ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

HANDLING HEAVY-TAILED TRAFFIC IN QUEUEING NETWORKS USING MAX WEIGHT ALGORITHM

M.Sathish Kumar *, G.Sathish Kumar

* Department of ECE, MNSK College of Engineering, Pudukottai, Tamilnadu, India

Department of ECE, MNSK College of Engineering, Pudukottai, Tamilnadu, India

ABSTRACT

All scheduling systems consider the delay properties of one-hop and multi-hop networks with general interference constraints and multiple traffic streams with time-correlated arrivals. We first treat the case when arrivals are modulated by independent finite state Markov chains. To show that the well known maximal scheduling algorithm achieves average delay that grows at most logarithmically in the largest number of interferers at any link. Further, in the important special case when each Markov process has at most two states (such as ON/OFF sources), we prove that average delay is independent of the number of nodes and links in the network, and hence is order-optimal. The system has to analyze the delay performance of a multi-hop wireless network with a fixed route between each source-destination pair. To develop a new queue grouping technique to handle the complex correlations of the service process resulting from the multi-hop nature of the flows. For the tandem queue network, where the delay-optimal policy is known, the expected delay of the optimal policy numerically coincides with the lower bound. To achieve the time based as well as efficient packet based data transmission in multi-hop networks that the average delay of the back-pressure scheduling policy can be made close to the lower bound by using appropriate functions of queue length.

KEYWORDS: Multi-hop network, Global recovery agent, Global failure detection agent, Queue network.

INTRODUCTION

Dynamic distributed systems and networks considerably increases with the increasing complexity of their services and policies, system administrators attempt to ensure high quality of services each user requires by maximizing utilization of system resources. To achieve this goal, correct, real-time and efficient management and monitoring mechanisms are essential for the systems. But, as the infrastructures of the systems rapidly scale up, a huge amount of monitoring information is produced by a larger number of managed nodes and resources and so the complexity of network monitoring function becomes extremely highly.

Network Environments

Also, there are heterogeneous and various network environments within the systems needed to be monitored and the nature of managed resources becomes almost dynamic, not static, which forces traditional static centralized and distributed monitoring mechanisms to be unsuitable for the systems. Thus, mobile agent-based monitoring mechanisms have actively been developed to monitor these large scale and dynamic distributed networked systems adaptively and efficiently.

Agent Based Mechanisms

Mobile agent is an autonomous and independent software program to satisfy the corresponding user's goal on behalf of the user while visiting various target nodes through a network. This mobile agent technology has several advantages such as reduction of network traffic, overcoming of network delay, enabling asynchronous execution and enhancement of dynamic adaptability. Thanks to these desirable features, this technology is very widely used in distributed systems, especially for network management. In a network management system, each mobile agent is generally designed to move to one or more agent-executable nodes in a network, sense temporally and permanently other nodes and resources, and filter and deliver the received management information to the appropriate network management nodes.

The previous mobile agent-based monitoring mechanisms are classified as follows: centralized and hierarchical distributed monitoring mechanisms. Most of them are based on the centralized monitoring model and divided into two categories, single mobile agent-based and segment-based mechanisms. In the first single management station creates a mobile agent and allows the agent to sequentially visit the required nodes in a particular order. This mechanism is simple to implement, but causes the task completion time of a mobile agent to become too long in large-scale distributed systems because the number of visiting nodes significantly increases and the size of the agent may grow considerably. In particular, if the visiting nodes are interconnected through low bandwidth links, the round-trip delay may extremely increase. Secondly, the segment-based mechanism partitions a network into several sub-networks or domains, and creates and transfers a mobile agent to each domain respectively.

Therefore, the collection and filtering of the management information for monitored nodes can be performed in parallel per domain, which addresses the scalability problem of the first mechanism to a certain extent. However, in this mechanism, the single manager should execute all the monitoring function and may become the performance bottleneck of the entire system. In addition, if the agent migration network includes expensive low bandwidth links, it is very difficult to perform the procedure to obtain and filter the monitoring information in real-time.

Hierarchical Monitoring Mechanisms

The original hierarchical monitoring mechanisms were almost based on a static manager organization model. In other words, each network administrator configures a tree of network domains according to its initial monitoring policy and then the main manager at the root domain creates and migrate monitoring manager agents to other domains. However, if any dynamic changes in various aspects such as network traffic patterns, resource addition and deletion, network topology and so on occur, this mechanism cannot adapt to these changes and will degrade significantly the entire management performance. There were presented some adaptive mobile agent-based mechanisms to address this important issue. In these mechanisms, if each domain manager at level i estimates the need for some additional monitoring capability at run-time, it creates and installs a new manager agent to an appropriate node at level i+1 or migrates to another node for keeping location optimality of its network monitoring.

However, failures of some domain managers even assuming the main manager can be reliable using replication-based fault-tolerance mechanisms are very critical to providing correct, real-time and efficient monitoring functionality in a large-scale mobile agent based distributed monitoring system. To the best of our knowledge, the fault-tolerance mechanism proposed only one to address this issue. But, in this mechanism, every agent should periodically send heartbeat messages to global failure detection agents (GFDAs). If the GFDA receives no heart-beat message from an agent for a predefined number of consecutive timeout intervals, it generates and delivers an Agent Failure message to a global recovery agent (GRA). Afterwards, the GRA recreates a new agent based on its most recent configuration information and redeploys it to the appropriate target host. However, this behavior results in high failure-free overhead due to the centralization of failure detection functionality in a single point within a large-scale hierarchical monitoring organization.

RELATED WORK

The major part of the project development sector considers and fully survey all the required needs for developing the project. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, man power, economy, and company strength. Once these things are satisfied and fully surveyed, then the next step is to determine about the software specifications in the respective system such as what type of operating system the project would require, and what are all the necessary software are needed to proceed with the next step such as developing the tools, and the associated operations.

The stability works all use backlog-based transmission rules, which treat joint stability and utility optimization. However works introducing an interesting delay-based Lyapunov Function for providing stability, where the delay of the head-of-line packet is used as a weight in the max-weight decision. This approach naturally provides tighter control of the actual queuing delays.

We consider a simple distributed scheduling strategy, maximal scheduling, and prove that it attains a guaranteed fraction of the maximum throughput region in random wireless networks. The guaranteed fraction depends on “interference degree” of the network which is the maximum number of sessions that interfere with any given session in the network and do not interfere with each other. The outcome of throughput utilization is also suffered with those kinds of dependencies. Depending on the nature of communication, the transmission powers and the

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

transmission models, the guaranteed division can be lower bounded by the maximum link degrees dependencies beyond the bandwidth, or even by constant delivery of packets without the knowledge of server life. The guarantees also hold in networks with multicast communication and an arbitrary number of frequencies. Hence the guarantees are tight in that they cannot be improved any further with maximal scheduling.

- If the packet contain large size stuff, the stability of the network holds
- Tighter control of the actual queuing delays
- Packets suffers based on dependency because the scheduling process is equally applied for all hops in the network
- Low throughput is attained
- Requires higher bandwidth
- Low Performance
- Server Lifetime depends on the hops data delivery
- Difficult for multicast communication
- Cannot improve the Maximal Scheduling process because of the dependencies in bandwidth, time, and traffic.
- Requires high cost to manage the network.

Markov Chains

A Markov chain is a stochastic process with the Markov property on a finite or countable state space. The term "Markov chain" refers to the sequence (or chain) of states such a process moves through. Usually a Markov chain is defined for a discrete set of times (i.e., a discrete-time Markov chain) although some authors use the same terminology to refer to a continuous-time Markov chain. The use of the term in Markov chain Monte Carlo methodology covers cases where the process is in discrete time with a continuous state space.

Discrete Algorithm

A discrete algorithm is a system with a countable number of states. Discrete systems may be contrasted with continuous systems, which may also be called analog systems. A final discrete system is often modeled with a directed graph and is analyzed for correctness and complexity according to computational theory.

Maximal Scheduling Algorithm

Maximum scheduling algorithm highly concentrates on throughput, which is a procedure for scheduling data packets in a packet-switched best-effort communications network, typically a wireless network, in view to maximize the total throughput of the network, or the system spectral efficiency in a wireless network. This is achieved by giving scheduling priority to the least "expensive" data flows in terms of consumed network resources per transferred amount of information.

Min-Max Algorithm

Minmax (sometimes minimax) is a decision rule used in decision theory, game theory, statistics and philosophy for minimizing the possible loss for a worst case (maximum loss) scenario. Alternatively, it can be thought of as maximizing the minimum gain (maximin). Originally formulated for two-player zero-sum game theory, covering both the cases where players take alternate moves and those where they make simultaneous moves, it has also been extended to more complex games and to general decision making in the presence of uncertainty.

Lyapunov Function

Lyapunov functions are scalar functions that may be used to prove the stability of an balance of an ODE(Ordinary Differential Equations).For many classes of ODEs, the existence of Lyapunov functions is a necessary and sufficient condition for stability. Whereas there is no general technique for constructing Lyapunov functions for ODEs, in many specific cases, the construction of Lyapunov functions is known. For instance, quadratic functions suffice for systems with one state; the solution of a particular linear matrix inequality provides Lyapunov functions for linear systems; and conservation laws can often be used to construct Lyapunov functions for physical systems. Informally, a Lyapunov function is a function that takes positive values everywhere except at the stability in question, and decreases (or is non-increasing) along every trajectory of the ODE. The principal merit of Lyapunov function-based stability analysis of ODEs is that the actual solution (whether analytical or numerical) of the ODE is not required.

Scheduling Algorithm

Scheduling algorithm is the method by which threads, processes or data flows are given access to system resources (e.g. processor time, communications bandwidth). This is usually done to load balance a system effectively or achieve a target quality of service. The need for a scheduling algorithm arises from the requirement for most modern systems to perform multitasking (execute more than one process at a time) and multiplexing (transmit multiple flows simultaneously.

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

PROPOSED SCHEME

The proposed system considers the problem of scheduling for maximum throughput utility in a network with random packet arrivals and time varying channel reliability. The system considers on Hop technique, where each packet requires transmission over a single link.

At every slot the network controller assesses the condition of its channels and selects a set of links for transmission. The current system derives average delay bounds for one-hop wireless networks that use maximal scheduling subject to a general set of interference constraints. In particular, when arrival processes are modulated by independent Markov processes, we show that average delay grows at most logarithmically in the number of nodes in the network. Existing work provides explicitly computable and order-optimal delay bounds for time-correlated arrivals. Our work addresses the issues of general interference constraints and time-correlated "bursty" traffic simultaneously. We treat the general interference model use the concept of queue grouping to derive the order-optimal delay results. Queue grouping techniques have been used in to reduce scheduling complexity in switches and wireless networks.

- Optimized delay rules is applied, so if the packet contain large size stuff, the stability of the network remains stable
- Queuing delays depends on the Server capacity
- Packets transferred to the server based on the receiving capacity of the server
- Attains High throughput
- Bandwidth is low
- High Performance
- Server Lifetime increases, because the transmission ratio depends on the server
- Efficient way to Multicast communication
- Maximal Scheduling process increases its ability to schedule the packets in all source ends.
- Cost efficient transmission mode.

ARCHITECTURAL DESIGN

The major part of the project development sector considers and fully survey all the required needs for developing the project. Once these things are satisfied and fully surveyed, then the next step is to determine about the software specifications in the respective system such as what type of operating system the project would require, and what are all the necessary software are needed to proceed with the next step such as developing the tools, and the associated operations. Generally algorithms shows a result for exploring a single thing that is either be a performance, or speed, or accuracy, and so on. System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behavior) between them.

Initially all the nodes connected with Server have to receive the confirmation from server about its connection, after getting the response the nodes need to know the Router Information for transferring Packets. If the Router is Live the Packets are forwarded to the destination as well, otherwise the packets are forwarded to Queue for its internal accessing, whenever the Router becomes Live all the packets presented in the Queue will forwards to the destination after all the packets get transferred the Current packet reaches the destination as in manner.

In these mechanisms, if each domain manager at level i estimates the need for some additional monitoring capability at run-time, it creates and installs a new manager agent to an appropriate node at level i+1 or migrates to another node for keeping location optimality of its network monitoring. However, failures of some domain managers even assuming the main manager can be reliable using replication-based fault-tolerance mechanisms are very critical to providing correct, real-time and efficient monitoring functionality in a large-scale mobile agent based distributed monitoring system. To the best of our knowledge, the fault-tolerance mechanism proposed in [is the only one to address this issue. Lyapunov functions are scalar functions that may be used to prove the stability of a balance of an ODE (Ordinary Differential Equations).For many classes of ODEs; the existence of Lyapunov functions is a necessary and sufficient condition for stability. Whereas there is no general technique for constructing Lyapunov functions for ODEs, in many specific cases, the construction of Lyapunov functions is known. For instance, quadratic functions suffice for systems with one state; the solution of a particular linear matrix inequality provides Lyapunov functions for linear systems; and conservation laws can often be used to construct Lyapunov functions for physical systems. Informally, a Lyapunov function is a function that takes positive values everywhere except at the stability in question, and decreases (or is non-increasing) along every trajectory of the ODE. The principal merit of Lyapunov function-based stability analysis of ODEs is that the actual solution (whether analytical or numerical) of the ODE is not required.

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

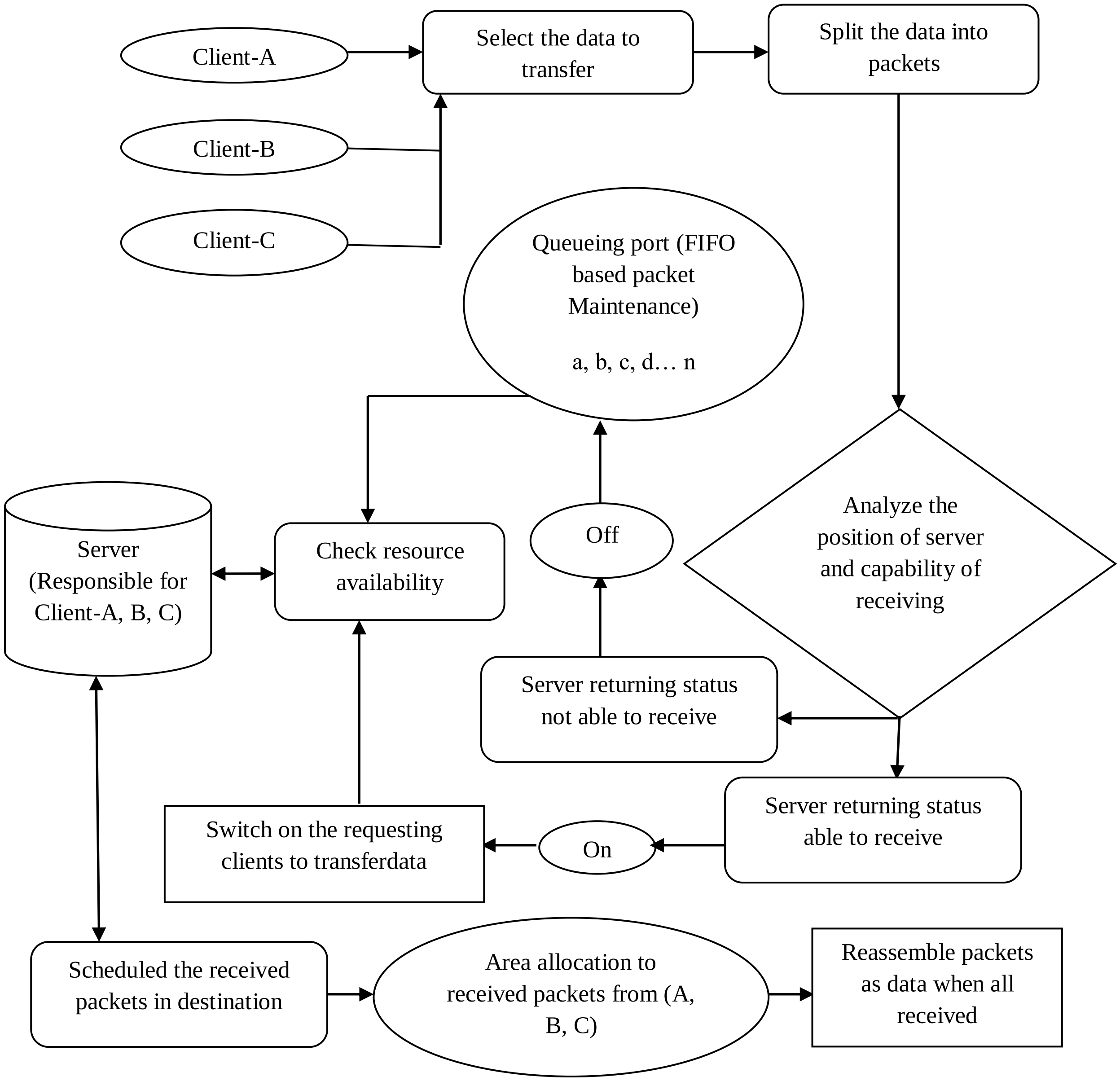

Fig 1: System Architecture

METHODOLOGY

Following are the most frequently used project management methodologies in the project management practice:

1. Network Architecture
2. Node Status Identification
3. Fault Mechanism
4. Maximal Scheduling Precedence
5. Optimal Data Processing

Network Architecture

Client-server computing or networking is a distributed application architecture that partitions tasks or workloads between service providers (servers) and service requesters, called clients. Often clients and servers operate over a computer network on separate hardware. A server machine is a high-performance host that is running one or more server programs which share its resources with clients. A client also shares any of its resources; Clients

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

therefore initiate communication sessions with servers which await (listen to) incoming requests.

Node Status Identification

In this Module we identify the Node is weather live or not. In this process we easily identify the status of the node and also easily identify the path failure.

- Message Transmission
- Change status
- Update status

Message Transmission:

A. In the module we just transfer the message to the destination or intermediate nodes.
B. The intermediate node just forwards the message to destination.
C. The receiver receives the message and sends the Ack.

Change Status: In this Module we identify the changed status of node. The Status is Live, Uncertain and Down

Update Status: In this module we update the status of the node. Then only we can identify whether the node is live or not.

Fault Mechanism

In these mechanisms, if each domain manager at level i estimates the need for some additional monitoring capability at run-time, it creates and installs a new manager agent to an appropriate node at level i+1 or migrates to another node for keeping location optimality of its network monitoring. However, failures of some domain managers even assuming the main manager can be reliable using replication-based fault-tolerance mechanisms are very critical to providing correct, real-time and efficient monitoring functionality in a large-scale mobile agent based distributed monitoring system. To the best of our knowledge, the fault-tolerance mechanism proposed in [is the only one to address this issue.

Maximal Scheduling Precedence

Maximum scheduling usually highly concentrates on throughput, which is a procedure for scheduling data packets in a packet-switched best-effort communications network, typically a wireless network, in view to maximize the total throughput of the network, or the system spectral efficiency in a wireless network. This is achieved by giving scheduling priority to the least "expensive" data flows in terms of consumed network resources per transferred amount of information.

Optimal Data Processing

Scalar functions that may be used to prove the stability of an balance of an ODE(Ordinary Differential Equations).For many classes of ODEs, the existence of Lyapunov functions is a necessary and sufficient condition for stability. Whereas there is no general technique for constructing Lyapunov functions for ODEs, in many specific cases, the construction of Lyapunov functions is known. For instance, quadratic functions suffice for systems with one state; the solution of a particular linear matrix inequality provides Lyapunov functions for linear systems; and conservation laws can often be used to construct Lyapunov functions for physical systems. Informally, a Lyapunov function is a function that takes positive values everywhere except at the stability in question, and decreases (or is non-increasing) along every trajectory of the ODE.

CONCLUSION

Our approach produces cost-effective data transfer because it requires less number of switches and links than other recently proposed structures for data centers, but the time delay is proportional to other issues. In future the system should be designed with traffic-aware routing in reliable time variations to make better utilization of the link capacities according to traffic states and the solution is to increase the bisection width in complete fault tolerant during its incremental deployment. In future the analysis should admit a large class of arrival processes. Also, the analysis can be readily extended to handle channel variations. The main difficulty in this system is in identifying the bottlenecks in the system. The lower bound not only helps us identify near-optimal policies, but may also help in the design of a delay-efficient policy as indicated by the numerical studies, but in future all the difficulties can be solved and propose the architecture with better results.

FUTURE WORK

In future the implemented system is further extended by means of Max-Scheduling process because the present system works with single client but in future it will be modified by means of other two clients, so there is a need of scheduling based on data. The fault tolerant mechanism is also improvised to suitable for the other two clients and provides a facility to work more efficiently than the proposed approach.

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

ACKNOWLEDGEMENTS

We would like to sincerely thank Assistant Prof. G.Sathish Kumar for his advice and guidance at the start of this article. His guidance has also been essential during some steps of this article and his quick invaluable insights have always been very helpful. His hard working and passion for research also has set an example that we would like to follow. We really appreciate his interest and enthusiasm during this article. Finally we thank the Editor in chief, the Associate Editor and anonymous Referees for their comments.

REFERENCES

[1] M. Isard, M. Budiu, and Y. Yu et al., “Dryad: Distributed data-parallel programs from sequential building blocks,” in Proc. ACM EuroSys, 2007, pp

[2] M. Al-Fares, A. Loukissas, and A. Vahdat, “A scalable, commodity data center network architecture,” in Proc. ACM SIGCOMM, Aug. 2008, pp. 63–74.ymantec, “Internet Security Threat Report,” vol. 6, 2011.

[3] C. Guo, H.Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, “DCell: A scalable and fault-tolerant network structure for data centers,” in Proc. ACM SIGCOMM.

[4] M. J. Neely, “Delay-based network utility maximization,” in Proc. IEEE INFOCOM, Mar. 2010, pp. 1–9.

[5] L. Georgiadis, M. J. Neely, and L. Tassiulas, “Resource allocation and cross-layer control in wireless networks,” Found. Trends Network vol. 1, no. 1, pp. 1–149, and 2006.

[6] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” IEEE Trans. Autom. Control, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.