

Agent based Monitoring and Scheduling Technique Framework for Backup Configurations

Rekha V.
Associate Professor
Dept of CSE
SJBIT, Bangalore

M. Indiramma, PhD
Professor
Dept of CSE
BMSCE, Bangalore

ABSTRACT

In Multi Route Configurations (MRC) protocol when failure is detected, it forwards the packets over preconfigured alternative next-hops and immediately recovers from single or multiple failures. Multipath routing provides multipath communication and improves network transmission reliability. However, when there is a huge amount of traffic in the network existing MRC follows default scheduling which results in large transmission delay for high priority packet. Hence it is a challenging problem, and highly influences the performance of MRC. In this paper, we propose an Agent based Monitoring and Scheduling Technique (AMST) algorithm which provides a generic framework for packet scheduling. The main idea of AMST algorithm is to schedule the packets based on its priority and by selecting the transmission path with lowest transmission delay.

Keywords

Delay, Scheduling, Bandwidth, Priority, Congestion, Jitter, Reorder, Multipath TCP.

1. INTRODUCTION

Internet is growing fastly and it is entirely different from what it was earlier. The user demand for bandwidth is growing with the development of the Internet services. However, the Internet resources, especially the limited bandwidth resources, cannot be effectively used because the existing internet uses single-path transmission. Using multiple path interfaces for data transmission, this can lead to higher throughput and robustness [1].

Today end hosts often have multiple interfaces to access the World Wide Web. Multiple interfaces closes gap between the multipath network and the single-path transmission. Multipath TCP provides the multiple paths between two end-hosts for the transmission of a single data

stream [2]. Hence for such kind of transmission important parameters like throughput, end-to-end delays and also buffer space requirements become more relevant [3].

MRC [4] is a proactive and local protection mechanism that allows fast recovery. When a failure is detected, MRC forwards the packets over pre-configured alternative next hops immediately. Since no global re-routing is performed, fast failure detection mechanisms like fast hellos or hardware alerts can be used to trigger MRC without compromising network stability [5]. The shifting of

recovered traffic to the alternative link may lead to congestion and packet loss in parts of the network [6]. Ideally, a proactive recovery scheme should not only guarantee connectivity after a failure, but also do so in a manner that does not cause an unacceptable load distribution.

MRC does not provide any approach, to capture Quality of Service (QoS) in the event of failure. Hence there is a need to focus research on QoS. Without QoS capabilities, it is impossible to offer services to different applications.

MRC primarily designed to provide multi path and backup configurations. In case of network failure, MRC concentrate only on calculating alternate paths to destination using default scheduler. However this scheduling decision in network may leads to transmission delay and also results in head-of-line blocking or receive-window limitation [7] and this occurs majorly when available paths are heterogeneous. In such a scenario, high priority applications experience high delays and resulting in lower throughput. Therefore, the scheduler can have a significant impact on the performance of Multipath Transmission [8]. Hence there is a need to focus on the implementation of a scheduler for the distribution of data over multiple paths during transmission. An Agent based scheduling frame work is proposed for the MRC as a part of QoS module.

The remainder of this paper is organized as follows. We review the previous works related to scheduling in Section II. In Section III Proposed frame work Architecture is discussed. In Section IV detailed implementation of Agent Monitoring Scheduling Technique (AMST) Algorithm is presented. In section V Performance Evaluation and experimental work is discussed.

2. RELATED WORK

MRC is based on building set of backup configurations that are used to route recovered traffic on alternate paths after failure, but this lacks QoS parameters such as bandwidth optimization, priority based scheduling and packet loss. Hence there is a need to focus on priority based scheduling for Multiple Path Recovery Protocol.

Different routing algorithms and scheduling of packets are presented in this section. The basic problems that occur during transmission of data packets across the multiple paths are: i) lack of proper scheduling decision or priority. ii) delay in data delivery [7]. Hence applications will need to add large amount of buffer, forcing end systems to cope with burstiness and provide continues streaming experience to the end user.

There are several scheduling algorithms which perform the scheduling of packets in network and one of them is Round Robin scheduler (RR). This technique selects one sub flow after the other in round robin fashion. This scheduling algorithm guarantees the capacity of each path by using distribution mechanism. However in case of heavy load of incoming packets RR scheduler may not perform well and such scheduler are used for Stream Control Transmission Protocol (SCTP) [9]. Transmission over concurrent paths evaluates performance of RR schedulers for multi streaming

compared to a scheduler that assigns each stream to specific path.

To schedule the different SCTP streams across the different SCTP associations has been analyzed in [10]. Since standard SCTP design does not support the transmission of a single stream across different paths. With this concurrent multipath transfer has been introduced for SCTP [9] and still exposes to scheduling problem similar to the multipath transmission. Multipath TCP schedulers consider delay of each path into account and minimize the head of line blocking [11]. QoS improve services to the network users with guaranteed dedicated bandwidth and timely delivery of data packets with high priority to a particular destination. Hence in this paper we are proposing AMST algorithm to schedule the packets based on its priority and transmission path delays.

3. PROPOSED FRAME WORK

In this paper, we propose a Agent based monitoring and scheduling technique (AMST) algorithm, a generalized frame work module for MRC [4]. Agent is the software module implemented in MRC layer of network, which performs packet scheduling by considering sender and receiver transmission delays and also overcome the problem of receiver buffer blocking during the period of transmission. Figure 1 shows an overview of the fundamental structure of AMST.

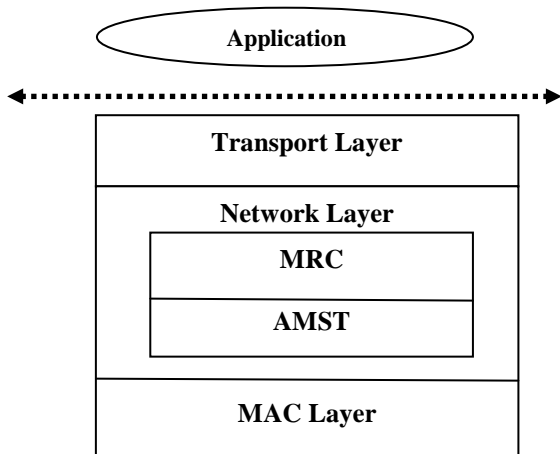


Figure 1 Layered Architecture of Agent Monitoring and Scheduling Technique (AMST) between network and MAC layer.

In this AMST based Frame work the applications and network devices need to deal with Multi Route Configurations (MRC) which handles the route construction and calculation of shortest path. AMST is embedded over the MAC layer to perform packet scheduling and transmission of packet in order. AMST decides which packet to be dispatched via different path. Data packets need to be received in the same order as transmitted at the sender. AMST provides a technique which leads to reduce reordering effort. Control Packets (CP) are sent to all nodes in network and path score is calculated based on information. Then the packets are scheduled on path with lower transmission delay.

Figure 2 shows the detailed design of AMST, complete Scheduling Blocks embedded on MAC layer.

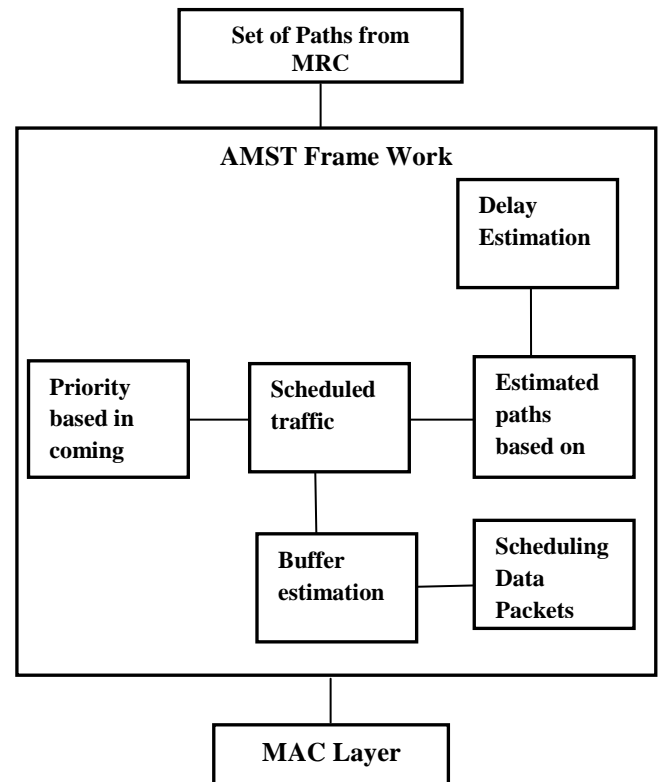


Figure 2. AMST generalized scheduling frame work for packet scheduling based on Priority.

The above frame work has the following sub modules:

- **Delay estimation** block is used to calculate differences in sender and receiver delays between paths. Implemented using Agent Delay Calculation (ADC) metric and its sub module is Agent_ADC ().
- **Priority based in coming Traffic** block used to find the priority of incoming packet and store it in priority queue. Implemented using priority of data packets and its sub module is Agent_Priority ().
- **Buffer estimation** block deals with the problem of receiver buffer blocking. Implemented using Agent Buffer Avoidance metric and its sub module is Agent_ABA ().
- **Scheduling Data Packets** block used to schedule the packets based on priority by selecting path with lowest delay. Implemented using Agent_Schedule ().

3.1 Agent Monitoring Scheduling Technique (AMST) algorithm

In this section, we present the details of our proposed Agent monitoring scheduling technique (AMST) algorithm in Fig 3.

Algorithm 1: Agent Monitoring Scheduling Technique
START
Input: A set of paths obtained from MRC as $P = \{P_1, P_2, \dots, P_N\}$
Compute: Find the path with shortest delay using `Agent_ADC ()`
Output: Store the set of paths with shortest Estimated delays in Queue
`Store_TDPQueue ()`
Output: Check for priority of data packets Assigned by sender and store it in the Priority Queue from highest to lowest order.
`Agent_Priority ()`
`Store_PQueue ()`
Compute: Set Receiver Buffer according to buffer estimation calculation.
`Agent_ABA ()`
Schedule: Schedule the data packets according by to highest priority packets.
`Agent_Schedule ()`
END

Fig 3 The proposed AMST algorithm.

The algorithm consists of following sub modules: i) Agent Delay Calculation using `Agent_ADC ()`, ii) Agent based scheduling with Priority using `Agent_Priority ()`, iii) Agent based Buffer blocking Avoidance using `Agent_ABA ()` and final sub module iv) Schedule the data packets using `Agent_Schedule ()`

3.1.1 Agent Delay Calculation (ADC):

In hybrid networks, scheduling packets based on minimum transmission delay improves the performance, which is important for interactive applications [12]. The survey shows one way delay estimation without clock synchronization [8] and a method to estimate clock offset. Author of [8] proposed a method which does not require symmetric or synchronization assumption, but require transmission time measurement between sender and receiver. In this paper we have proposed a packet scheduling module based on minimum transmission delay factor. Our proposed ADC sub block focuses on calculation of sender and receiver transmission times.

`Agent_ADC ()` sub function calculates link delay by using the information in the Control Packet message. The Source node sends the Control Packet message to the path₁ and path₂. The intermediate node in each path receives a Control Packet message, stores received time difference and in time field of Control Packet and forwards to its next node. This process repeated throughout the path. The delay is calculated using equation (1).

$$\Delta S = T_{STD,1} - T_{STD,2} \quad (1)$$

ΔS is sender time difference calculated from $T_{STD,1}$ is the time at which Control Packet₁ is transmitted on path₁ and $T_{STD,2}$ is the time at which Control Packet₂ is transmitted on path₂ by equation 1. Let ΔR is the receiver time difference calculated from $T_{RTD,1}$ is the time taken by Control Packet₁ to reach the receiver R on path₁ and $T_{RTD,2}$ is the time taken by Control Packet₂ to reach receiver R on path₂ shown in below equation (2).

$$\Delta R = T_{RTD,1} - T_{RTD,2} \quad (2)$$

Let ΔT be the difference in clock time between S (clk_s) and R (clk_r) and given in below equation

$$\text{i.e. } \Delta T = clk_s - clk_r, \quad \Delta T \in R \quad (3)$$

Then total delay (TD) of Control Packet₁ on path₁ is calculated as:

$$TD_{1,1} = T_{RTD,1} + T_{STD,1} + \Delta T \quad (4)$$

and similarly, the total delay (TD) of Control Packet₂ on path₂ is:

$$TD_{2,2} = T_{RTD,2} + T_{STD,2} + \Delta T \quad (5)$$

Finally when CP arrives at destination it contains the summation of all the total delay along the traversed path. The transmission delay for each path is calculated as:

$$\text{Transmission delay} = \frac{\Delta S - \Delta R}{\text{Size CP}} \times \text{Size of Data Packet} \quad (6)$$

AMST calculates the total delay for each path and stores in queue by using `Store_TDPQueue ()` function given in algorithm 1.

3.1.2 Priority of data packets:

The priority is assigned by source node, depending on how important is it for the packet to reach destination node. The source node stores the priority in each data packet's header which contains field to store priority and remains as a part of the header, not tampered, till it reaches the destination node. Once the priority is assigned by source, then `Agent_Priority ()` sub module stores the packets in priority queue.

3.1.3 Agent based buffer blocking avoidance (ABA)

To deal with out-of-order packets at receiver, `Agent_ABA ()` sub module calculates buffer size according to the equation (7). In Multipath transmissions [2] due to delay differences the receiver need to reorder the packets. Thus the size of receiver buffer should be large enough to deal with out-of-order packet and buffer blocking. The recommendation for multipath receiver's buffer size is defined in [13]:

$$\text{Buffer_size} = \sum_{i=1}^n B_{wi} \times \text{Transmission delay} \times 2 \quad (7)$$

Where $\sum_{i=1}^n B_{wi}$ is sum of the bandwidths of link i and

Transmission delay of each path from equation (6). Each sub flow will be able to send at full capacity during the time interval of the highest transmission delay among all the paths, even if a loss event occurs (multiply by 2). To utilize the capacity of all paths, a receiver must provide enough buffer space so that the sender can keep all paths fully utilized, even in the event of reordering.

3.1.4 Schedule the data packets

Once application layer transmits the data to the sender buffer. Data in sender buffer is partitioned into packets, each packet has the size equal to the MSS (Message Segment Size) of TCP paths. The Agent Scheduler chooses data packets with highest priority using `Agent_Priority ()` sub module and selects the packet with highest priority then schedule them by

assigning the path with lowest transmission delay using **Agent_ADC** () sub module according to equation (6). Once the first packet is sent and the other packets with next priority will be sent with next lowest transmission delay to the buffer. AMST aims to make packets transmitted on different path arrive in order at the receiver. The proposed scheduler uses the receive buffer more efficiently according to equation (7) than default scheduler in MRC. The Proposed scheduler can improve performance by limiting packets transmission towards path.

4. PERFORMANCE EVALUATIONS

In this section NS2 simulations of AMST algorithm is presented. For the initial simulations and the validation of the system, the following parameters have been chosen:

Parameter	Value
Number nodes	50
Simulation time	50 secs
Data packet size	1000 Bytes
MSS	512 bytes
ACK option	Selective ACK
Router Queue Size	50 packets
Receive buffer	Bandwidth x Transmission delay
Bandwidth	2Mb/s
Delay	1ms

In order to evaluate our AMST algorithm, we compare it with existing MRC and study the variation effect in the following metrics:

4.1 Average end to end delay

Average end to end delay is calculated with respect to the network load from equation (6). Average end to end delay increases as number of node increases. The AMST algorithm shows the better results, under heavy load (40 connections) compared to MRC as shown in Fig .4.

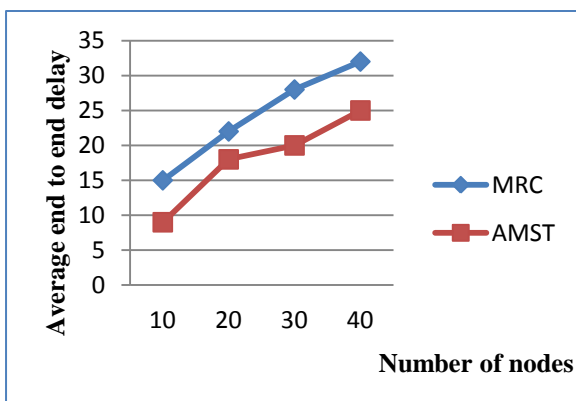


Figure 4 Average end to end delay with respect to network load.

4.2 Packet Delivery Ratio (PDR)

Packet delivery ratio can be calculated with respect to network load from equation (6). Fig 5 shows the packet delivery ratio, decreases according to connections number. When traffic load is about 40 connections (which is heavy load), the PDR achieved by AMST algorithm is better than MRC.

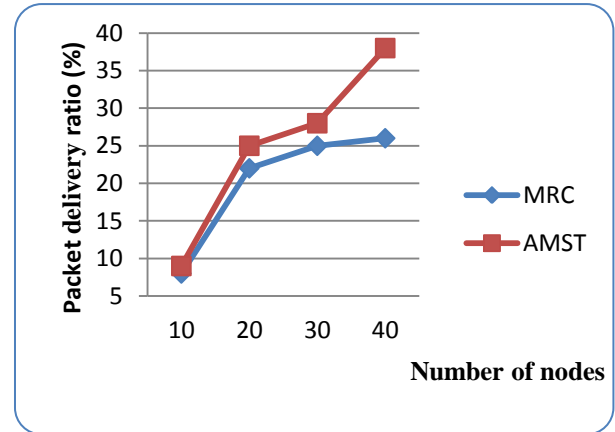


Fig 5 Packet delivery ratio with respect to network load.

5. CONCLUSION

MRC does not provide any approach, to capture Quality of Service (QoS) in the event of failure. Hence in this paper we have proposed AMST based monitoring frame work. In this work, we present a new AMST based scheduling algorithm for MRC protocol as a part of QOS module. The main contribution of this paper is to schedule the packets by assigning them to the transmission route with shortest delay based on priority and minimizing buffer blocking at the receiver buffer. Experimental work shows that our new scheduling algorithm improves network performance in terms of end to end packet delivery and average transmission delays compared to MRC algorithm.

6. REFERENCES

- [1] C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley. Improving Datacenter Performance and Robustness with Multipath TCP. in Proceedings of the ACM SIGCOMM conference, Pages 266-277, ISBN: 978-1-4503-0797-0, August 2011.
- [2] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "TCP extensions for multipath operation with multiple addresses," Internet Requests for Comments, IETF, RFC 6824, ISSN: 2070-172 Jan 2013.
- [3] X. Chen, R. Jin, K. Suh, B. Wang, and W. We Network Performance of Smart Mobile Handhelds in University Campus WiFi Network. In proceedings of ACM on Internet measurement conference ,Pages 315-328, ISBN: 978-1-4503-1705-4, 2012.
- [4] A. F. Hansen, T. Cicic, S. Gjessing, A. Kvalbein, and O. Lysne. "Multiple Routing Configurations for Fast IP Network Recovery," Networking, IEEE/ACM Transactions Volume17, pages 473-486, 2009.
- [5] S. Nelakuditi, S. Lee, Y. Yu, Z.-L. Zhang and C.-N. Chuah. "Fast local rerouting for handling transient link failures," IEEE/ACM Transactions on Networking, Volume 15(2), pages 359-372, 2007.

- [6] S. Iyer, S. Bhattacharyya, N. Taft, and C. Diot. “An approach to alleviate link overload a observed on an IP backbone.” IEEE computer and communications Volume1 pages 406–416, 2003.
- [7] M.Scharf and S.Kiesel. Head-of-line Blocking in TCP and SCTP: Analysis and Measurements. Published in Global Telecommunications Conference, GLOBECOM '06, IEEE, Pages 1 - 5, ISBN: 1-4244-0357-X, 2006.
- [8] Christoph Paasch, Simone Ferlin, Ozgu Alay, Olivier Bonaventure. Experimental Evaluation of Multipath TCP schedulers in Proceedings of the ACM SIGCOMM Pages 27-32 , ISBN: 978-1-4503-2991-0 , 201 4.
- [9] J. Iyengar, P. Amer, and R. Stewart. Concurrent Multipath Transfer using SCTP Multihoming over Independent End-to-End Paths. Published in Journal IEEE/ACM Transactions on Networking, Volume 14 , Issue 5, Pages 951-964, October 2006.
- [10] R. Seggelmann, M. Tuxen, and E.P. Rathgeb. Stream Scheduling Considerations for SCTP Published in IEEE, Software Telecommunications and Computer Networks (SoftCOM), International Conference, Pages 412 - 416, ISBN: 978-953-290-004-0, Sept. 2010.
- [11] G. Sarwar, R. Boreli, E. Lochin, A. Mifdaoui, and G. Smith. Mitigating Receiver’s Buffer Blocking by Delay Aware Packet Scheduling in Multipath Data Transfer. Published in IEEE, Advanced Information Networking and Applications Workshops (WAINA), International Conference, Pages 1119 - 1124, ISBN: 978-0-7695-4952-1, March 2013.
- [12] C. Raiciu, C. Paasch, S. Barre, A. Ford, M. Honda, F. Duchene, O. Bonaventure, and M. Handley. How Hard Can It Be? Designing and Implementing a Deployable Multipath TCP. In 9th USENIX Symposium on Networked Systems Design and Implementation, April 2012.
- [13] A. Ford, C. Raiciu, M. Handley, S. Barre, and J. Iyengar, “Architectural guidelines for multipath TCP development,” Internet Requests for Comments, IETF, RFC61822011.