for existing system analysis. The table contains energy usage for Carousel attack Stretch attacks in sensor node detection are shown.

Table 4.1Energy Usage with Various Attacks for Existing System

S. No	Carousel Attack	Stretch Attack
1	0.03	0.06
2	0.07	0.14
3	0.12	0.21
4	0.23	0.27
5	0.25	0.34
6	0.29	0.39
7	0.33	0.42
8	0.45	0.51
9	0.53	0.62
10	0.63	0.69

The following **Fig 4.1** describes experimental result for existing system analysis. The table contains energy usage for Carousel attack Stretch attacks in sensor node detection are shown

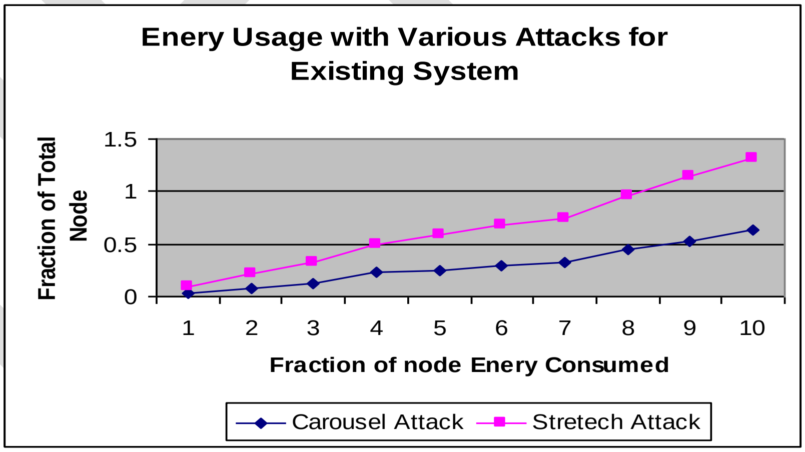


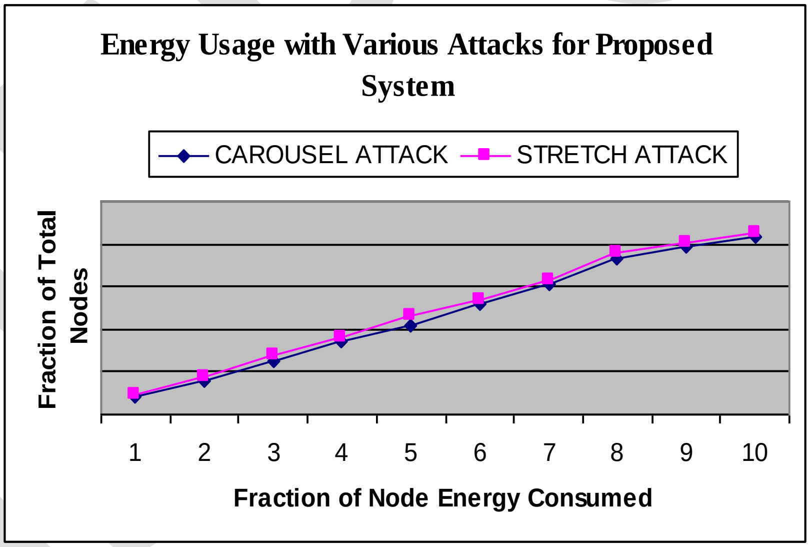
Fig 4.1Energy Usage with Various Attacks for Existing System

The following **Table 4.2** describes experimental result for proposed system analysis. The table contains energy usage for Carousel attack Stretch attacks in sensor node detection are shown

Table 4.2 Energy Usage with Various Attacks for Proposed System

S. No	Carousel attack	Stretch attack
1	0.08	0.09
2	0.16	0.18
3	0.25	0.28
4	0.34	0.36
5	0.42	0.46
6	0.52	0.54
7	0.61	0.63
8	0.73	0.76
9	0.79	0.81
10	0.83	0.85

The following **Fig 4.2** describes experimental result for proposed system analysis. The table contains energy usage for Carousel attack Stretch attacks in sensor node detection are shown



The following **Table 4.3** describes experimental result for proposed system performance analysis. The table contains set of synchronization node details; total number node sending packets and average percentage for existing and proposed system in sensor node detection are shown.

Table 4.3Experimental Performances For Existing And Proposed System

Set of Secure Synchronization Node	Total Number of Node	Existing System (%)	Proposed System (%)
{N1,N3,N6,N11}	4	52.34	55.22
{N2, N3, N4, N12, N13}	5	63.33	65.21
{N2, N5, N141, N8, N14, N12}	6	74.12	75.33
{N1, N8, N5, N13, N42, N5, N13}	7	83.11	84.67
{N1,N5, N3, N8, N12, N16,N20}	8	83.11	84.78
{N7, N8, N10, N12, N20}	5	66.44	68.36
{N1,N3,N6,N11}	4	57.33	60.11
{N17, N28, N20, N2, N1}	5	67.22	70.36
{N5, N8, N12, N15, N20}	5	69.22	72.45
{N4,N15, N13, N18, N12, N1}	6	78.22	80.22

The following **Fig 4.3** describes experimental result for proposed system performance analysis. The table contains set of synchronization node details, total number node sending packets and average percentage for existing and proposed system in sensor node detection are shown

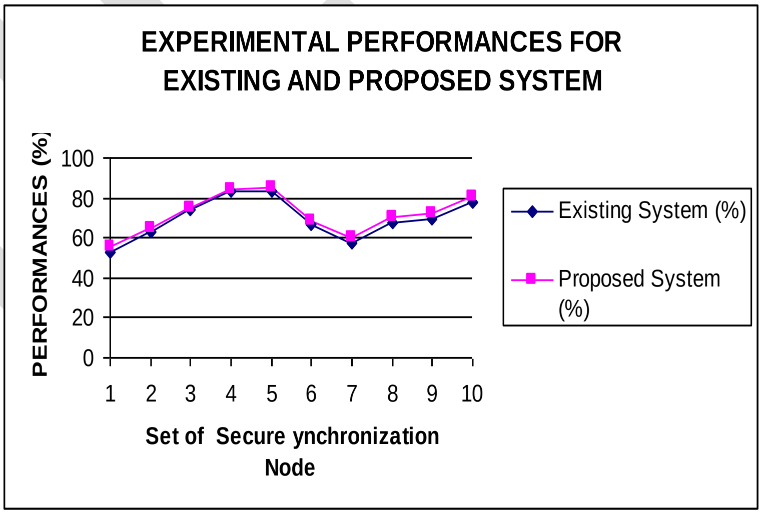


Fig 4.3Experimental Performances for Existing And Proposed System

5. CONCLUSION AND FUTURE ENHANCEMENTS

The central question addressed is how to effectively exploit secondary user co-operation when conventional cooperation method becomes inefficient. FLEC, a flexible channel cooperation design is proposed to allow SUs to customize the use of leased resources in order to maximize performance.

The problem of synchronizing the periodic transmissions of nodes in ad hoc networks, in order to enable battery lifetime extensions without missing neighbor's updates is studied. Several solutions, both lightweight and scalable but vulnerable to attacks is proposed.

Extension of generic algorithm to use transmission stability as a metric for synchronization is made. The implementation and simulations show that the protocols are computationally inexpensive, provide significant battery savings, are scalable and efficiently defend against attacks.

The application works well for given tasks in windows environment. Any node with .Net framework installed can execute the application. The underlying mechanism can be extended to any / all kind of web servers and even in multi-platform like Linux, Solaris and more.

The system eliminates the difficulties in the existing system. It is developed in a user-friendly manner. The system is very fast and any transaction can be viewed or retaken at any level.

REFERENCES:

[1] Aad, J.-P. Hubaux, and E.W. Knightly, "Denial of Service Resilience in Ad Hoc Networks," Proc. ACM MobiCom, 2004.

[2] G. Acs, L. Buttyan, and I. Vajda, "Provably Secure On-Demand Source Routing in Mobile Ad Hoc Networks," IEEE Trans. Mobile Computing, vol. 5, no. 11, pp. 1533-1546, Nov. 2006.

[3] T. Aura, "Dos-Resistant Authentication with Client Puzzles," Proc. Int'l Workshop Security Protocols, 2001.

[4] J. Deng, R. Han, and S. Mishra, "Defending against Path-Based DoS Attacks in Wireless Sensor Networks," Proc. ACM Workshop Security of Ad Hoc and Sensor Networks, 2005.

[5] J. Deng, R. Han, and S. Mishra, "INSENS: Intrusion-Tolerant Routing for Wireless Sensor Networks," Computer Comm., vol. 29, no. 2, pp. 216-230, 2006.

[6] S. Dosh i, S. Bhandare, a nd T.X. Brow n, "An On-Demand Minimum Energy R outing Protocol for a Wireless Ad Hoc Network," ACM SIGMOBILE Mobile Computing and Comm. Rev., vol. 6, no. 3, pp. 50-66, 2002.

[7] J.R. Douceur, “The Sybil Attack,” Proc. Int’l Workshop Peer-to-Peer Systems, 2002.

[8] H. Eberle, A. Wander, N. Gura, C.-S. Sheueling, and V. Gupta, “Architectural Extensions for Elliptic Curve Cryptography over GF(2m) on 8-bit Microprocessors,” Proc. IEEE Int’l Conf’ Application-Specific Systems, Architecture Processors (ASAP), 2005.

[9] L.M. Feeney, “An Energy Consumption Model for Performance Analysis of Routing Protocols for Mobile Ad Hoc Networks,” Mobile Networks and Applications, vol. 6, no. 3, pp. 239-249, 2001.

[10] J. Deng, R. Han, and S. Mishra, “Defending against Path-Based DoS Attacks in Wireless Sensor Networks,” Proc. ACM Workshop Security of Ad Hoc and Sensor Networks, 2005.

[11] D.B. Johnson, D.A. Maltz, and J. Broch, “DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks,” Ad Hoc Networking, Addison-Wesley, 2001.

[12] R. Govindan and A. Reddy, “An Analysis of Internet Inter- Domain Topology and Route Stability,” Proc. IEEE INFOCOM, 1997.

[13] J.W. Bos, D.A. Osvik, and D. Stefan, “Fast Implementations of AES on Various Platforms,” Cryptology ePrint Archive, Report 2009/ 501, http://eprint.iacr.org, 2009.

[14] H. Eberle, A. Wander, N. Gura, C.-S. Sheueling, and V. Gupta, “Architectural Extensions for Elliptic Curve Cryptography over GF (2m) on 8-bit Microprocessors,” Proc. IEEE Int’l Conf’ Application-Specific Systems, Architecture Processors (ASAP), 2005.

[15] Y.-C. Hu, A. Perrig, and D. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. In Proceedings of the ACM Conference on Mobile Computing and Networking (Mobicom), 2002