CiSJ

computing@tanet.edu.te.ua
www.tanet.edu.te.ua/computing

ISSN 1727-6209
International Scientific
Journal of Computing

NETWORKING PARADIGM FOR DISTRIBUTED SENSOR NETWORKS

S. Sitharama Iyengar [1], Qishi Wu [2], Nageswara S.V. Rao [2]

1) 298 Coates Hall, Louisiana State University, Baton Rouge, LA 70803, iyengar@bit.csc.lsu.edu,
http://bit.csc.lsu.edu/~iyengar/
2) Center for Engineering Science Advanced Research, Computer Science and Mathematics Division, Oak Ridge
National Laboratory, Oak Ridge, TN, 37831-6355, {wuqn,raons}@ornl.gov

Abstract: *An increasing number of distributed sensor networks are being deployed in a wide spectrum of scenarios for both military and civilian applications. The design of such systems requires the integration of many methods for solving various problems ranging from sensor deployment, through data communication, information fusion to environmental inference and hypothesis testing. This paper is focused on the network aspects in support of data communication services. We present some recent developments in networking techniques for multiple sensor systems in three topics: efficient mobile agent routing in mobile agent-based distributed sensor networks, bandwidth aggregation using multiple paths in wide-area wireline networks, and connectivity enhancement using node movements in small-area wireless mobile ad-hoc networks.*

Keywords: *Distributed sensor networks, data routing, mobile agent, bandwidth measurement, network connectivity*

1. INTRODUCTION

Sensor networks have become an important research area and attracted a great deal of attention during the past decade. The increasing popularity of sensor networks is not only driven by their practical applications in a broad variety of fields, but the many conceptual and methodological challenges they have posed.

A distributed sensor network (DSN) is composed of intelligent sensors that are geographically dispersed in the region of interest and interconnected via a communication network. The (acoustic, seismic, and infrared) data collected by leaf sensor nodes are transmitted over the network and integrated at the processing element to derive appropriate inferences about the environment for different purposes, such as target tracking, location, and surveillance. Since a DSN is typically deployed for remote operations in unstructured areas and the measurements are usually collected in a harsh, unreliable, and even adversarial environment, it is critical to provide extended networking capability to guarantee the performance of the entire system. This paper addresses three specific topics related to the networking aspect of distributed sensor networks.

Instead of sending all sensor data to the processing element, which performs a one-time data fusion as in a conventional server/client system, the mobile agent-based distributed sensor network (MADSN) proposed in [1] enables the computation to be spread out onto the participating leaf nodes with the intention of decreasing the consumption of scarce network resources (mostly the bandwidth) and the risk of being spied with hostile intent. In such a network scheme, a mobile agent carrying the executable instructions of data integration is dispatched from the processing element and selectively visits the leaf sensors along a certain path to fuse the data incrementally on a sequential basis.

The path quality of a mobile agent has a significant impact on the overall performance of MADSN implementation because the communication cost and the detection accuracy depend on the order and the number of nodes to be visited. We formulate the mobile agent routing problem with certain constraints and construct an appropriate objective function that reflects the routing requirements. The NP-completeness of this problem, whose detailed proof can be found in [2], rules out any polynomial solutions. We propose an approximate solution based on a two-level genetic algorithm (GA) and compare the simulation results with those computed by two other heuristics, namely Local Closest First (LCF) and Global Closest First (GCF).

The underlying networks supporting DSNs range from wide-area wireline networks, for example, deployed on a set of radar sites located across the country that track aircrafts, to small-area wireless networks, for example, deployed on a small robot team that measures radiation levels in an unstructured urban area suspected of nuclear or chemical contamination [3]. The current deployment of DSNs of various interconnection patterns utilizes the present Internet technologies mostly due to their wide availability. Consequently, the resultant networks often do not meet the DSN requirements since the Internet technologies are based on best effort services with very limited control at the network core [4].

For data transmission in DSNs built over wireline wide-area networks, the routing is solely determined by the intermediate routers and therefore the end nodes cannot reroute packets to avoid the congested network segments or accumulate bandwidths via multiple paths. To achieve better network performances, we propose a framework of network daemon, which performs bandwidth measurement, multiple quickest paths computing, and data routing.

The wireless connection is usually the only feasible way of communication among moving computing devices, sensors or robots, especially when the environment is harsh. Depending upon the implementation environments and application purposes, the wireless DSNs can operate under either access point or ad hoc modes. Access point mode, which is typically used in some structured areas with certain networking infrastructures, covers all node movements so that the node connections through access points essentially form a conventional Local Area Network (LAN) [5]. However, an ad hoc wireless network does not provide any specially designed routing hardware and software, and its link connectivity through wireless radio is highly dynamic and unpredictable due to the unstructured nature of the terrain and the distant movements of network nodes. The communication among moving nodes via conventional Transmission Control Protocol (TCP) byte-streams may experience difficulties that are not involved in wired networks because TCP needs the support of underlying routers and requires a direct or indirect connection between the source and destination nodes exist during the entire period of transmission. To meet these challenges, we design a novel transmission control and routing mechanism based on Connectivity-Through-Time (CTT) algorithm.

The rest of the paper is organized as follows. In Section 2, we describe the mobile agent routing problem and present a solution based a two-level genetic algorithm. In Section 3, we discuss the design of network daemon for DSNs built on wireline wide-area networks to achieve aggregate bandwidth via multiple paths. In Section 4, we present the CTT algorithm for DSNs built on mobile wireless small-area networks to enhance data transmission by exploring the node movements. A summary of our work is given in Section 5.

2. MOBILE AGENT ROUTING IN MADSN

A DSN typically consists of three basic components: processing elements, sensor nodes, and communication links. A group of neighboring sensor nodes that are commanded by the same processing element forms a cluster, which exchanges information with other clusters in the DSN to achieve higher-level data integration.

A MADSN with a simple network configuration is shown in Fig. 1 just for illustrative purpose.

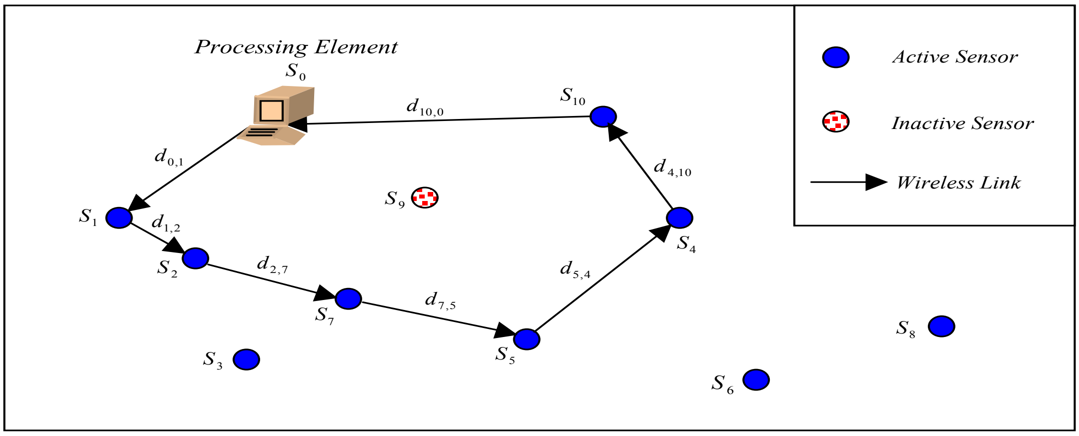


Fig.1 - An illustration of MADSN with simple configuration.

This sensor network contains one processing element, labeled as S_0, and N=10 leaf sensor nodes, labeled as $S_i, i = 1,2,..., N$, one of which is down or in sleep state. The physical distances of wireless links are represented by $d_{i,j}, i \neq j, i = 0,1,...,N, j = 0,1,..,N$.

The processing element dispatches a mobile agent that visits a subset of sensors within the cluster to fuse data collected in the coverage area. Generally speaking, the more sensors are visited, the higher detection accuracy will be achieved using any reasonable data fusion algorithm [6]. However, visiting more sensors often incurs more communication and computing cost in terms of energy consumption and path loss. The routing objective is to find a path for a mobile agent that satisfies the desired detection accuracy while minimizing the energy consumption and path loss.

We define an objective function of path *P* that considers the tradeoff between energy consumption *EC(P)*, path loss *PL(P)*, and detected signal energy *SE(P)*

$$O(P) = SE(P)\left(\frac{1}{EC(P)} + \frac{1}{PL(P)}\right) \qquad (1)$$

The detailed formulas for calculating the above quantities are derived in [2]. We further construct a fitness function based on *O(P)* to facilitate the solution using genetic algorithm

$$f(P) = O(P) + g \qquad (2)$$

where *g* denotes the punishment applied for dissatisfying the detection accuracy constraint and is define by

$$g = \begin{cases} 0, & SE(P) \geq E \\ \delta \cdot (SE(P) - E)/E & SE(P) < E \end{cases} \qquad (3)$$

where *E* is the desired detection accuracy or signal energy level and δ is a properly selected penalty coefficient.

By designing a two-level encoding scheme, we are able to adapt the generic string-based genetic algorithm to the mobile agent routing problem in MADSN. The first level is a numerical encoding of the sensor label ID sequence *L* in the order of sensor nodes being visited by a mobile agent, and the second level is a binary encoding of the visit status sequence *V* in the same visiting order. Therefore, masking the first level of numerical string *L* with the second level of binary string *V* yields a candidate path *P* for a mobile agent.

This two-level genetic algorithm applies some common genetic operators such as crossover, mutation, inversion, translocation as well as a proportional selection procedure. These operators are modified from those used in the conventional GA solution to Traveling Salesman Problem in order to suit the current context of two-level string encoding.

We compare the search results computed by GA, LCF and GCF to demonstrate the effectiveness of our algorithm. A series of sensor networks with random distribution patterns and node sizes ranging from 200 to 1600 are created for test. We manually choose an appropriate desired level of the detected signal energy for each network as well as the number of potential targets, which are randomly placed in the region. The performance comparisons of GA, LCF and GCF are illustrated in Fig. 2, Fig. 3, Fig. 4, and Fig. 5. Note that the quantities of path losses and energy consumptions are "normalized" to reasonable ranges before we plot them in the figures. Note that the objective value only serves as an indicator of the path quality according to our objective function, which does not bear a regular unit.

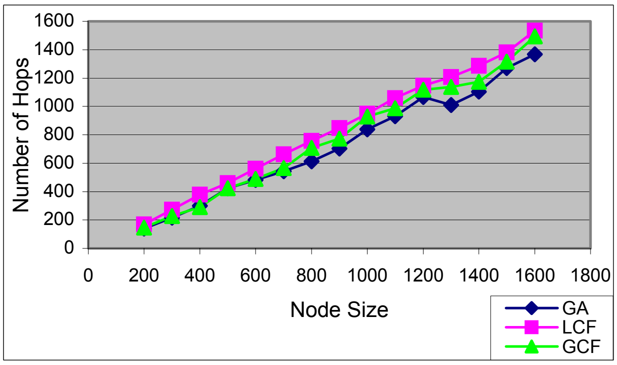


Fig.2 - Performance comparison: Node sizes vs. hop numbers.

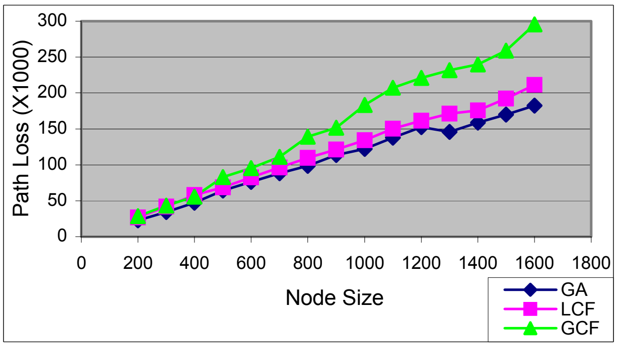


Fig.3 - Performance comparison: Node sizes vs. path losses.

From Fig. 2, Fig. 3, Fig. 4, and Fig. 5, we have observed that in most cases we studied, GA is able to find a satisfying path with smaller number of hops, lower energy consumptions, and less path losses than LCF and GCF algorithms. Fig. 5 clearly shows that the GA we implemented has a superior overall performance over two other heuristics in terms of the objective function we defined in Equation (1).

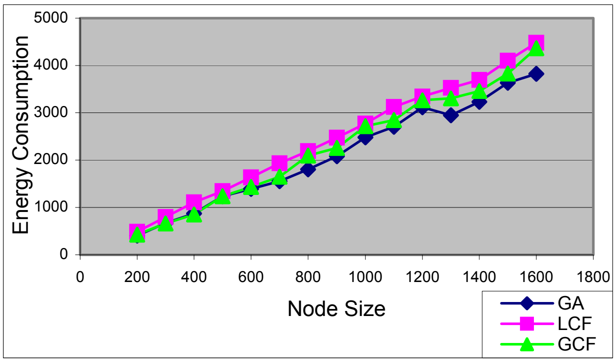


Fig.4 - Performance comparison: Node sizes vs. energy consumptions.

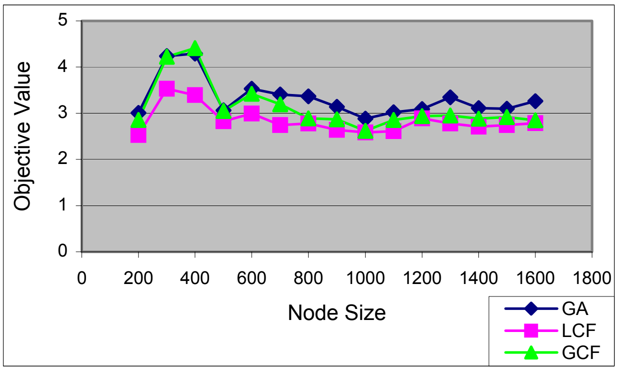


Fig.5 - Performance comparison: Node sizes vs. objective values.

More discussions on the algorithm comparisons such as computing complexity, real-time constraint, and selection of starting point can be found in [2].

3. NETWORK DAEMONS FOR BANDWIDTH AGGREGATION

The reliance on Internet technologies by DSNs often results in undesirable network performance due to the inherent limitations of the current Internet. We propose a framework of network daemons, which can be deployed at DSN nodes in wide-area networks, to enhance the functionality of DSNs. As shown in Fig. 6, a network daemon consists of four main components: link delay measurement, statistical regression estimate, multiple path computation, and data routing.

The end-to-end message transmission delay is collected using active measurement method. The source daemon creates a set of messages of various sizes and sends them to the destination daemon, which calculates the delay upon the completion of transmission. Such delay measurements between Oak Ridge National Laboratory (ORNL) and a number of other universities such as Louisiana State University (LSU), University of Oklahoma (OU), Old Dominion University (ODU), are plotted in Fig. 7. Each cluster represents the measured delays for transmitting messages with widely ranging sizes from ORNL to a single destination. As portrayed by these measurements, the "slope" corresponds to the available bandwidth at the time of experiment and the minimum delay can be estimated by the "intercept" (where the message size is approaching zero). The "noise" components in delay measurements are mostly caused by the random queuing delays incurred in the intermediate routers or end hosts. The distributions of these random components are very complicated due to the dynamically changing network status.

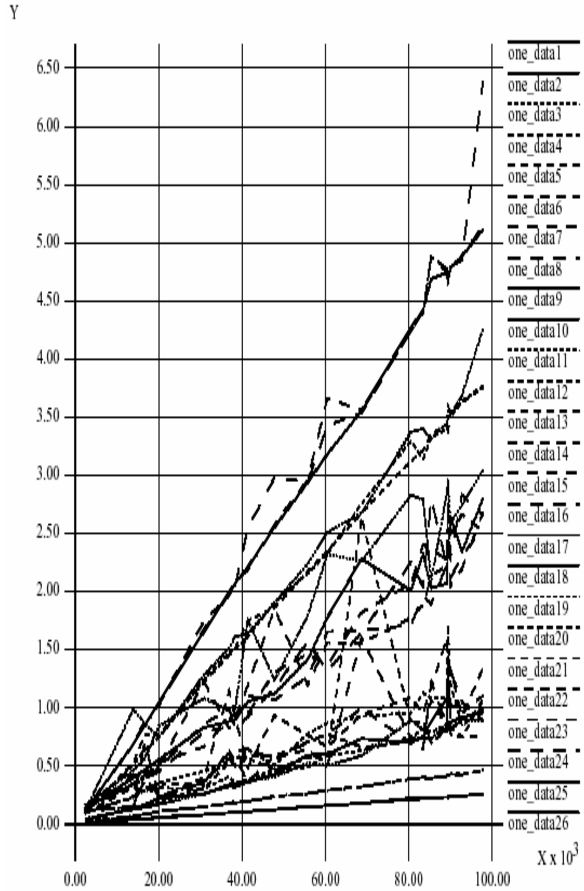


Fig.6 - Framework of network daemon.

Linear regression is performed on these delay measurements to estimate the available bandwidth and minimum delay between two daemon nodes. The link information is then broadcast to the other daemons in the rest of the network so that a complete topology of the overlay network of daemon nodes can be established. The path computation module computes multiple quickest paths for a given user data size based on the network topology and updates the routing table when either the network topology or the link characteristics change. The detailed description of the algorithm used for computing multiple quickest paths can be found in [7]. The data routing module conducts similar activities to the Internet routers such as sending/forwarding/receiving data packets except that it is implemented at the user level. Actually, the whole daemon framework resides at the application layer so that they can be installed on most Internet nodes without the need to modify the system kernel or network infrastructures.

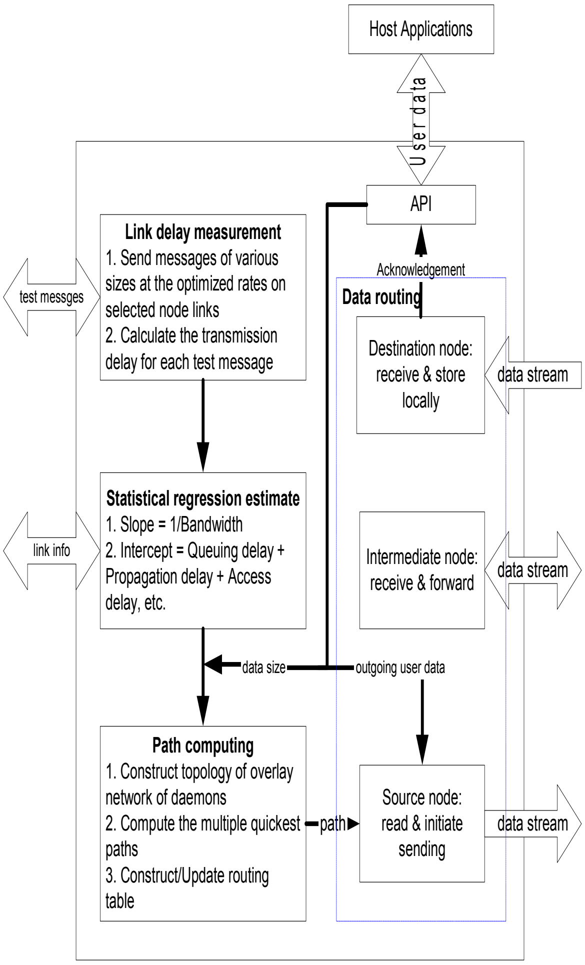

Host Applications
User data
API
Acknowledgement
Data routing
Link delay measurement
1. Send messages of various sizes at the optimized rates on selected node links
2. Calculate the transmission delay for each test message
test messges
Destination node: receive & store locally
data stream
Statistical regression estimate
1. Slope = 1/Bandwidth
2. Intercept = Queuing delay + Propagation delay + Access delay, etc.
link info
Intermediate node: receive & forward
data stream
data size
outgoing user data
Path computing
1. Construct topology of overlay network of daemons
2. Compute the multiple quickest paths
3. Construct/Update routing table
path
Source node: read & initiate sending
data stream

Fig.7 - Transmission delays (seconds) vs. message sizes (bytes) from ORNL to LSU, OU, and ODU.

We run network daemons at the above four sites. The overlay network topology is illustrated in Fig. 8. A set of messages with various sizes are sent from ORNL to OU using two different sending methods.

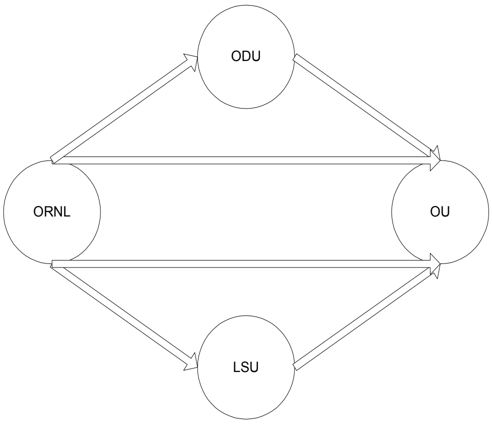


Fig.8 - Overlay network topology of 4 daemon nodes.

Typical experimental results are plotted in Fig. 9, where the upper curve corresponds to the message transmission using a single default TCP stream, while the lower curve corresponds to the message transmission using multiple paths computed by the daemon. As shown in Fig. 8, the multiple paths consist of two direct TCP streams and two indirect TCP streams via intermediate daemons. The application of multiple paths provides an aggregate bandwidth that reduces the end-to-end delay by about 35% in all the cases we studied.

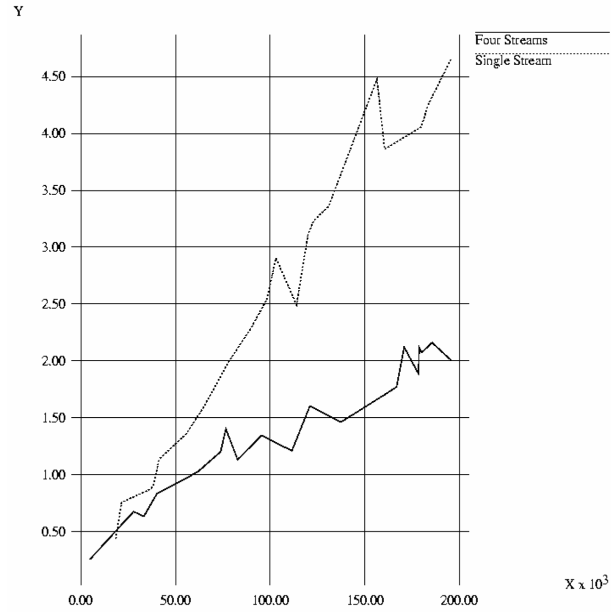


Fig.9 - Experimental results of data transmission using a default TCP stream and multiple paths.

4. CONNECTIVITY-THROUGH-TIME IN MOBILE WIRELESS NETWORKS

Wireless networks have found many successful applications ranging from campus access to sensor networks [8]. In some scenarios like the robot team exploring a potentially radioactive area, the moving nodes need to effectively communicate with other nodes in the network to coordinate their activities as well as to combine the gathered information. However, the networking needs for this class of applications are quite specific and are not adequately addressed by the existing wireless ad hoc networking technologies.

Wireless networks have very different operational characteristics from wired networks. Firstly, the packet losses in wireless networks are mostly due to physical link failures instead of network congestion. Secondly, the signal attenuation often causes the link to break down when environmental interferences increase or the node moves out of the maximum radio distance. Therefore, the network connectivity through wireless radio in ad hoc mobile networks can be highly dynamic, intermittent, and unpredictable. These wireless operational characteristics have posed significant challenges for data transmission based on conventional TCP flows because TCP is an end-to-end transport protocol that does not provide capabilities specifically accounting for connectivity constraints in wireless environments. In general, TCP needs routing support from the underlying routers and requires a continuous byte-stream connection between source and destination during the entire period of transmission.

We present a Connectivity-Through-Time (CTT) concept and design a CTT protocol that utilizes node movements to enhance data transmission in ad hoc mobile networks. A typical CTT example is illustrated in Fig. 10, where data is successfully delivered from a source node v_1 to a destination node v_3 even though they never "see" each other. The two-direction arrow represents a direct wireless link between two nodes located within the maximum wireless radio range.

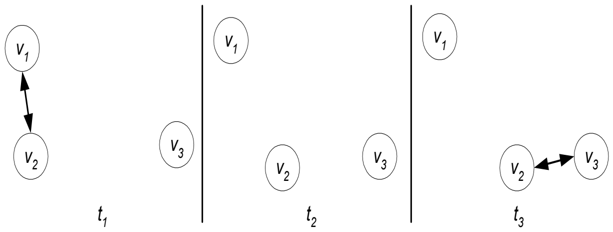


Fig.10 - A CTT example: transmission through intermediate node movements.

At time t_1, the source node v_1 knows from its neighbor list that the destination node v_3 is unreachable at the moment. There are two options for node v_1: it can either wait until node v_3 come into its radio coverage area or broadcast the data to whoever are reachable, i.e. node v_2 in this case. Suppose node v_1 broadcasts the data as well as its destination information, and node v_2 receives and buffers the data because it is only destined to node v_3. Afterwards, node v_2 carries the data and moves towards node v_3. At time t_2, node v_2 goes out of the radio ranges of both node v_1 and node v_3. Eventually, node v_2 enters the radio area covered by node v_3 at time t_3. Once this new link is detected, node v_2 checks for the destination availability for all temporary data in its buffer. In this case it happens that node v_3 is the destination of the buffered data, and hence node v_2 retrieves the data from its data repository and transmits it to the destination node v_3.

The CTT protocol is implemented based on User Datagram Protocol (UDP). The framework of CTT function modules is given in Fig. 11.

The connectivity computation module is a relatively independent unit, which exchanges the neighbor information and builds a routing table. A special datagram named I-am-here with the neighbor list is broadcast to all neighbor nodes, and the receipt of an I-am-here datagram indicates that there exists a wireless connection between the sender and receiver. The up-to-date path information is obtained from the routing table and provided to the transport control module.

There are two main components in the transport control module: datagram receiving and sending units. The datagram receiving unit accepts UDP datagrams either from adjacent nodes or local host. If the arrived message is interpreted as a "send" command issued by the local host, the designated data source will be read directly from local storage devices and packed in fixed-size chunks, each of which has a user-defined header (destination information) attached. Otherwise, if the received datagrams are neither originated from nor destined to the local host, they will be simply placed in the datagram table of a corresponding file buffer.

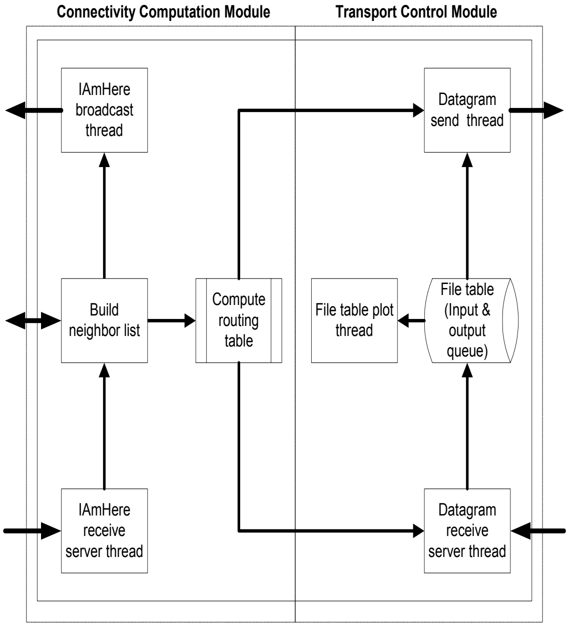


Fig.11 - Framework of CTT function modules.

The datagram sending unit repeatedly scans the whole file list on a sequential basis and assigns a sending priority level for each datagram. If datagrams destined to the local host are received in order, they are delivered to host applications immediately; otherwise they are held until the holes are filled. Any passing-by datagrams are loaded into the outgoing queue for forwarding or broadcasting.

The implementation details including control flow and data structures of CTT protocol can be found in [9]. Here we only show the experimental result in Fig. 12 and Fig. 13 for the scenario illustrated in Fig. 10.

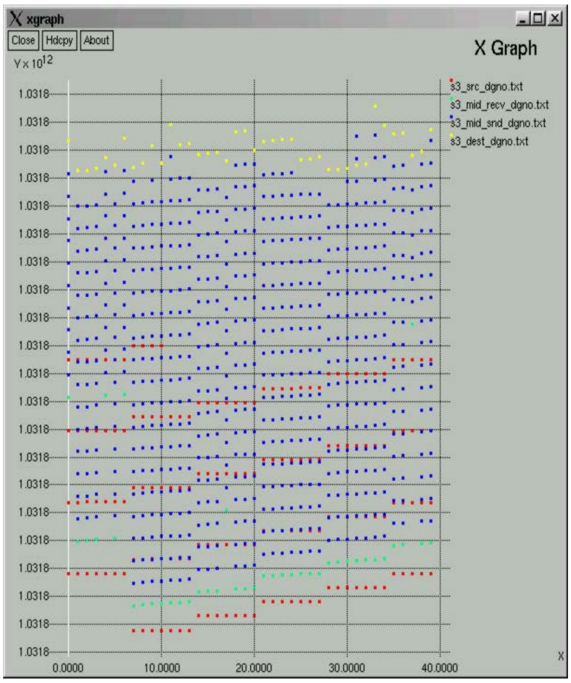


Fig.12 - CTT: datagram number vs. sending/receiving time.

As portrayed in Fig. 10, this scenario has three stages. In the first stage, the source node is only able to communicate with the intermediate node so that the datagrams are broadcast after a certain time-out expires. This connection breaks in the second stage where the intermediate node is the only active node performing broadcast. In the third stage, the link between the intermediate node and the destination node comes up and the datagrams are delivered right after the new path is computed.

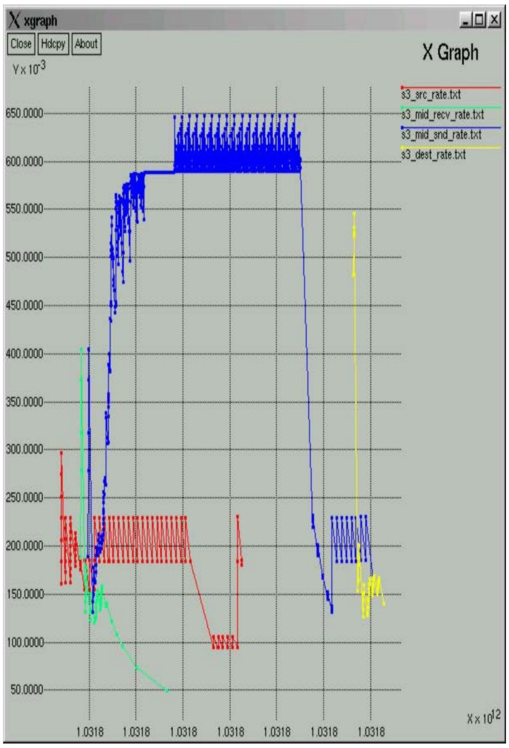


Fig.13 - CTT: time vs. average sending/receiving rate.

Simulation results in more various scenarios can be found in [9].

5. CONCLUSION

In this paper, we presented some recent developments of networking techniques in distributed sensor networks. Firstly, we formulated a route computation problem for the mobile agent in MADSN in terms of maximizing the received signal strength while keeping path loss and energy consumption low, and proposed a genetic algorithm to solve this problem by employing a two-level genetic encoding and suitable genetic operators. Secondly, we designed a framework of network daemons to perform bandwidth measurement and achieve bandwidth aggregation via multiple quickest paths in DSNs deployed over wireline wide-area networks. Finally, we described a CTT concept and designed a CTT protocol specifically accounting for connectivity changes due to node movements to enhance the networking capability in ad hoc mobile wireless networks.

6. REFERENCES

[1] H. Qi, S. S. Iyengar. K. Chakrabarty. Multi-resolution data integration using mobile agents in distributed sensor networks. *IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews*, vol. 31, no. 3, pp383-391, August, 2001.

[2] Q. Wu, S.S. Iyengar, N. S.V. Rao, J. Barhen, V.K. Vaishnavi, H. Qi, K. Chakrabarty. *On Computing the Route of a Mobile Agent for Data Fusion in a Distributed Sensor Network*, submitted to IEEE Transactions on Knowledge and Data Engineering, 2003.

[3] R.R. Brooks, S.S. Iyengar. *Multi-Sensor Fusion: Fundamental and Applications with Software*. Prentice-Hall, 1998.

[4] L.L. Peterson, B.S. Davie. *Computer Networks*. Morgan Kaufman Publication, Second Edition, 2000.

[5] W. Stallings. *Wireless Communications and Networks*. Prentice-Hall, 2001.

[6] N. S. V. Rao. *Multisensor Fusion Under Unknown Distributions: Finite Sample Performance Guarantees, in Multisensor Fusion*, A.K. Hyder, E. Shahbazian, E. Waltz (editors), 2002.

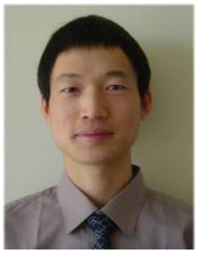
[7] Q. Wu. Control of Transport Dynamics in Overlay Networks. P*h.D. dissertation*, Louisiana State University, 2003.

[8] C. Perkins. *Ad Hoc Networking*, Addison-Wesley, 2001.

[9] N.S.V. Rao, Q. Wu, S.S. Iyengar, A. Manickam. Connectivity-Through-Time Protocols for

Dynamic Wireless Networks to Support Mobile Robot Teams. *2003 IEEE International Conference on Robotics and Automation*, Taiwan, 2003.

S. Sitharama Iyengar *received his M.S. from the Indian Institute of Science in 1970 and his Ph.D. degree at Mississippi State University in 1974. He is the Chairman and Roy Paul Daniels Chaired Professor in the Computer Science Department at Louisiana State University and is also Satish Dhawan Chaired Professor at the Indian Institute of Science. His research interests include high-performance algorithms, data structures, sensor fusion, data mining, and intelligent systems. His publications include 13 books (authored or coauthored textbooks; Prentice-Hall, CRC Press, IEEE Computer Society Press, John Wiley & Sons, etc.) and over 280 research papers in refereed journals and conferences in the areas of high-performance parallel and distributed algorithms and data structures for image processing and pattern recognition, and distributed data mining algorithms for biological databases. He has served as an associate editor for the Institute of Electrical and Electronics Engineers and as guest editor for several IEEE transactions and other journals. He has served on review panel committees for NSF, NASA, DOE-ORNL, the U.S. Army Research Office, etc. He has been on the prestigious National Institute of Health-NLM Review Committee, in the area of Medical Informatics for 4 years. He is a series editor for Neuro-Computing of Complex Systems for CRC Press. He is also a Fellow of Association of Computing Machinery (ACM), a Fellow of the IEEE, a Fellow of American Association of Advancement of Science (AAAS), a Williams Evans Fellow, IEEE Distinguished Visitor, a member of the European Academy of Sciences and the New York Academy of Sciences. He has been a Distinguished Lecturer for the IEEE Computer Society for the years 1995 – 1998, an ACM National Lecturer (1986–1992), and Member of the Distinguished SIAM Lecturer Program (2000–2002). He was a member of the ACM accreditation committee for 2000 – 2002.*



Qishi Wu *received his B.S. degree in remote sensing and GIS from Zhejiang University, People's Republic of China in 1995, his M.S. degree in geomatics from Purdue University in 2000, and his Ph.D. degree in computer science from Louisiana State University in 2003. He is currently a post-doc research fellow in the Computer Science and Mathematics Division at Oak Ridge National Laboratory. His research interests include computer networks, distributed sensor networks, algorithms, and artificial intelligence.*

Nageswara S. V. Rao *received B. Tech in Electronics and Communications Engineering from Regional Engineering College, Warangal, India in 1982, M. E. from School of Automation, Indian Institute of Science, Bangalore in 1984 and PhD in Computer Science from Louisiana State University in 1988. He is currently a Distinguished Research Staff Member at Oak Ridge National Laboratory, where he joined in 1993. He was an assistant professor of computer science at Old Dominion University during 1988-1993.His research interests include network transport dynamics, statistical approaches to transport control, information and sensor fusion, robot navigation and fault diagnosis.*