Volume 2, No. 4, July-August 2011
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
TECHNICAL NOTE
Available Online at www.ijarcs.info

Analyzing and Managing the Loss of Packets in Sensor Networks

Sachin Bhardwaj*
Mtech Scholar Department of Computer Science & Applications, CDLU, Sirsa,
Haryana, India
sachinbhardwaj43@gmail.com

Harish Rohil
Assistant Professor Department of Computer Science & Applications, CDLU, Sirsa,
Haryana, India
harishrohil@gmail.com

Abstract: In large sensor area networks with dense sensors there are some nodes in that network that have to bear a heavy traffic load. This load is bearable up to some threshold value but as the time passes the sensor network goes weak due to heavy load and nodes start losing packets. Thus to minimize the loss we need to do load balancing in which the load will be shared by the other low energy nodes to minimize the load of that node and thus minimize the loss of packets.

Keywords: DSR, MAC, SENSOR NETWORK.

I. INTRODUCTION

Recent technological developments in micro-electromechanical (MEMS) systems, wireless communications and digital electronics present a new trend for the rapid advances that will follow in the near future: complete systems on a microscopic chip, integrated low-power communication, and integrated low-power transducers at an extremely low cost. A new, post-PC era is taking shape with functionality being pushed into smaller, cheaper, lower-power units in respect to traditional desktop and server platforms. The future systems are envisioned to be deeply embedded into the physical environment, with capabilities of sensing it, perhaps even powered by ambient energy, and used in many smart space scenarios. Sensor technology is already capable of interacting with various fields and forces to detect light, heat, position, movement, chemical presence, and so on. In each of these areas, the technology is crossing a critical threshold that makes the operation of networked sensors an exciting research area. We envision that, in the near future, wireless sensor networks will become an integral part of our lives, maybe more so than today's personal computers. The central idea is to operate a system consisting of a vast number of sensor devices that integrate sensing with wireless network interfaces that collect and disseminate information about the physical environment. The system is deployed in areas of interest (ranging from homes, schools and universities to inaccessible terrains, disaster places, etc.) making those smart spaces where fine grained monitoring services and applications can be provided[7].

Current system solutions, protocol frameworks and paradigms typically provide the following services:

A. Periodic Sensing (the sensor devices constantly monitor the physical environment and continuously report their sensors' measurements to a control center),
B. Event driven (to reduce energy consumption, sensor devices monitor silently the environment and
C. communicate to report when certain events are realized) and
D. Query based (sensor devices respond to queries made by a supervising control center).

Recently, new applications have been proposed, that require different approaches for disseminating sensor data to the control center, such as Target Tracking (where sensors exchange sensor readings in order to detect the movement pattern of a detected target) or Area Surveillance (where sensors are equipped with video capturing devices). Certainly, more services will become feasible in the near future that will allow different kinds of interaction with the surrounding environment (e.g. via actuators and servo mechanisms). It should not be surprising that the unique characteristics of this regime give rise to very different design trade-offs than current general-purpose systems. The missing elements are simple but efficient optimization strategies at the protocol level, overall system architecture and a methodology for systematic advance. Indeed, the realization of such efficient, robust and secure ad-hoc networking environments is a challenging algorithmic, systems and technological task. Large numbers of such tiny and resource-constrained devices should self-organize into an ad-hoc network under highly dynamic ambient conditions, carrying out computations locally and engaging into a collaborative computing and communication effort. The required solutions differ significantly, not only with respect to classic distributed computing but also with respect to ad-hoc networking [10]. To further emphasize on the difference consider that

A. the number of interacting devices in a sensor network is extremely large compared to that in a typical ad-hoc network,
B. sensor networks are typically prone to faults (as a result of the low cost equipment) and
C. The limitations in energy, computational power and memory are much more severe in sensor networks.

In this sense, new models must be provided, novel methodologies should be thought of, integrated (but flexible) networking and software architectures should be designed and implemented, efficient algorithmic solutions must be devised, and integrated environments for application development are needed[2].

II. PROPOSED SYSTEM

The detection platform is intended to identify and recognize attacks focusing on a target it is guarding. The platform performs this task by constant monitoring the traffic into and out of the target and additional parameters.

Different systems have different classifications for attacks, and are calibrated differently to recognize them. For example, a system monitoring an HTTP server will focus on malicious DoS attacks while a system monitoring a small network will attempt to identify an unauthorized intruder and any misuse of a computer system.

Advanced detection systems are only the first phase of a defense from such an attack. After identifying an attack and recognizing it, the protection phase must make adjustments in order to protect the intended target. The protection phase is completely separate from the detection phase, yet for successful and quick defense the detection system must not only be capable of identifying quickly that an attack is taking place and categorizing it, but it would be helpful if the detection system could recognize and pinpoint parameters unique to the attackers and that differ from "innocent" users. Fig 1 in section 3 shows the proposed system flow chart.

A. Detection system location:

Detection systems tend to be constructed as either "host based" or "network systems". Each one of these architectures has advantages and shortcomings. Neither one can be defined as "better", yet different protections call for different system architecture.

B. Host based Systems:

These systems usually work off audit logs provided by the operating system. The system detects attacks by watching for suspicious patterns of activity on the host. This system can learn quite quickly the different patterns of use in the system and recognize any abnormalities that appear during an attack.

The system has the advantage of access to the Innermost processes in the host, and can notice any slight change that occurs (for example – access to kernel activities). In addition, since the system sits physically on the host, it can receive real time information about the host's resources during peak activities (such as occurs during an attack). This is important in a situation where due to a crippling amount of packets that arrive to the host, the host discards some of them and responds only to a small amount. The only way to know that the host is discarding some of these packets is by direct access to the innermost processes in the host.

However, the "Host based" systems have a major shortcoming: they are only aware of what enters the host, and have no clue about low level network events. Since the "Host-based" systems are autonomous, they have no idea regarding the state their neighboring computers are in and rarely share information on a regular basis in order to enable enhanced protection and detection. An example for such correlated detection is several computers noticing a port scan being executed on them (extensive port scanning might warn against an upcoming attack)

C. Network Detection:

Network systems are driven off interpretation of raw network traffic. They watch traffic on the network and try to detect attacks by watching for specific patterns or abnormalities in network traffic. The systems work by examining the contents of packets transmitted on the network analyzing the types of protocols used and different packet attributes. This is usually done passively by eavesdropping on the network using a sniffer or any similar type of tool. This type of analysis is unobtrusive and at the lowest levels of network operation, extremely difficult to evade. An installation of such a system does not require any network adjustments and does not degrade the network performance in any way.

Network detection systems are good at noticing low level network manipulations of the network and can identify correlated attacks against several targets. An important advantage is the ability to recognize attacks focused on the network itself and not at a specific target (overloading a network with packets to a nonexistent machine for example).

III. DETECTION PARAMETERS

There are many approaches to identifying a DoS attack and yet after reviewing many of them, one can notice that there are several detection parameters that are considered in the majority of the systems. The weight given to each parameter varies from system to system but important detection parameters are always used. In this part we will review these more common parameters, their strengths and weaknesses. The effectiveness of the detection parameters varies from system to system. Some equipment tends to be more stable than others and at times other equipment might have a better history that enables finer tuning for detection.

A. Load and Traffic Monitoring:

Load and traffic volume monitoring at ISPs can provide early warning of attacks. Traffic-limiting IDS can monitor loads of all incoming traffic and search for abnormalities. In addition, the system might also attempt to reframe data communications between two points by asking the sender to slow down the rate of data acknowledgment. Legitimate servers will do so. Those that don't are deemed untrustworthy, so their packets are then filtered out. This method is mainly effective against "script kiddies" who work within Microsoft Windows and download hacking scripts from the Internet. Such hackers don't know sophisticated methods of concealing their IP addresses. In theory, a traffic-limiting device installed outside the firewall should strip out and redirect bad traffic without becoming a choke point for good traffic. It could also deny inbound data from specified IP addresses, either for a set time or until an attack stops. The denial automatically ends when traffic flow returns to normal.

B. Latency to Victim:

Checking the time it takes the system to respond to requests is a good indicator (assuming otherwise the system works well). The first way to implement such a monitor is to construct an agent placed on a different network from the potential target, and having the agent constantly send requests to the potential target. The agent measures the average time of response, and when a big deviation from this time is identified, the alarms go off. This method is nicknamed 'what's up? 'The second method is to have the potential target send a test packet to some outer agent that simply resends the packet back. When done constantly, the potential target can learn when the inbound bandwidth becomes congested and can sound an alarm.

C. Committed Access Rate (CAR):

This method checks if a specific type of packet uses up more than average amount of bandwidth it usually does. This idea is derived from a defense method, in which the router will limit the bandwidth consumed by certain types of traffic (configurable via an extended access control list). This can be used to limit the bandwidth consumed by SYN packets, so that non-SYN packets (i.e., legitimate established connections) will have bandwidth available. The downside to this approach is that it will be difficult for a legitimate client to establish a new connection while the target is under attack. One technique using CAR is to permit unrestricted access to a specific set of known critical clients and apply CAR to others.

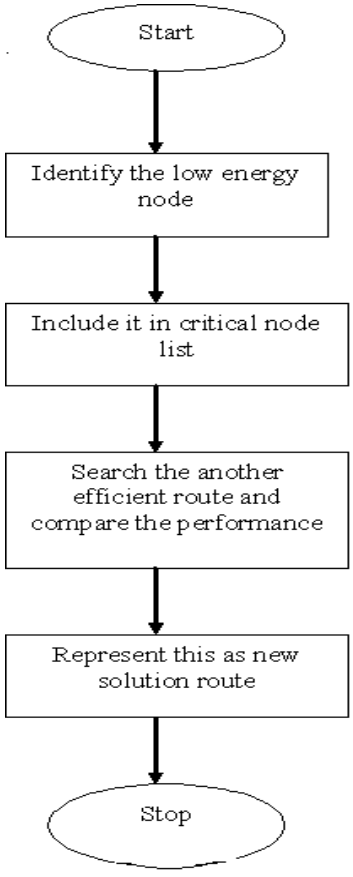

Figure1.Flowchart to find the solution route

IV. HOW DETECTION SYSTEM WORKS

The following figure2 describes the detection system flow indicating inputs, internal information flow and output. Bear in mind that the thread work concurrently and the flow demonstrate only the logical path of the information through the system.

System internal flow consists of the following stages:

A. Collector daemons constantly collect incoming statistics. Periodically the collector threads query the daemons for the current statistics.

B. The sampled statistics received by the collector threads are committed to the database, normalized by time.

C. The post collector periodically samples the database for raw statistics samples and estimates the probability for common events such as spoofed traffic or changes in traffic behavior such as changes in packet size, TCP/UDP destination ports distribution and etc.

D. The post collector commits the estimations in the database.

E. The analyzers periodically sample the database for estimations and raw statistics and evaluate the probability for an attack. Attack evaluation are written to log files or printed to the screen upon user request.

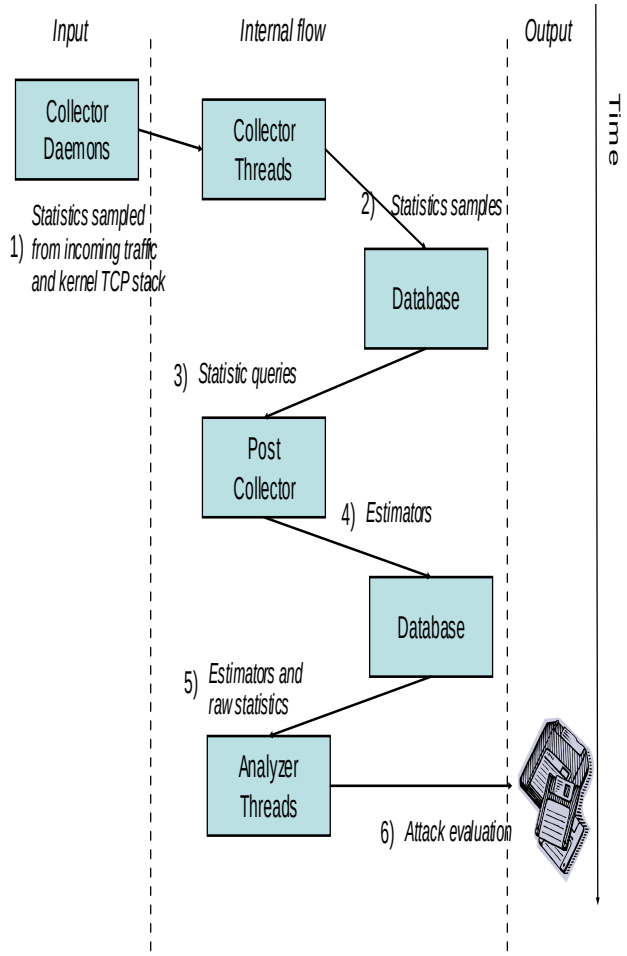

Figure2: How detection system works.

V. ALGORITHM

With a set of selected detecting paths, the detecting algorithm will probe over each of them. Given a detecting path, there are at least two ways of probing. One way is to probe from the farthest node to the nearest. The other way is to probe from the nearest node to the farthest. Each has its own advantages and disadvantages. Detecting from far to near is better if the detecting path is GOOD since it takes only one probe message and proves the goodness of all the intermediate nodes. But it may take more probe messages if a MALICIOUS node is located near the detecting node. This method can be applied to a network where we have the confidence that the majority of the nodes in the network are GOOD. The advantage of probing from near to far is that it generates smaller number of probing messages to detect a MALICIOUS node located near the probing node. Another advantage is that we have the prior knowledge of the states of all the intermediate nodes along the path to the probed node except its immediate predecessor node. The disadvantage is an intelligent attacker may be able to avoid detection by forwarding all packets (including probe messages destined to the downstream nodes) for a certain period of time immediately after receiving a probe message for itself. A received probe message therefore serves as a signature to an attacker that a diagnosis process is ongoing, and it would start to behave normally for a short period of time. Other search strategy (e.g., binary search) can also be deployed to reduce network overhead. In this paper, we present the algorithm for the first method, probing from the farthest nodes to the nearest, since it is stronger than the other alternatives in detecting malicious nodes. For a probing path, the probing node sends a probe message to the farthest node. If an acknowledgment message is received within a certain period of time, all the intermediate nodes are shown to be GOOD.

Otherwise, a probe message is sent to the second farthest node. This process is repeated until one node responds to the probe message or the nearest node (a neighbor node) is probed and it is not responsive. In the latter case, we know that the neighbor node in the probed path either is DOWN or has moved out to another location. Since the neighbor node is not responsive, there is nothing we can do to monitor the rest nodes in the path [2]. Therefore, probing over this path is stopped. If an intermediate node is responsive but a node subsequent to it is not, it is possible:

DOSDetectopm(S,D)

/* S is the source node and D represents the Destination Node over the network*/

{

A. As transmission begins it will search for all the intermediate nodes and send data on to it.
B. The intermediate node failed forwarding the probe message to the next node;
C. it will check the RESPONSE time for the intermediate node

If(ResponseTime>HopTime+Threshold)

{

The Attacker Node is Detected.

Update Neighbor Node Table & Routing Table for the Intermediate Nodes

}

D. The unresponsive node is incapable of responding to the probe message.
E. The diagnosis algorithm will then be called to decide which one is the case.

VI. CONCLUSIONS

The given purposed research will provide the solution of packet loss in case of any one weak sensor node over the sensor network. The purposed system will first detect the weak sensor node over the network and then block it or set its load to the minimum. Now instead of transferring data on this node, it will pass on from the surrounding nodes; it will only handle the transmission that is directed to it only. The algorithm will provide the better solution for reducing the packet loss in case of some weak nodes over the network [2].

VII. REFERENCES

[1]. K Whitehose, D Culler Calibration as Parameter Estimation in Sensor Networks [C].In: FirstACM International Workshop on Wireless Sensor Networks and Application, Proceedings of IEEE GLOBECOM '01, 2001-11.Atlanta GA, 2002-09.

[2]. Chris Savarese. Robust Positioning Algorithms for Distributed Ad-Hoc WirelessSensor Networks(Mater's Thesis). University of California at Berkeley,2002. Vargar A. OMNET++ Version 2.2.UserManual. http://www.hit.bme.hu, 2002.

[3]. Want R, Hopper A, Falcao V, Gibbons J. , Self-Configuring localization systems [Ph.D. Thesis]. Los Angeles: University of California, 2002.

[4]. Hightower J, Borriello G. Location sensing techniques. Technical Report UW CSE 2001-07-30, Seattle: Department of Computer Science and Engineering, University of Washington, 2001.

[5]. Manjeshwar A, Agrawal D P. TEEN: A routing protocol for enhanced efficiency in wireless sensor networks. In: Proc 15th Int'l Parallel and Distributed Processing Symp (IPDPS'01), San Francisco, CA. 23-27 April, 2001.

[6]. Ye W, Heidemann J, Estrin D, An energy-efficient MAC protocol for wireless sensor networks. In: Proc 21St Int'l Annual Joint Conf IEEE Computer and Communications Societies (INFCOM 2002), New York, NY, June 2002.

[7]. N. Asokan, P. Ginzboorg. Key Agreement in Ad- hoc Networks. Computer Communications,23:1627- 1637, 2000.

[8]. Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu and Jorjeta Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. Proceedings of MobiCom'98, Dallas, TX, October 1998.

[9]. Sonja Buchegger, Jean-Yves Le Boudec. Nodes Bearing Grudges : Towards Routing Security, Fairness, and Robustness in Mobile Ad hoc Networks. In Proceedings of the 10thEuromicro Workshop on Parallel, Distributed and Network-based Processing, 2002.

[10]. Shayan Ghazizadeh, Okhtay Ilghami, Evren Sirin, Fusun Yaman. Security-Aware Adaptive Dynamic Source Routing Protocol. In Proceedings of the 27th Annual IEEE Conference on Local Computer Networks(LCN'02),2002.