International Journal of Science and Research (IJSR)
ISSN (Online): 26021501
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Localization Technique in Wireless Sensor Network

M. Akshaya[1], M. RajKamal[2]

[1]PG Student Anand Institute of Higher Technology, Kazhipattur, Chennai-603103, India

[2]Assistant Professor Anand Institute of Higher Technology, Kazhipattur, Chennai-603103, India

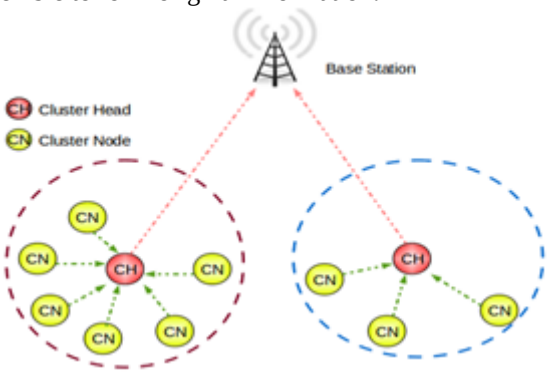
Abstract: ***Localization is a fundamental issue of wireless sensor networks. Localization in wide areas remains very challenging due to various interfering factors. Combined & Differentiated Localization approach that exploits the strength of range free approaches and range-based approaches using received signal strength indicator. A critical observation is that ranging quality greatly impacts the overall localization accuracy. To achieve a better ranging quality virtual-hop localization is used. For the first phase of CDL, virtual-hop localization initially computes node locations. This is an enhanced version of hop count- based localization. Virtual-hop particularly addresses the issue of non uniform deployment. The data transfer takes place between the network by using IP address. External Agents an IP-based host. An EA node is a full-fledged computer with full TCP/IP networking stack or IP network. Sensor Nodes provide data that is requested by EA.. It has both IP network stack and DD sensor network capability. Data returned by sensor nodes is also processed and forwarded by gateway node to IP-based hosts. The data transfer between the nodes is not assured full data packet transfer to the destination due to various factors such as heavy load information, source-destination length. To overcome this drawback load balancing algorithm is implemented.***

Keywords: Localization, wireless sensor network, nodes, master and slave, load balancing algorithm, IP address.

1. Introduction

To overcome the data packet loss in virtual hopping localization technique, load balancing algorithm is being used.

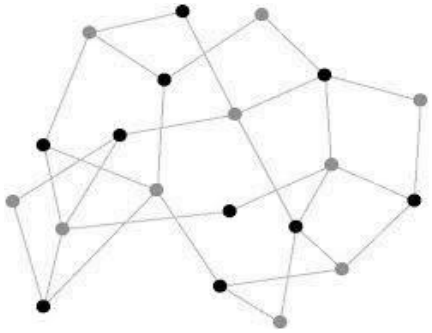
This technique increases the efficiency of data transfer by transferring the data's equally in all nodes with respect to load so that when they reach the destination all the data's reassemble to form original information**.**



2. Block Diagram

Nodes in communication networks, a node is either a connection point, a redistribution point or a communication endpoint (some terminal equipment). The definition of a node depends on the network and protocol layer referred to. A physical network node is an active electronic device that is attached to a network, and is capable of sending, receiving, or forwarding information over a communications channel. A passive distribution point such as a distribution frame or patch panel is consequently not a node.

A "sleepy node" is a device that episodically from time-to-time puts itself in a low-power, quiescent state, or "sleeps". While sleeping, these devices cannot perform major functions, such as transmitting



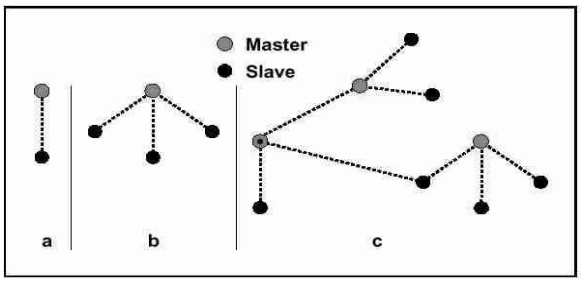
Cluster of Nodes

Packets, receiving packets or performing computations. Often, these devices wake on the expiration of a timer, although some devices are capable of waking in response to an external event. A sleep state differs from a power-down state in that most device state is preserved; therefore, the transition from a sleep state to an operating state requires much less time than does booting from a powered-down state. Generally, these devices sleep to conserve energy.

An active/active system is a network of independent processing nodes, each having access to a common replicated database, so that all nodes participate in a common application. In a general case, the nodes are completely symmetric; any transaction can be routed within the application network to any node which reads or updates any set of data items in the database. This approach provides the most flexibility and maximizes system investment as requests are load-balanced across all available processing capacity. If a node fails, users at the other nodes are unaffected. Furthermore, the users at the failed node are quickly switched to surviving nodes, thus restoring their services in seconds or less.

Master/slave is a model of communication where one device or process has unidirectional control over one or more other devices. In some systems a master is elected from a group of

eligible devices, with the other devices acting in the role of slaves.



1.2 Interconnection of Master and Slave Node

In other words "The master/slave configuration is basically used for load sharing purposes when 2 identical motors connected to two different drives are coupled to a common.

B) Experimental Setup

Load balancing differs from channel bonding in that load balancing divides traffic between network interfaces on a network socketbasis, while channel bonding implies a division of traffic between physical interfaces at a lower level, either per packet. The most commonly used applications of load balancing is to provide a single Internet service from multiple servers. Commonly load-balanced systems include popular web sites, large Internet Relay Chat networks, high-bandwidth File Transfer Protocol sites, Network News Transfer Protocol servers and Domain Name System servers. Lately, some load balancers have evolved to support databases; these are called database load balancers.

1) Round-robin DNS An alternate method of load balancing, which does not necessarily require a dedicated software or hardware node, is called round robin DNS. In this technique, multiple IP addresses are associated with a single domain name; clients are expected to choose which server to connect to. Unlike the use of a dedicated load balancer, this technique exposes to clients the existence of multiple backend servers. The technique has other advantages and disadvantages, depending on the degree of control over the DNS server and the granularity of load balancing desired.

Round Robin: It works well in most configurations, especially in processing speed and memory.Dynamic Ratio: The Dynamic Ratio methods are used specifically for load balancing traffic to Real Networks, Real System Server platforms and Windows. Fastest (node): The Fastest methods are useful in environments where nodes are distributed across separate logical networks.

Hardware and software load balancers may have a variety of special features. The fundamental feature of a load balancer is to be able to distribute incoming requests over a number of backend servers in the cluster according to a scheduling algorithm. Most of the following features are vendor specific:
Asymmetric load: A ratio can be manually assigned to cause some backend servers to get a greater share of the workload than others. This is sometimes used as a crude way to account for some servers having more capacity than others and may not always work as desired.

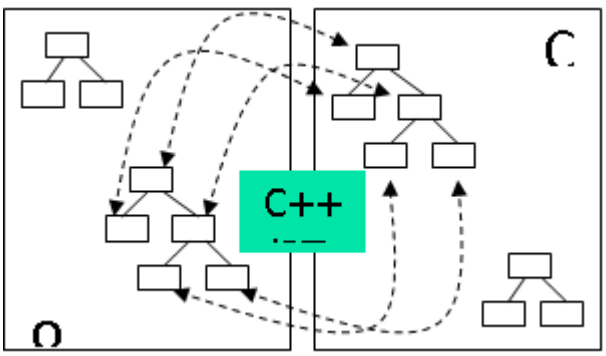
Priority activation: When the number of available servers drops below a certain number, or load gets too high, standby servers can be brought online. TCP offload: different vendors use different terms for this, but the idea is that normally each HTTP request from each client is a different TCP connection. This feature utilizes HTTP to consolidate multiple HTTP requests from multiple clients into a single TCP socket to the back-end servers.

TCP buffering: the load balancer can buffer responses from the server and spoon-feed the data out to slow clients, allowing the web server to free a thread for other tasks faster than it would if it had to send the entire request to the client directly. Priority queuing: also known as rate shaping, the ability to give different priority to different traffic. Content-aware switching: most load balancers can send requests to different servers based on the URL being requested, assuming the request is not encrypted or if it is encrypted that the HTTPS request is terminated at the load balancer.

Programmatic traffic manipulation: at least one balancer allows the use of a scripting language to allow custom balancing methods, arbitrary traffic manipulations, and more. Firewall: direct connections to backend servers are prevented, for network security reasons Firewall is a set of rules that decide whether the traffic may pass through an interface or not.

3. Software Used

Ns is a discrete event simulator targeted at networking research provided by USC/ISI [NS2]. It models system as events, which the simulator has, list of. The process is made such way: “take next one, run it, until done”. Each event happens in an instant of virtual (simulated) time, but takes an arbitrary amount of real time. The design of the simulator is separating the “data” from the control: C++ for “data” (per packet processing, core of ns, fast to run, detailed, complete control); and OTcl for control.



Ns functionalities comprise routing, transportation (TCP and UDP), traffic sources (web, ftp, telnet…), queuing disciplines, QoS (IntServ and Diffserv) and emulation for the wired systems. For the wireless part, ns provides ad hoc routing and mobile IP. Ns provides also traffic and topology generators and simple trace analysis.

First ‘ns-allinone-2.34’ package is installed in Red Hat Linux and the path is set as path=$path/home/ns-allinone-2.34/bin. Then NS is started with the command 'ns <tclscript>‘where '<tclscript>' is the name of a TCL script file which defines the simulation scenario (i.e. the topology and the events). Everything else depends on the TCL script. The script might create some output on std out. Starting nsFirst ‘ns-allinone-2.34’ package is installed in Fedora version 9.0 and the path is set as path=$path/home/ns-allinone-2.34/bin. Then NS is started with the command 'ns <tclscript>‘where '<tclscript>' is the name of a TCL script file which defines the simulation scenario (i.e. the topology and the events).

Everything else depends on the TCL script. The script might create some output on stdout. It might write a trace file or it might start NAM to visualize the simulation. Startingnam When a simulation is finished, NS produces one or more text-based output files that contain detailed simulation data, if specified to do so in the input script. The data can be used for simulation analysis or as an input to a graphical simulation display tool called NAM. NAM has a nice graphical user interface. It can graphically present information such as throughput and number of packet drops at each link NAM is started with the command 'nam<nam-file>' where '<nam-file>' is the name of a NAM trace file that was generated by NS, or it can execute it directly out of the TCL simulation script for the simulation to visualize.

Red Hat Linux, assembled by the company Red Hat, was a popular Linux based operating system until its discontinuation in 2004. Early releases of Red Hat Linux were called Red Hat Commercial Linux; Red Hat first published the software on November 3, 1994. It was the first Linux distribution to use the RPM Package Manager as its packaging format, and over time has served as the starting point for several other distributions, such as Mandriva Linux and Yellow Dog Linux. In 2003 Red Hat discontinued the Red Hat Linux line in favor of Red Hat Enterprise Linux (RHEL) for enterprise environments. Fedora, developed by the community-supported Fedora Project and sponsored by Red Hat, is the free version best suited for home use.

4. Experimental Analysis

Besides gateway-based approaches to interconnecting wireless sensor and IP networks, there also exists IP-enabled WSNs. One of these approaches assumes a full TCP/IP stack on each sensor node. In this approach, the WSN is directly connected to the IP network to enable direct communication between WSN sensor nodes and IP-based hosts.

The main advantage of using TCP/IP in this way is that there is no need for protocol conversion or gateways. However, the overhead for the full networking stack on an energy-constrained sensing device may be prohibitive, especially when the end-to-end retransmissions incurred by the TCP protocol cause even more undue retransmissions at intermediate nodes. It has been shown that the majority of energy in a WSN is used for wireless communication.

IP hosts must be supplied with each individual WSN node IP address it wishes to query for data. There could potentially be many WSN nodes in possibly multiple remote WSNs so this may be an inefficient method of information retrieval, especially considering the wasted energy with TCP retransmission attempts. This approach also uses IP routing algorithms which are proactive and less energy-efficient than reactive routing algorithms, such as directed diffusion, used in WSNs.

In overlay approaches, gateway nodes are used to interconnect WSNs with IP networks and assign virtual identification information to either IP-based hosts, sensor nodes, or both. According to overlay approaches come in two basic forms: sensor network overlay IP network and IP network overlay sensor network. These two approaches employ application layer gateways through which the WSN is identified and information is passed.

In the sensor network overlay IP network approach, IP-based hosts are required to register with the WSN application-layer gateway node and be assigned a virtual sensor node ID by the gateway node. Once a packet from a sensor node destined for a virtual sensor node ID reaches the gateway node, the gateway node encapsulates the whole packet into a TCP or UDP and IP packet, while the IP-based host communicates with sensor nodes by supplying the sensor node ID to the gateway node.

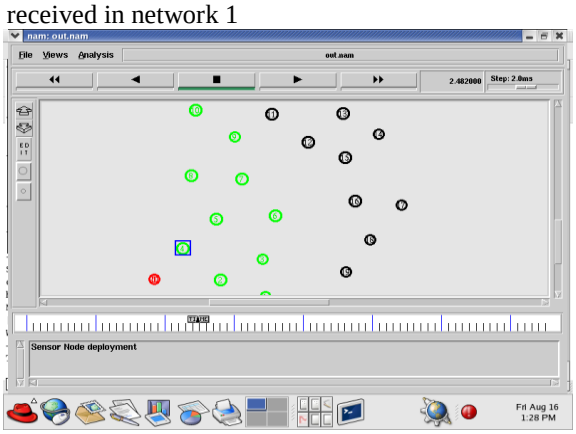
In the IP network overlay sensor network, sensor nodes are required to register with the WSN gateway node and are assigned a virtual IP address. Individual sensors themselves do not actually possess an IP address in the WSN. Sensor nodes are instead assigned a WSN-wide unique standard 16-bit TCP/UDP port number by the gateway node. IP-based hosts communicate with individual WSN nodes by supplying the IP address of the gateway node and port of the sensor node with which it wishes to communicate.

System uses an approach for providing seamless interconnection and transparent interoperability between different sensor data dissemination paradigms of IP and WSNs via gateways which also decouple the IP networks and WSNs, allowing for specialized and more efficient protocols to be implemented in WSNs.

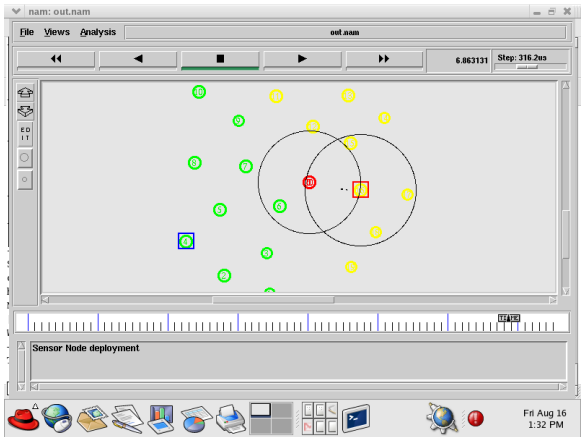
There are three node roles in the network. (i)External Agents: An IP-based host that is not a part of the WSN is termed an external agent (EA). An EA node is a full-fledged computer with full TCP/IP networking stack and can access the Internet or IP network. (ii)Sensor Nodes: Sensor nodes provide data that is requested by EA. Given the attributes of robustness, scalability, and energy-efficiency in multi-hop communication directed diffusion (DD) is used as routing protocol within sensor nodes. A middleware layer, dynamic service (DS), is designed on top of DD. DS will be introduced in Section. (iii)Gateway Node: gateway node is on the boundary of WSN and directs incoming and outgoing traffic of WSN. It has both IP network stack and DD sensor network capability. Any interest or subscription from an IP-based host is processed through the gateway node, translated into WSN interest, and disseminated using DD protocol. Data returned by sensor nodes is also processed and forwarded by gateway node to IP-based hosts.

5. Experimental Results

The below screen shot implies that node is created in the network. Initially node will be in sleeping mode before data gets received in network 1



After data received in network 1. The node changes from sleeping mode to active mode.



The above result indicates that Master receives data from slave node in network1. Packet transmission takes place between two networks by using elevator node. Elevator node which is also called as collector node. It collect data from network1 and transmit data to network 2. Data received by master node in network.

6. Conclusion

In this project presents an approach for seamless interconnection between IP networks and WSNs, whereby IP-based hosts can access and manipulate IP-enabled WSNs. This approach uses common dynamic services for transparent communication with IP-enabled WSNs that allows IP-based hosts to easily task and harvest data from remote dynamic services enabled directed diffusion wireless sensor networks. APIs for both IP-based and sensor node application programmers are presented. A description of an application-layer gateway has been given which is used to enable IP-based hosts to gather data from one or more remote WSNs. Future work could follow many different paths. Currently only EAs submitting interests to WSNs are supported. DD is a best effort service but essentially does not guarantee delivery of data. The DS API could be extended to ensure guaranteed delivery of data to other sensor nodes. The current EA API only supports UDP. Extensions could be made to the API that also allow for TCP connections between the EA and the remote gateway node.

References

[1] "GPS-less Low-Cost Outdoor Localization for Very Small Devices" NirupamaBulusu, John Heidemann, and Deborah EstrinUniversity of Southern California/Information Sciences Institute (2013).
[2] "Network Traffic Classification Using Correlation Information " Issue No.01 - Jan. (2013vol.24)
[3] "Performance evaluation of two layered mobility management using mobile IP and session initiation protocol" (2010)
[4] "Perpendicular Intersection: Locating Wireless Sensors With Mobile Beacon" (2011) ZhongwenGuo, Member, IEEE, Ying Guo, Student Member, IEEE, Feng Hong, Member, IEEE, Zongke Jin, Yuan He, Student Member, IEEE, Yuan Feng, Member, IEEE, and Yunhao Liu, Senior Member, IEEE.
[5] "Simulation of IP Packet Classification on Network Simulator (NS2)" Fatin G. Sabbagha Yang Li Jinzhu Chen Department of Computer Sciences and Engineering Michigan State University East Lansing, Michigan 48823, U.S.A (2013)

Author Profile

M. Akshaya obtained B.E Electronics and Communication Engineering in SMK Fomra institute of technology in 2013 and pursuing ME Embedded System Technologies in Anand Institute of Higher Technology affiliated to Anna University.