Secure Packet Transfer in a Network Using Hierarchical Scheduler with Crypto-Algorithm

C.Suganthi
UG student
Department Of Computer Science And Engineering
Alpha College Of Engineering And Technology
Puducherry-607402, India
e-mail: suganthi.c.cs@gmail.com

A. Sumiya
UG Student
Department Of Computer Science And Engineering
Alpha College Of Engineering And Technology
Puducherry-607402, India
e-mail: sumiyalohit@gmail.com

R. Sendhil
Assistant professor
Department Of Computer Science And Engineering
Alpha College Of Engineering And Technology
Puducherry-607402, India
e-mail: sendhildit6464@gmail.com

ABSTRACT

The real-time scheduling algorithms are in care of timing constraints; they don't pay any attention to enhance or optimize the real-time packet's security performance. In this work, the cryptographic algorithm with hierarchical scheduler is used for packet transfer from the source to destination. The proposed system which combines the functionality of real-time scheduling with the security service enhancement, the real-time scheduling unit uses the cryptographic algorithm to provide more secure for the data transmission. The QoS guarantees for different classes of real-time data flows text, video, audio, while adaptively enhances the packet's security service levels according to a hierarchical scheduling model, which efficiently utilizes the packet transfer using the selection of multiple path to transfer the packet to the destination through the bandwidth that is given for those packets. By using hierarchical scheduling, the proposed system minimizes the flows miss rates and the flows average total delays compared to the earliest-deadline-first (EDF) and the first-come-first-served (FCFS) schedulers. From the other hand, our security enhancement scheme minimizes the pending packets, and the average total packet delays at the end users compared to the EDF (earliest-deadline-first). Through this that the packet is transferred to the destination by using the nodes by selecting the multiple path with their optimal distance is chosen and the cryptographic algorithm is used to encrypt the data by using the key values. Then the packets are splitted and send. Then hierarchical scheduling algorithm is used to schedule the packets to get the reorder packet at the destination then merging the packets and then decrypt the packets with the using the key. Then the contents are decrypted with that the original content is visible at the destination.

Keywords: Cryptographic algorithm, hierarchical scheduling, multiple paths.

I. INTRODUCTION

The internet which plays a vital role in our day to day life and that provides guarantees for the users with the quality of services (QoS).

The QoS which provides the users in the form of the capacity, mean time to transfer and restore, and the delivery delay with such metrics [1]. The real time applications in the network are audio and the video streams have been clogging the internet. For this that the QoS with the real time network application which provides the real time scheduling algorithms [2]. The real time application for packet transfer which has security services and with this that the data is to be more secure from the different types of threats [3],[4]. To provide the security services that the security protocols are implemented transport layer security protocol (TLS), secure socket layer protocol (SSL) and internet protocol security (IPSec). Nowadays real time data streams which does not guarantee for some different classes of protocols [5].

Real time network application which as the technologies to provide more secure for the data and it also reserve the performance for that whole data in the network. With that the network performance metrics (NPMs) is based on the data miss ratio, packet delay, and the throughput [6]. A NPMs is affected by using the key factor in the network buffering system. In this that the buffering system which as the size with the limited buffering for the throughput [7]. The different network based algorithms in the network buffer techniques are routing, maintenance, scheduling, security, balancing the load [8]. There are different methodologies to analyze and measure the performance for the overall data with off - line and on - line monitoring. Such techniques which are used for the monitoring are based on the queuing theory analysis model [9]. The QOS which does not guarantees for security and the data traffics in the real time networks. In the multi-agent system the collaboration and the interaction are designed to interact with the cooperating agents [10]. In our paper, we propose the real time scheduling to provide more secure for the packet transfer.

By using this scheduling that the packets are sending according to the order by using the time of those packets. The features of the proposed system in the real time are

- The proposed system has the functions to provide security services for the real time scheduling.
- The real time scheduling uses the hierarchical scheduling scheduler, the scheduling is based on the resource.
- The security is provided by using the cryptographic algorithms. With this that the cryptographic algorithm like password based encryption and MD5 is used to encrypt the content. Here the encryption is used for the reliable purpose for the content of the data.
- The proposed system which eliminates the ordering for the packets and the security is provided by using the cryptographic algorithms. By using this that the security for the packet transmission is more secure when it compared with the existing system.

The proposed system performance is valued by using the hierarchical scheduler with the cryptographic algorithms is used. In this that the proposed system which reduces size of the queue and here we transfer the transfer without using the queue. In the existed system which has the queue with the limited buffer size and it has the limited packets to be stored within that queue. The diff-Edf scheduler which only schedules the packet based on its packet length and the time taken to transfer that packet from the source to the destination. The packet which transfer in the diff-Edf is based on the time consumption with that the priority is given for the packets having less time is to be prioritized first and that packet is transferred. So at the destination it can't receive the order for the whole packet. For this purpose we are using the hierarchical scheduler to schedule the packet with order.

II. RELATED WORKS

The QoS is applied by using the scheduling algorithms to guarantee the data at the transmission. The scheduler which are given as FCFS this scheduler is used only for the dynamic systems for the first process which is processed first and later the other process is to be process [11]. In the area of networks and data communication, a huge amount of research has been performed to provide network flows from different classes with different levels of QoS guarantees. The data stream flowing through the network, that QoS could be in the different forms of metrics. Different categories of security threats attack different types of flowing data streams in the network; accordingly, data traffic generators are in care of applying security services to their data streams.

The researchers are studying for the effect of applying such guarantees on the overall performance in the network. And they are also trying to execute in network technologies to provide both QoS and security guarantees to their data traffics, while still preserving the overall performance of the network. The literature covers different methodologies that had been used to provide real-time applications with the required QoS guarantees. The proposed system which has a balance between providing the required guarantees to the network's applications and the overall performance of that network. The overall performance of the network is efficiently utilized by enhancing a key factor for the network's buffering system. The literature provides different methods for estimating the availability of the network's buffering system. As a method of providing the requested QoS guarantees to different classes of data flows in a real-time network, in this literature the real-time scheduling has been reviewed. In this we provide an extensive literature about different security protocols that had been adopted to provide the required security services to real-time networks. Scheduling mechanisms using secure ways at different environments will also be reviewed. Real-time agent system was the best method for modeling and analyzing our heterogeneous systems. The limitations of using conventional simulation based systems; accordingly, the literature provides an overview of using such methodology in real-time heterogeneous networks are controlled.

"S. Banerjee, C. Kommareddy, K. Kar, B. Bhattacharjee", proposed [12] real-time environment, the entire system must have the capacity to enforce the required timing constraints on its sub-tasks. Such constraints could be reflected by the associated relative deadline timing parameter. The real-time system should have a mechanism to check the validity of its functionality. The validation process could be achieved by applying two main parameters logical and temporal for the correctness. The logical correctness which checks for generating correct system outputs, and the temporal correctness deals with the system clock. It checks whether system outputs had been generated at the pre-defined instances of time or not. According to the type of real-time data traffic and its requested QoS, that the real-time system implements the appropriate scheduling algorithm such as traffic and guarantees its requested QoS requirements.

The real-time applications share and congest the same integrated real-time network; accordingly, integrated networks should have the capacity to provide different types of services for its real-time data flows. Real-time applications could be in different forms such as audio and video streams, multimedia applications for processing applications, and real-time control applications in real-time.

"I. Norros, J. W. Roberts, A. Simonian, and J. T. Virtamo", proposed [13] fixed data rate model (FDR), the variable data rate model (VDR), and the fixed data rate with variable size model (FDVS). In the FDR model, the generator generates equally-size real-time data units periodically such as real-time control and data processing systems for hard real-time medical applications. VDR model generates equally-size data units asynchronously, where different gaps isolate the stream of data at different instances of time such as the discrete real-time audio systems. In such systems, the data traffic interrupts the scheduler periodically.

The most efficient schedulers for soft real-time systems are the priority based schedulers such as the earliest deadline first (EDF) scheduler, where the task that is closer to expire will be given higher priority over other tasks [14]. Modified versions of EDF scheduler were implemented such as the dynamic queue deadline first (DQDF) scheduler, which integrates functionality of the EDF scheduler with the dynamic queuing model on a single processor environment. DQDF provides an efficient utilization for the system resources with a minimized processing overhead.

1) Resource Allocation for Real-Time Jobs

"K. Wongthavarawat and A. Ganz", proposed Real-time parallel applications with security requirements running on clusters are emerging in many domains, including online transaction processing systems, medical electronics, aircraft control and scientific parallel computing. These applications propose various security requirements like data privacy, data integrity check and software execution protection and thus are fundamentally distinguished by runtime uncertainties that are caused by security needs. For example, in parallel computing, the protection of computationally expensive or irreplaceable data, as well as valuable application software, is critical. In particular, in the business world and the government, where the data is considered sensitive, the potential data losses due to a security incident could be catastrophic. As a result, employing the security services provided by clusters is essential for security-critical real-time parallel applications.

2) Security Services

Using security services to satisfy the applications' security needs, however, incurs security overhead in terms of computation time, which might violate the applications deadlines. The conflicting requirements of good real-time performance and high quality of security protection imposed by security-critical real-time applications introduce a new challenge for resource allocation schemes, that is, how the real-time and security dilemma can be solved.

Moreover, security heterogeneity existing in heterogeneous clusters makes solving this dilemma more difficult, as the security overhead is node dependent, that means for the same level of security service, the different computation nodes incur distinct security overhead. Unfortunately, existing resource allocation schemes for real-time parallel applications on clusters normally do not factor in applications' security requirements when making resource allocation decisions and thus are inadequate for security critical real-time parallel applications.

Hence, security-aware resource allocation schemes must be developed to bridge the gap between the incapability of existing schemes and the need of high quality of security demanded by security-critical real-time applications. Motivated by this discrepancy, in this paper, we design and evaluate two security-aware resource allocation schemes called Task Allocation for Parallel Applications with Deadline and Security constraints (TAPADS)and Security-Aware and Heterogeneity-Aware Resource allocation for Parallel jobs (SHARP) for real-time parallel applications running on homogeneous and heterogeneous clusters, respectively. TAPADS is developed for parallel applications represented by directed acyclic graphs (DAGs), where precedence constraints and communications among tasks in an application exist, whereas SHARP is dedicated to embarrassingly parallel applications with no such precedence constraints and communications.

III. EXISTING SYSTEM

The network system is which as *N* source nodes communicate with *N* destination nodes through secure data channels. The destination nodes are connected to the default gateway, i.e. an edge router, with a star topology. The proposed real-time hierarchical scheduling system is modeled and designed as a real-time multi-agent system [15]. We adopt the agent-oriented methodology. It deals with both the macro (network) level and the micro (agent) level aspects of the design [16]. The agent-oriented methodology is appropriate for our problem because it has the following characteristics

- The goal is to obtain a system that maximizes some global quality measure which may be, however, suboptimal from their point of view of the system components.
- Agents are heterogeneous, because different agents is Implemented by using different programming languages, architectures, and techniques.
- The organization structure of the system is static such that the inter-agent relationship does not change at the run-time.

- The abilities for the agents and there services that they provide are static such that these do not change at run-time.
- The overall system that contains the comparatively small number of different agent types. With this the agent does their job at the given time within the deadline.

The core entities of the multi-agent system are implemented at the edge router, since most of the hacking activities occur at the local-area network level. According to the multi-agent methodology, the system is designed through three phases: decomposition, modeling, and communication protocol. In the decomposition phase, the network architecture is decomposed into three heterogeneous cooperating entities: source, destination, and edge router. Moreover, the edge router is decomposed into four sub-entities are given as: the coordinator, server and Diff- EDF scheduler. In the modeling phase, each entity is modeled with an interactive agent by defining its main tasks and behaviors. In the communication protocol phase, the protocol is specified that governs both interaction and data communication among the multiple real-time agents. Fig. 3.1 shows a diagram of the multi-agent system using diff-edf scheduler. The agent of each entity is described in detail as follows.

A. *Source Agent*

The source agent is the data packet generator. Suppose that it generates one of the two types of real-time data packets: audio or video. Each packet has a fixed size Ps=1500 bytes, which is the maximum Ethernet frame payload. The source agent sends real-time traffic f with a rate of λf. An exponential distribution with mean $1/\lambda f$ is used to model the packet inter-arrival time. A uniform distribution is used to model the relative deadline Df associated with real-time traffic f. A QoS requirement is specified for traffic f in terms of deadline miss ratio Φf . This agent interacts with the coordinator agent by sending requests to serve its real-time traffic with the QoS requirement.

B. *Coordinator Agent*

The coordinator agent is a software agent. It interacts with other agents to regulate their functionalities. The coordinator does not have a global view of the entire system. However, because it locates at the edge router (the default gateway of the LAN), it has the capability of interacting with the source agent using a known IP address and with the destination agent using a known MAC address. The coordinator interacts with the scheduler, and the server to deliver the packets. It also monitors the system behavior and informs other agents to change their behaviors if necessary.

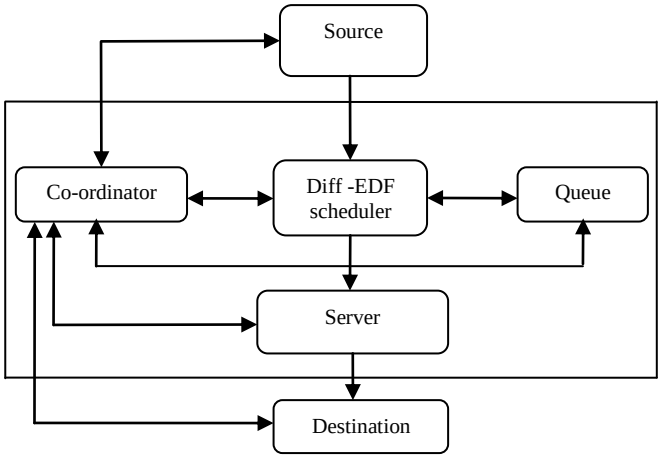

Figure 3.1. multi-agent system using diff- Edf scheduler

C. *Diff-EDF Scheduler Agent*

The EDF scheduler and FCFS agent enforces the timing constraints on the packets to provide the QoS requested by the source. The Diff-EDF is one of the real-time priority scheduling algorithms that are based on the EDF scheduling algorithm. The Diff-EDF scheduler provides smaller miss ratio and shorter average total packet delay at the edge router compared with other real-time hierarchical schedulers. Instead of using the relative deadline as the priority key, the Diff-DEF uses the not effective deadline Def. The hierarchical scheduler applies a shadow function to the packet relative deadline by adding a parameter Cf and generates the effective deadline. It is constructed such that higher priorities are assigned to the video flows over the audio flows. Among the video flows, higher priorities are assigned to the streams with smaller deadline miss ratio Φf.

D. *Server Agent*

The server agent is responsible of serving the real-time data packets that are chosen by the scheduler. It determines to serve or drop a packet based on the packet's remaining time till expiration. If the packet is not expired, the server sends it to the specific destination according to the MAC address with an exponentially distributed service time. An exponential distribution with mean $1/\mu f$ is used for the packet service time, where μf is the packet service rate given by $\mu f = Bw/(8Ps)$. Here where Bw is the average aggregate bandwidth needed for both types of real-time traffics (audio and video). The server records and keeps track of a QoS parameter, the miss ratio, and reports it to the coordinator. The coordinator then adjusts system parameters accordingly in order to meet the QoS requirement.

E. *Buffer Queue Agent*

The buffer queue agent has two processes: the queuing (storing) process and the dequeuing (fetching) process.

In the queuing process, the queue agent places the arriving packets in its buffer according to their effective deadlines. This process is in response to a request from the scheduler. In the dequeuing process, the queue agent fetches the packet that is closest to expire (with the smallest effective deadline) and sends it to the scheduler. The scheduler consequently passes the packet to the server. There is a link from the queue agent to the coordinator, through which the queue notifies the coordinator of its buffer usage. If the buffer usage exceeds a limit, the queue agent sends a message to the coordinator. In response, the coordinator adjusts system parameters to avoid dropping little bits of the real-time packets.

F. Destination Agent

The destination agent performs a FCFS scheduling algorithm on the received packets from the server. It sends two parameters to the coordinator. The first is its processing rate *Pf* of traffic flow *f*. It is sent to the coordinator at the initiation phase. The second is the size *Bf* of its available buffer for accommodating the packets of traffic flow *f*. The coordinator specifies a time period *T*. At the end of every time period, the destination agent sends *Bf* to the coordinator. It is used in the process that determines the packet's not high security service level.

IV. PROPOSED SYSTEM

Most of the previous research has focused on providing Quality of Service (QoS) on session basis, that there is a growing need to support link-sharing in hierarchical way, or QoS guarantees for the traffic that aggregate such as those belonging to the same organization service provider, or the application family. The QoS which supports for both single sessions and traffic aggregates is difficult as it requires the network to meet multiple QoS requirements to be occur simultaneously at different granularities. This problem is exacerbated by the fact that there are no formal models that specify all the requirements. The developed system as an idealized model that is the first to simultaneously capture the requirements of the three important services in an integrated services computer network: guaranteed for the real-time, adaptive best-effort for packets, and link-sharing in hierarchical services.

We then designed hierarchical scheduling algorithms, Hierarchical Packet Fair Queuing (HPS). It is the first algorithm to simultaneously support all three of the real-time, adaptive best-effort for security, and the link-sharing services. HPS allows a more flexible resource management than existing scheduling algorithm. From a conceptual point of view, HPS is the first algorithm that goes beyond Fair Queuing and still satisfies all important requirements in integrated services networks.

Hierarchical scheduling has been proposed as a scheduling technique to achieve aggregate resource partitioning among related groups of threads and applications in uniprocessors and packet scheduling environments and the cryptographic algorithm is used to encrypt and decrypt the data.

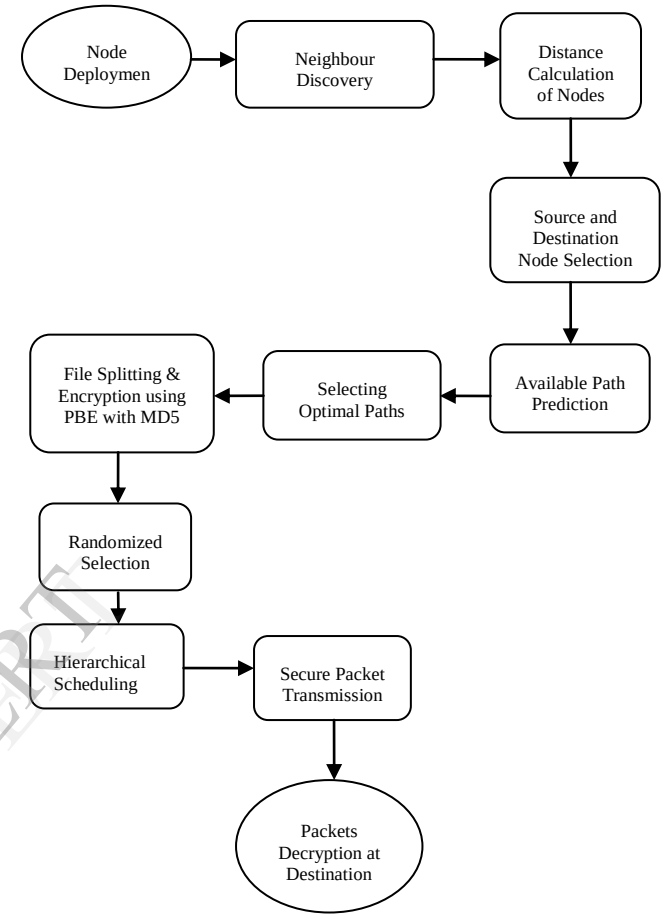

Figure 4.1. Data flow diagram

Existing hierarchical schedulers are not easily extensible to multiprocessor environments because (i) they do not incorporate the inherent parallelism of a multiprocessor system while resource partitioning, and (ii) they can result in unbounded unfairness or starvation if applied to a multiprocessor system in a naive manner. In this paper, we present hierarchical packet scheduling (HPS) a novel hierarchical scheduling algorithm designed for multipath selection in hierarchical structure. The novelty of this algorithm lies in it to achieve desired bandwidth partition among the nodes of the hierarchical scheduling tree. Proposed system is shown below. After deployment of the nodes it will find the neighbors based upon their X & Y co-ordinates to discover neighbor positions. Then we can select the source and destination nodes randomly. Then start calculating the distance between each other nodes from source nodes.

We will predict the optimal path from available paths on the basis of shortest distance to forward our file. After selecting a best path we can now choose a file to forward it to the receiver node. Now we can select a file to further split up the file into multiple packets so as to send through multiple paths. For the secure packet transmission from the sender to the receiver it must be converted to the cipher text using the cryptographic schemes. By using the encryption method we can change the plain text into the intermediate format called cipher text which contains the Symbols so that intruders cannot attack the packet during the packet transmission. The packets contents can be encrypted using the PBE with MD5. The PBE denotes the Password-Based-Encryption using hash function generated keys to encrypt the original contents into the cipher text.

Once the encryption was made now the packets are ready to transmit further to the destination in a secured manner. Then finally the encrypted spitted packets will be forwarded by the random selection of path to reach the destination using our proposed scheduling algorithm called hierarchical scheduling. Once the multiple packets are received at the receiver node it will reorder the packets together to construct the original file streams using its sequence number and can be decrypted to get the source packet contents using the same key for the decryption. The node which obtains the successful the key verification can decrypt the packets remaining node cannot get the source data from any other nodes. Thus we provided the enough file secure guaranties during the transmission among multiple nodes.

V. PERFORMANCE METRICES

In this simulation, we demonstrate the efficiency of using the Diff-EDF algorithm at the scheduler agent over the EDF and the FCFS scheduling algorithms. There are two QoS metrics, the miss ratio at the server agent and average total packet delay at the queue agent, respectively. The figures show that the Diff-EDF scheduler has the least miss ratio and the smallest average total packet delay. Therefore, it is suitable for our time-critical video/audio packets. The edge router for a network with a small number of source-destination pairs, N, handles less number of data streams. Therefore, its chance of missing the traffic's deadline is low and the average total packet delay is also low. On the other hand, a network with a large N suffers more with large miss ratio and large packet delay, as the edge router serves different real-time sources according to their QoS requirements. One of the challenges in the wireless sensor applications which are gaining much attention is the real-time transmission of continuous data packets across the network.

Though advances in communication in networks are providing guaranteed quality data packet delivery they still have some drawbacks. One such drawback is transmission of incessant data packets over high speed networks. Here in this paper we have designed a network having with no buffer just not at the sink but also in selected intermediate nodes to minimize the packet loss caused due to congestion. In this approach the results is haggle congestion and less packet loss in the designed network. From the other hand, the security enhancement scheme eliminates the buffer consumption.

The Proposed work that minimizes the average total packet delays when it received at destination, the pending packets at the end users compared to the existing work. As the performance metrics the proposed system eliminated the repeated security associations performed by the where our adaptive security system hence less overhead and increases the chances to meet the flows QoS requirements. We estimate the performance of the proposed scheme at the destination agent. With the security enhancement process in the network, the security levels of the real-time audio and video streams reach the steady state when there are no more level-changes with the feedbacks.

1) *Average Total Packet Delay*

For the N packets arrival the buffer at the same time every (L / R) * N seconds: The first packet transmitted has no queuing delay; the second packet transmitted has a queuing delay of L / R seconds; and more generally, the nth packet transmitted has a queuing delay of ((n -1) * L / R) seconds. Total queuing delay = 0 + L / R + 2 * (L / R) + ……. + ((n - 1) * L / R) = (N * (0 + ((n - 1) * L / R))) / 2

= (n * (n - 1) * (L / R)) / 2 (1)

Average queuing delay time = Total delay / n = ((n -1) * (L / R)) / 2. (2)

2) *Miss Ratio Calculation*

Hit ratio = percentage of memory accesses satisfied by the cache. Miss ratio = 1-hit ratio

3) *Number of Packets in the Queue*

For the discussion on queuing systems, the book uses the term "traffic intensity, I", I=a L/R where L is the packet size, R is the link speed and the a is the load in packets/second. Note that I is defined to lie between the 0 and 1. In the case of I=1, there are exactly as many packets arriving per unit of time (say1second), as can be serviced in that same unit of time (say1second). In other words, since I=a*L/R and according to the above definition, a is lower bounded by 0 and upper bounded by R/L. Clearly, to know what the values is to make sense for a given queuing system, we have to know the packet size L and the link speed R.

4) Bandwidth Calculation

The Maximum bandwidth that can be calculated as follows: Throughput= (RWIN/RTT) Where, RWIN is the TCP Receive Window and RTT is the round-trip time for the path. The Maximum TCP Window size for the absence of TCP window scale option is 65,535 bytes. Example: Max Bandwidth which is given as = 65535 bytes / 0.220 s = 297886.36 bytes/s * 8 = 2.383 Mbit/s. We multiply the Byte per second times 8 to get the Bit per second rate. A single TCP connection between those endpoints is to tested Bandwidth will be restricted to the 2.376 Mbit/s even if the contracted Bandwidth is to be greater. The performance for the proposed system when it compared with the existing system is given as

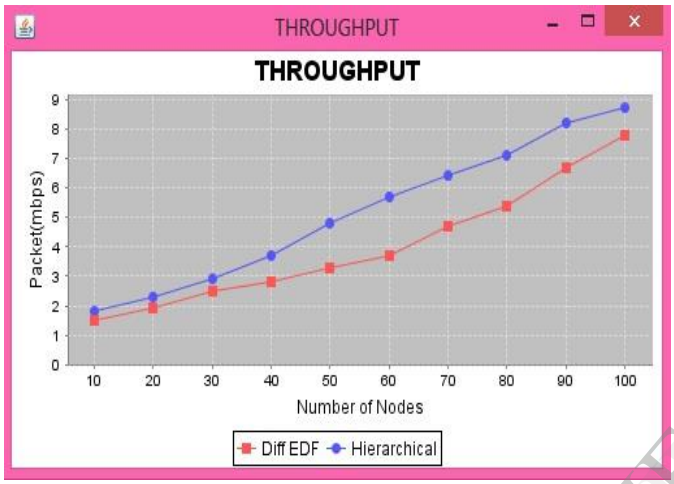

Figure 5.1. Throughput for Hierarchical Scheduler

The above graph which shows the throughput for the proposed system which is better when it compared with existed system. Here the proposed system which plays a good result for the throughput with packet mbps is attaining a high throughput.

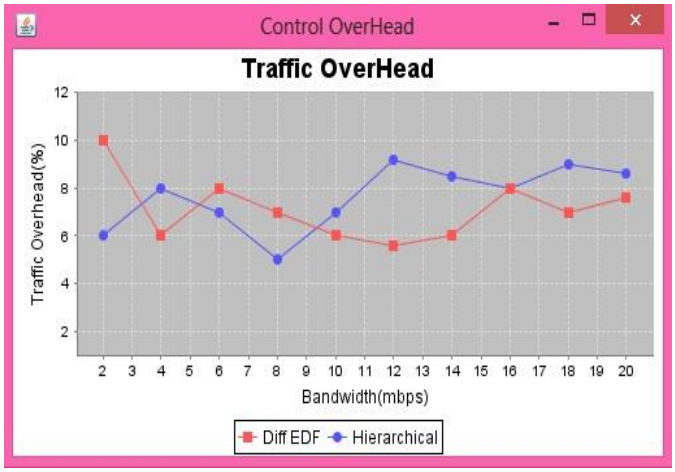

Figure 5.2. Control Overhead Of Hierarchical Scheduler Compared With Diff-Edf Scheduler

Form the above graph that the traffic overhead for the packet which is controlled by using the hierarchical scheduler which controls the traffic overheads when it compared with the differentiated Edf- scheduler.

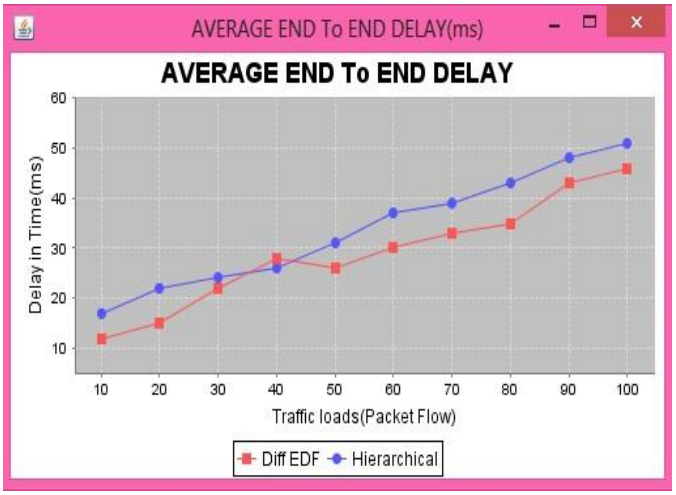

Figure 5.3. Comparing the Average End To End Delay With Diff-Edf Scheduler and Hierarchical Scheduler

From the above graph that the average end to end delay is minimized when it compared with the diff-Edf scheduler while it transferring the packets from the source to the destination.

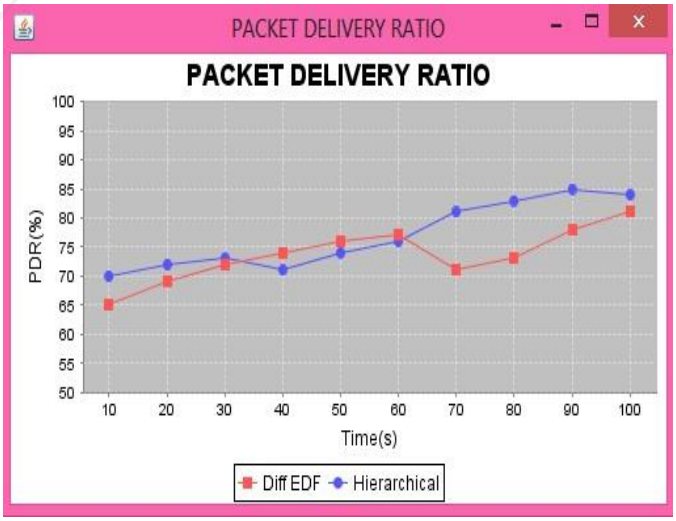

Figure 5.4. Packet Delivery Ratio of Hierarchical Scheduler Compared With Diff-Edf Scheduler

From the graph that shows the hierarchical scheduler which delivers the packet within the time with the same order as it is from the source to destination. So the delivery ratio is good when it compared with the diff-Edf scheduler. In the existing system that the packet delivery ratio is not occur properly due to the traffic in the packets.

VI. CONCLUSION AND FUTURE ENHANCEMENT

The proposed system combines the functionality of real-time scheduling with the security service enhancement, where the real-time scheduling unit uses the hierarchical scheduler, while the security service enhancement scheme adopts timing with the packet transfer and it reaches the destination within the estimated time. The proposed system provides the required QoS guarantees for different classes of real-time data flows (video, audio), while adaptively enhances the packet's security service levels according to hierarchical scheduling, and thus protects the network from being congested by heavy traffic load. In our feature enhancement we will implement this system in the internet network and local area network by using the internet protocol suite with link layer technologies.

REFERENCES

[1] J. Silvestre-Blanes, L. Almeida, R. Marau, and P. Pedreiras, "Online QoS management for multimedia real-time transmission in industrial networks," *IEEE Trans. Ind. Electron.*, vol. 58, no. 3, pp. 1061–1071, Mar. 2011.

[2] A. Gupta, D. Ghosh, and P. Mohapatra, "Scheduling prioritized services in multihop OFDMA networks," *IEEE/ACM Trans. Netw.*, vol. 18, no. 6, pp. 1780-1792, Dec. 2010.

[3] Y. Jung and M. Peradilla, "Tunnel gateway satisfying mobility and security requirements of mobile and IP-based networks," *J. Commun. and Networks*, vol. 13, no. 6, pp. 583–590, Dec. 2011.

[4] F. Hashim, K. S. Munasinghe, and A. Jamalipour, "Biologically inspired anomaly detection and security control frameworks for complex heterogeneous networks," *IEEE Trans. Network and Service Management*, vol. 7, no. 4, pp. 268–281, Dec. 2010.

[5] T. Xie and X. Qin, "Scheduling security-critical real-time applications on clusters," *IEEE Trans. Comput.*, vol. 55, no. 7, pp. 864–879, Jul. 2006.

[6] A. Goldsmith, M. Effros, R. Koetter, M. M´edard, A.Ozdaglar, and L. Zheng, "Beyond Shannon: The quest for fundamental performance limits of wireless ad hoc networks," *IEEE Commun. Mag.*, vol. 49, no. 5, pp. 195–205, May 2011.

[7] M. Daneshtalab, M. Ebrahimi, P. Liljeberg, J. Plosila, and H. Tenhunen, "Memory-efficient on-chip network with adaptive interfaces," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, no. 1, pp. 146–159, Jan. 2012.

[8] A. Aziz, D. Starobinski, and P. Thiran, "Understanding and tackling the root causes of instability in wireless mesh networks," *IEEE/ACM Trans.Netw.*, vol. 19, no. 4, pp. 1178–1193, Aug. 2011.

[9] A. Bulut, N. Koudas, A. Meka, A. K. Singh, and D. Srivastava, "Optimization techniques for reactive network monitoring," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1343–1357, Sep. 2009.

[10] J.Y. Wang and Z. Zhu, "Integration system of network information resources based on multi-agent collaboration," in *Proc. 2011 International Conference on Fuzzy Systems and Knowledge Discovery*, vol. 3, pp. 2044–2049.

[11] D. Tsafrir, Y. Etsion, and D. G. Feitelson, "Backfilling using system generated predictions rather than user runtime estimates," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 6, pp. 789–803, Jun. 2007.

[12] S.Banerjee, C. Kommareddy, K. Kar, B. Bhattacharjee, and S. Khuller, "Construction of an efficient over multicast infrastructure for real-time applications", Proceedings, vol.2,pp. 1521-1531, 2003.

[13] I. Norros, J. W. Roberts, A. Simonian, and J. T. Virtamo, "The superposition of variable bit rate sources in an ATM multiplexer", *IEEE Journal on Selected Areas Communications*, vol. 9, no. 3, pp. 378-387, 1991.

[14] V. Sivaraman and F. Chiussi, "Providing end-to-end statistical delay guarantees with earliest deadline first scheduling and per-hop traffic shaping", *Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM*), vol. 2, pp. 631-640, 2000.

[15] F. Zambonelli, N. R. Jennings, and M. Wooldridge, "Developing multiagent systems: The Gaia Methodology," *ACM Trans. Software Eng. Methodology*, vol. 12, no. 3, pp. 317–370, Jul. 2003.