Exploring TCP/IP Performance for Wireless Networks with multi-agent system

B. Sasikumar[1], and V. Vasudevan[2]
1. Professor, Department of CSE, RMD Engineering College, Kavaraipettai
2. Professor, Department of IT, A.K. College of Engineering

ABSTRACT

Rapid advances in consumer electronics have led to use off-the-shelf gaming hardware and software that provide better interactive graphics. In recent times the uses of wireless devices are fetching ubiquitous and these networks have turn into an increasingly important part of global communication architecture. Information gathering is the problem of sending one packet each from a subset of the nodes to a single sink node in the network. In view of the fact that the vast majority of data applications make use of the TCP/IP protocols, optimizing TCP performance over these networks would have a broad and significant impact on the user-perceived data application performance.

Due to high data rate in wireless networks the TCP performance has a broad and significant impact on data applications. In this paper we explore the performance of TCP/IP with a RETSINA, UTSAF, and JADE multi-agent system. Performance has been analyzed for the TCP/IP with and without multi-agent system for the metrics throughput and latency.

Keywords: *latency, performance, RETSINA agent, UTSAF agent, JADE agent, TCP, throughput, wireless networks.*

I. INTRODUCTION

A wireless network consists of mobile or stationary nodes which can communicate with each other over the wireless links without the aid of any established infrastructure or centralized administration. These types of networks are useful in any situation where temporary network connection is needed, in case of disaster relief or in battlefield. Probabilistic broadcasting simultaneously addresses two problems endemic to wireless networks, which are power limitation and collisions [26].

TCP performance over wireless networks has been studied over the last several years. Early research [4], [6] showed that wireless link losses have dramatic adverse impact on TCP performance due to the difficulty in distinguishing congestion losses from wireless link losses. These networks concentrated on improving the MAC layer protocols and routing protocols [20].

Ad-hoc wireless networks are multihop wireless networks which consist of finite number of radio equipped nodes that are autonomous. In this each node has a transmission radius and is capable to transmit message to all its neighbors that are within the transmission range [15].

Flooding applications include paging a particular host or sending an alarm signal, which is also used for route discovery in a source initiated on-demand routing [7]. In this paper we concentrated our attention to the transport layer routing which uses the Transmission Control Protocol (TCP). Transport connections in wireless networks are plagued by problems such as high bit rates, frequent route change and partitions.

While running TCP over such networks, the throughput of the connection is viewed to be poor because this protocol takes the lost or delayed acknowledgements as congestion [11]. The reason for poor performance in Transmission Control Protocol has been examined in [21] and that need more effective mitigation strategies.

The rest of the paper is organized as follows. Section 2 presents the performance of TCP/IP. Implementation of multi-agent system is given in section 3. In section 4 details of experimental setup is given and section 5 shows the simulation results. Finally conclusion is given in section 6.

II. TCP/IP

Transmission Control Protocol is a reliable connection-oriented transport layer protocol that allows a byte stream originating on one node to be delivered without error on any other node in the network. TCP reacts to any packet losses by reducing its congestion window size before retransmitting lost packets, initiating congestion control or avoidance mechanisms and backing off its retransmission timer [9].

Manuscript received September 5, 2008.
Manuscript revised September 20, 2008.

When packets are lost in networks, due to congestion and which results in an unnecessary reduction of end-to-end throughput and its related performance [27]. Increased delay variations which results in delay can cause timeouts in TCP, where the source incorrectly assumes that a packet is lost while the packet is only delayed, which forces TCP to slow start that reduce the TCP performance.

Figure 1 shows that at different timings node N1 has an open TCP connection to node N2. But the network gets portioned causing nodes N1 and node N2 to be in different portions. The change in node connectivity has incredible consequences for TCP's throughput that will drop at very low levels.

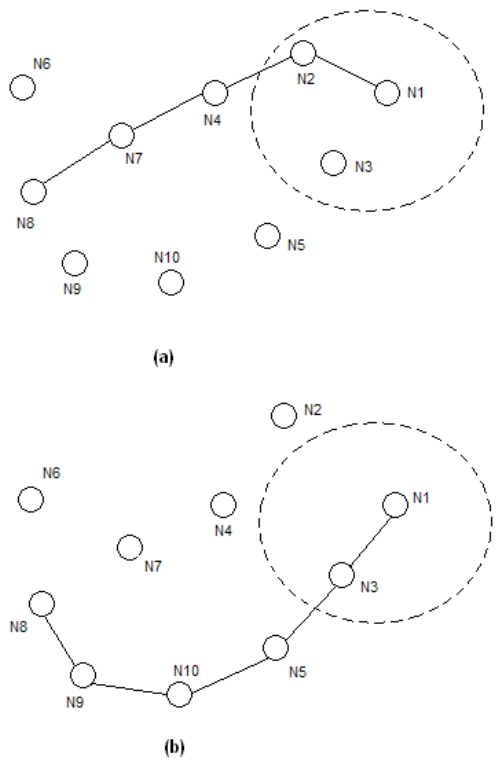

Figure 1 Partitions formed due to mobility of node

In this paper we explore new multi-agents named RETSINA and UTSAF to the TCP which improves the performance of TCP/IP. These agents reduces the problems at the congestion window and when the agent executes tasks, they organize themselves to avoid processing bottlenecks and form teams of deal with dynamic changes in information, tasks, number of agents and their capabilities.

III. MULTI-AGENT SYSTEM

A single intelligent agent is created to solve only a particular, small problem due to its limited knowledge, its computing resources, and its perspective. Multi-agent systems were introduced to offer modularity as a tool to handling complexity and heterogeneity.

A. RETSINA

RETSINA (Reusable Environment for Task-Structured Intelligent Networked Agents) is a multi-agent system that provides a domain independent componentized and reusable substratum to allow heterogeneous agents to co-ordinate in a variety of ways and to enable a single agent to be part of a multi-agent infrastructure.

Each single agent is provided with an internal planning component called the RETSINA planner and the internal planner formulates detailed plans and executes them to achieve local and global goals. Figure 2 shows an example of multi-agent RETSINA system in which each agent communicates with each other to perform a specific task [20].

Each RETSINA agent consists of several modules, each of which is implemented as multithreaded code. The functional modules are a communicator, a planner, a scheduler and an execution monitor. The modules can operate asynchronously and concurrently [14]. Communicator supports peer-to-peer communication between agents and receives requests from users or other agents in KQML format and the massages are transmitted through TCP/IP sockets.

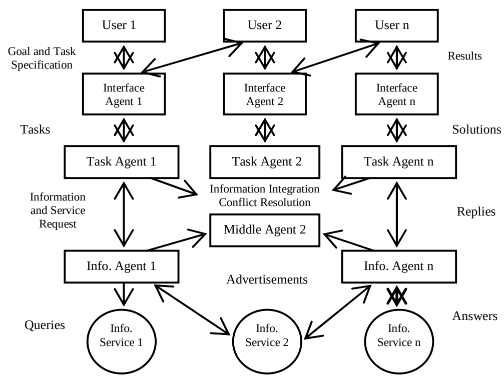

Figure 2 Multi-agent RETSINA System

The planner module receives goals through communication messages and solves those goals and the planning component is reusable and capable of accepting different planning algorithms [2]. The scheduler has a default scheduling algorithm that uses the earliest deadline first heuristic. The execution monitor executes the actions and monitors the executions. These modules are implemented as autonomous threads of control to allow concurrent planning and actions scheduling and execution.

This system consists of three agents named interface agent, task agent and information agent. Interface agent interacts with users receiving their specifications and delivering results. Task agent formulate the plans and carry out them, which have knowledge of the task domain and which other task agents or information agents are relevant to perform various parts of the task. Information agents provide intelligent access to a heterogeneous collection of information sources [17].

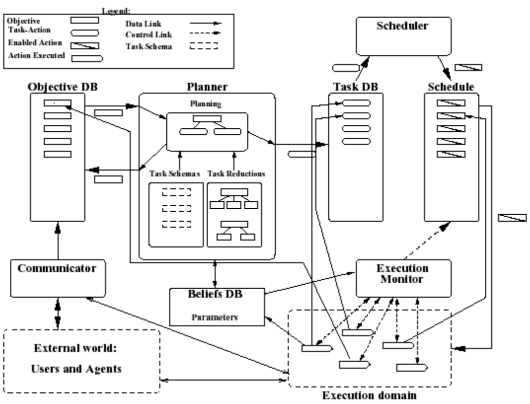

Figure 3 RETSINA Planning Architecture

Figure 3 shows the overall architecture of the agent [19]. The objective DB dynamically stores the objective with higher priority on the queue will be handled first by the planner. The task DB has data store unit that holds the plan. Tasks are added by the planner and removed by the scheduler. The schema library has a static data store unit that holds tasks schemas which are used by the planner for task instantiation. The task reduction library holds reductions of tasks and is used by the planner for task decomposition. The belief DB maintains the agent's knowledge of the domain in which the plan will be executed.

B. UTSAF

UTSAF (Unreal Tournament Semi-Automated Force) is bridging software written to take benefit of the power of gaming systems by allowing them to participate in distributed simulations with military simulators such as OneSAF which use the DIS protocol Distributed simulation has long been used in military applications for simulated battlefield training and strategic planning, which involves different operators, agencies and partners to get better the interoperability [10].

The proposed multi agent uses a multi domain, Multi Agent System (MAS) in which each agent performs a specific task and interacts with other agents through a standard infrastructure to achieve interoperability between incompatible systems. MAS allow processing to be distributed among a group of agents with preexisting infrastructure for communication and coordination [22]. Agents act as wrappers around the incompatible systems and use this communication and coordination infrastructure to grant and abstraction layer for exchanging data and messages [23].

UTSAF uses a modification route by the GameBots [1] to provide the simulation interface, which has two types of entities named human players who run individual copies of the game and connect to the server and bots, simulated players running simple reactive programs. This architecture uses a common communication infrastructure and language, agents can freely exchange information with one another and is extensible and able to support novel systems mediated by new agents. MAS offers modularity as a tool to handling complexity and heterogeneity [25].

Figure 4 shows the UTSAF architecture a multi agent environment is used to span the gap between the two simulations. To keep the system scalable, diverse and extensible architecture allows subtasks to be assigned to specialized agents. A communication protocol between agents is essential to allow tasks and data to be distributed.

To acquire information from the traffic, a Protocol Data Unit (PDU) parser extracts relevant information like entity type, location, velocity, and orientation from each DIS Protocol Data Unit and sends this information to a SAF manager agent. SAF manager agents control the flow of information between OTBSAF and UT at the entity level. These agents receive information from an SAF broker, update an internal database, and forward relevant updates to an agent representing the entity on the UT side of the simulation. GameBot agent serves as the final connection between the agent space and the UT representation of the entities.

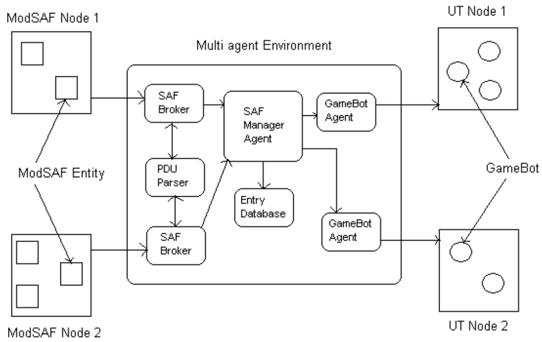

Figure 4 Multi-agent UTSAF System

C. JADE

The JADE (Java Agent Development Environment, [12]), that allow to develop FIPA (Foundation for Physical Intelligent Agents, [8]) compliant multi-agent systems in wireless devices, is essential for the social acceptance of technology. Each JADE system has a connection endpoint. A system can handle messages through any other connection. Connection binding provides an override of the default behavior so that the single user can handle all activity for a specific connection. The TCP/IP listen connection is automatically created and opened when the first user of the transport group invokes the begin listening method.

A user joins a transport group by creating a jade multiuser TCP transport instance, and when the last user leaves the group all connections are closed and the transport group is deleted. For multicast transport group it uses a separate instance of jade multicast user TCP transport instance for each group that joins. The client connection is idle, queued or assigned.

A connection is idle when it has no queued events such as data ready for reading. A connection is queued when it has one or more events ready for a worker to process. A connection is assigned when the user handles an event for the connection. Various exceptions can be raised when transport or connection properties are updated or methods are called.

All unhandled exceptions that cut back through a connection event method cause the connection to be closed. A TCP/IP connection treats the data being sent or received as a continuous stream of bytes. The application sending or receiving the data is responsible for deciding where one message stops and the next one begins.

In agent based system functionality is divided into agents, and these agents coordinate their actions and communicate by exchanging messages. If large amount of agents are used, large number of messages are to be expected. Here the message load efficiency of JADE agent is find out.

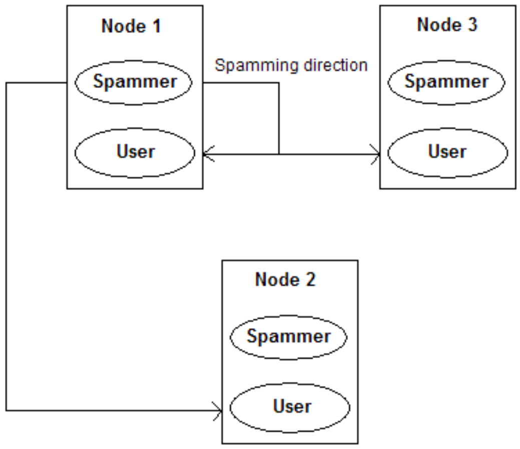

Figure 5 Spamming scheme

The general scheme of interaction between three spammer agents and three user agents is shown in the figure 5. This is designated to flood the system with Agent Communication Language messages. In this spammer agents send to user agents a large number of messages.

All spammer agents start sending messages to all user agents. In the JADE platform all posted messages are put in a receiver message queue and then it is processed by the receiver. During the execution each spammer agent broadcasts a certain number of messages and the total time of this broadcast is measured. A total of 50000 messages were sent by each spamming agent to each user agent.

IV. EXPERIMENTAL SETUP

The simulation environment is created in NS-2, a network simulator that provides support for simulating multihop wireless networks. NS 2 was written using C^{++} language and it uses the object oriented tool command language. The simulations were carried out using a MANET environment consisting of 50 wireless mobile nodes roaming over a simulation area of 1000 meters x 1000 meters flat space operating for 900 seconds of simulation time.

Each node had a radio propagation range of 250m and channel capacity was 2Mb/s. We did our experiments with movement patterns for seven different pause times of

0,100, 200, 300, 400, 500, and 600 seconds. The PDU packets are generated at the rate of 100ms/packet and 20 ms/packet, respectively. The packet rate in the test bed is above 1000 packets/sec or more. The radio and IEEE 802.11 MAC layer models were used and the nodes move according to Random Waypoint mobility model.

Each simulation scenario uses a total of 16 TCP connections, each is established for one of the configurations source – sink pairs. Every source generates an infinite stream of data bytes. The parameters used in the simulation include the TCP packet size, the TCP window size and the switch buffer size.

Each mobile node in the network starts its journey from a random location to a random destination with a randomly chosen speed. Once the destination is reached, another random destination is targeted after a pause by the mobile node. Once the node reaches the boundary area mentioned in the network, it chooses a period of time to remain stationary.

After the end of this pause time, the node chooses a new direction, this time between 0 an 180 degrees, adjusted relative to the wall of the area on which the node is located. By varying the pause time, the relative speeds of the mobiles are affected.

V. SIMULATION RESULTS

Performance of TCP/IP has been analyzed for the metrics throughput and latency. Unless otherwise noted, each one of the graphs presented in this section portrays one of these performance metrics on the y-axis as a function of speed and time on the x-axis respectively.

Throughput

Throughput is the amount of data that is delivery from the source node to the destination node in a network. Figure 6 shows the performance analysis of throughput for the TCP/IP with and without multi-agent system.

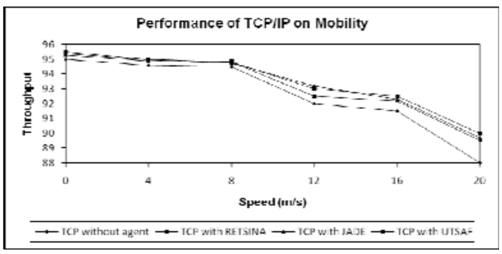

Figure 6 Analysis of Throughput

Latency

Latency or response time is the time between the transmission of data packets from the source node and the time of its reception by a receiver node. Figure 7 shows the performance analysis of latency for the TCP/IP with and without multi-agent system.

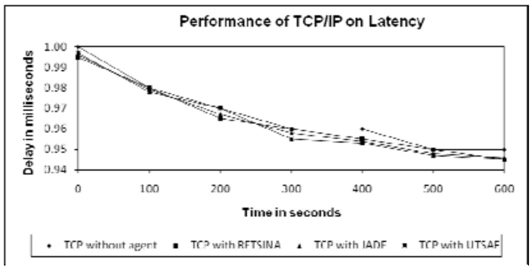

Figure 7 Analysis of Latency

VI. CONCLUSION

In this paper, we explore the multi-agent system with the TCP to improve the performance of TCP/IP in the wireless networks. The end-to-end throughput of TCP/IP connection has increased while using the multi-agent system. We performed the simulation by varying the speed for various scenarios. So we conclude that by using these agents with the TCP, the total performance of TCP/IP is increased. This approach can easily be extended to integrate other components such as human behavior models, intelligent information gathering and filtering, and decision-making modules. Using agents to wrap incompatible systems lacking other provisions for interoperability allows the rapid construction of middleware.

REFERENCES

[1] R. Adobbati, A. N. Marshall, A. Scholer, S. Tejada, G. Kaminka, S. Schaffer, and C. Sollitto, "GameBots: A 3D virtual world test-bed for multi-agent research" Proceedings of the second international workshop on Infrastructure for Agents, MAS, and Scalable MAS, May – June 2000, Montreal. Canada.

[2] Anand S. Rao, and Michael P. Georgeff, "Modeling rational agents within a bi-architecture", in proceedings of the second International Conference on Principles of knowledge Representation and Reasoning, Cambridge, MA, 1991.

[3] Azer Bestavros, and Gitac Kim, "Implementation and Performance Evaluation of TCP Boston A fragmentation-tolerant TCP Protocol for ATM Networks", Boston University, pp. 1049-1053, IEEE 1997.

[4] A. Bakre and B. R. Badrinath, “Handoff and System Support for Indirect TCP/IP”, proceedings of second Usenix Symposium on Mobile and Location-Independent Computing, Apr. 1995.

[5] A. Bakre, and B. R. Badrinath, “I-TCP : Indirect TCP for Mobile hosts”, in proceedings of 15th International Conference Distributed Computing Systems, Vancouver, BC, Canada, pp. 136-143, June 1995.

[6] H. Balakrishnan, S. Seshan, E. Amir, and R. H. Katz, “Improving TCP/IP Performance over Wireless Networks”, proceedings of ACM Mobicom, Nov. 1995.

[7] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, “Routing with Guaranteed Delivery in ad-hoc wireless Networks”, proceedings of third International Workshop Discrete Algorithms and methods for Mobile Computing and Communication (DIALM)< pp. 48-55, Aug. 1999.

[8] Foundation for Intelligent Physical Agents: http://www.fipa.org.

[9] S. Floyd, “TCP and Explicit Congestion Notification”, http://www.nrg.ee.lbl.gov/floyd/ecn.htm.

[10] Institute of Electrical and Electronics Engineers, 1993, IEEE P1278, International standard ANSI/IEEE standard 1278-1993. In Standard for information technology: Protocols for distributed interactive simulation, Piscataway, NJ: IEEE.

[11] W. D. Ivancic, D. Brooks, B. Frantz, D. Hoder, D. Shell, D. Beering, “NASA’s Broadband Satellite Networking Research, IEEE Communications Magazine, pp.40-47, July 1999.

[12] JADE–LEAP:http://sharon.cslet.it/ project/ jade.

[13] Jain Liu, and Suresh Singh, “ATCP : TCP for Mobile Ad-hoc Networks”, IEEE journal on selected areas in Communications, Vol. 19, No. 7, pp 1300-1315, July 2001.

[14] Jeremy L. Mineweaser, J. Scott Stadler, Susan Tsao and Maurice Glanagan, “Improving TCP/IP performance for the Land Mobile Satellite Channel”, MIT Lincon Laboratory, Lexington, pp. 711-718, IEEE 2001.

[15] D. B. Johnson, and D. A. Maltz, “Dynamic source routing in ad-hoc wireless networks”, in Mobile Computing, T. Imielinski and H. F. Korth, Eds., Norwell, M. A Kluwer, pp. 153-191, 1996.

[16] Jorg P. Muller, “The Design of Intelligent Agents”, Springer, 1996.

[17] Katia Sycara, Keith Decker, Anadeep Pannu, Mike Williamson, and Dajun Zeng, “Distributed intelligent agent”, IEEE Expert, Intelligent Systems and their Applications 11(6), pp. 36-45, 1996.

[18] F. Khafizov, and M. Yavuz, “TCP over CDMA2000 networks”, Internet Draft, draft-khafizov-pilc-”, in ACM Computer Communications Review, Vol. 30, No. 1, Jan. 2000.

[19] M. Paolucci, D. Kalp, A. Pannu, O. Shehorg, and K. Sycara, “A planning Component for RETSINA Agents”, Internet Draft.

[20] V. D. Park, and M. S. Corson, “A highly adaptive distributed routing algorithm for mobile wireless networks”, in Proceedings of IEEE INFOCOM, Kobe, Japan, pp. 1405-1413, April 1997.

[21] C. Partridge, and T. J. Shepard, “TCP/IP Performance over Satellite Links”, IEEE Network Magazine, pp. 44-49, Sep.-Oct. 1997.

[22] Phongsak Prasithsangaree, Joseph Manojlovich, Stephen Hughes, and Mike Lewis, “UTSAF : A Multiagent based Software bridge for Interoperability between distributed military and commercial Gaming Simulation”, The society for modeling and simulation international, SIMULATION, Vol. 80, Issue 12, pp. 647-657, Dec. 2004.

[23] J. M. Sardella, and D. L. High, “ Integration of fielded army aviation simulators with ModSAF: The eighth army training solution, Proceedings of Interservice/Industry Training, Simulation and Education Conference, Nov. 27-30, Orlando, Finland.

[24] M. Seddigh, J. Solano Gonzalez, and I. Stojmenovic, “RNG and Internal Node Based Broadcasting Algorithms for Wireless One-to-One Networks”, ACM Mobile Computing and Communication Rev. Vol. 5, No. 2 to appear.

[25] K. P. Sycara, “Multi-agent systems” AI Magazine 10 (2), pp. 79-93, 1998.

[26] Sze-Yao Ni, Yu-Chie Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu, “The broadcast storm problem in a mobile ad-hoc network”, proceedings of the fifth annual ACM/IEEE, International Conference on Mobile Computing and Networking, pp. 151-162, Aug. 1999.

[27] Vinod Vulupala, and Vijay Kumar, “Dynamic Mobile IP and Nice-TCP for Improving TCP/IP Performance”, proceedings of the third International Conference on Mobile Data Management (MDM’02), IEEE Computer Society, 2002.

[28] Xiao Chen, and Jian Shen, “Reducing Connected Dominating Set Size with Multipoint Relays in Ad-hoc Wireless Networks”, Proceedings of the Seventh International Symposium on Parallel Architecture, Algorithms and Networks (ISPAN’04), IEEE Computer Society, 2004.