IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Vol.-6, Issue-9, Sep 2018 E-ISSN: 2347-2693

Adapting TFMCC Protocol in PIM-DM for avoiding Congestion Control in Wired Multicast

V B Deepa[1*], M B Ushadevi[2]

[1]Information Science and Engineering,JNN College of Engineering,Shimoga
[2]Telecommunication and Engineering, JNN College of Engineering,Shimoga

Available online at: www.ijcseonline.org

Accepted: 17/Sept/2018, Published: 30/Sept/2018

Abstract— PIM-DM is a multicast routing protocol that uses the underlying unicast routing information base to flood multicast datagrams to all multicast routers. End-to-End Multicast Congestion Control (MCC) is a complex problem. TFMCC (TCP Friendly Multicast Congestion Control) is a congestion control mechanism for multicast transmissions. Where the sending rate is adapted to the receiver experiencing the worst network conditions TFMCC shows better performance. TFMCC is stable and responsive under a wide range of network conditions and scales to receiver sets on the order of several thousand receivers. In this paper, we implemented TFMCC protocol to PIM-DM model and also we compare TFMCC with TCP behaviour. TFMCC is designed to be reasonably fair when competing for bandwidth, sending rate, varying number of links, different receiver capacity and scalability. Experimental results show tremendous performance improvement in throughput without affecting the TCP fairness of the protocol.

Keywords— PIM-DM,TFMCC, Congestion control, Scalability

I. INTRODUCTION

Multicasting refers to a communication technique in which a stream of data is transmitted from a single sender or source to multiple destinations on a network, eliminating the need for the data to be sent individually from the source to each 'interested' receiver. The increasing popularity of group communication applications such as multi-party teleconferencing tools and information dissemination services motivated the development of several multicast transport protocols layered on top of IP multicast for efficient multipoint data distribution. The precise requirements for multicast congestion control are perhaps open to discussion given the efficiency savings of multicast, but we take the conservative position that a multicast flow is acceptable if it achieves no greater medium-term throughput to any receiver in the multicast group than would be achieved by a TCP flow between the multicast sender and that receiver. Such a requirement can be satisfied either by a single multicast group if the sender transmits at a rate dictated by the slowest receiver in the group, or by a layered multicast scheme that allows different receivers to receive different numbers of layers at dif Congestion control for multicast transmission of multimedia data is a challenging research area[1]. In any proposed solution, one has to find the balance between the attributes of multimedia applications (bandwidth consuming applications, tolerant to packet losses, sensitive to delays) and the need for TCP-friendly behaviour. Although, congestion control for multimedia data transmission involves various contradictory requirements, we believe that at least the three following requirements should be satisfied:

- Any proposed congestion control should prevent oscillations, as much as possible, in order to minimize the Audio-Video (AV) encoding and decoding distortion**.**
- Inter-arrival jitter delay should be small in order to meet the multimedia application's requirements.
- Packet losses should be minimized and when exist they should have minimal negative results in end user's perception.

The rest of this paper is organized as follows. Our proposed nd framework and protocol are presented in section 2. Simulation results are presented in section 3. Conclusions and future work are discussed in section 4.

II. RELATED WORK

Up to now there are promising approaches in the field of the single layer multicast congestion controls in bibliography. TFMCC[2], extends the basic mechanisms of TFRC[3] to support single layer multicast congestion control. The most important attribute of TFMCC is the suppression of feedback receiver reports. TFMCC is using the receiver with the lowest receiving capacity to act as the representative of the multicast group. PGMCC[4] uses a window-based TCP controller based on positive ACKs between the sender and the group representative. TBRCA[5] targets at maximizing the overall amount of multimedia data to the whole set of receivers. With the use of a bandwidth rate control algorithm it dynamically controls the output rate of the video coder. LDA+ [6] employs a TCP equation based congestion control for measuring the TCP friendly bandwidth share in the event of packet losses. Intention is to use this congestion control as the rate control protocol in their proposed framework for PIM-DM Model. They restrict their evaluation in this work only in the wired portion of the network, as the protocol has to be modified and enhanced in order to support wireless receivers.

III. PIM-DM

Protocol Independent Multicast-Dense Mode PIM-DM[7] is a source-specific tree routing protocol that characterized RPF and pruning and grafting schemes for multicasting. Unicast protocol can be a link state protocol, or distance vector protocol. PIM-DM will be deployed in LAN where group membership is comparatively dense and bandwidth is easily obtainable. PIM-DM protocol acts in two phases shown in figure 2.1: In the first phase, the entire network is flooded with multicast data and this is done by propagation of packet on all interfaces exclude on the upstream interface. Because of its network overflowing technique, this phase is extremely ineffective because it directs to extreme network resource usage. In the second phase, called a prune phase, Prune message are cuts out , in unneeded branches by means of a network machine, after a reception of a Prune packet, stops further forwarding of multicast traffic on this interface and the interface is set to be in prune state. Hello packets are periodically exchanged between PIM-DM routers. The presence of PIM-DM capable to know the neighbour routers in the network aids the routers. When a source starts sending in PIM-DM, all downstream systems want to receive multicast datagram. At staring period, all multicast datagram are flooded to all areas of the network. PIM-DM uses RPF method to prevent looping of multicast datagram while flooding. Prune state instantiating if some areas of the network do not have group members, PIM-DM will prune off the forwarding branch by Prune state has a finite lifetime. If the lifetime expires, data will again be forwarded down the previously pruned branch. Prune state is known with an (S,G) pair. When a new member for a group G enters in a pruned area, a router can "graft" toward the source S for the group, thereby turning the pruned branch back into a forwarding branch.

PIM-DM uses a state refresh message, to minimize repeated flooding of datagram and subsequent pruning associated with a particular (S,G) pair. The router(s) sent this message directly connected to the source and is propagated throughout the network. The state refresh message causes an existing prune state to be refreshed, when received by a router on its RPF interface. PIM-DM has a simplified design and is not hard-wired into a specific topology discovery protocol, comparing with multicast routing protocols with built-in topology discovery mechanisms like in DVMRP. Since sufficient topology information were available ,this simplification does incur more overhead by causing flooding and pruning to occur on some links that could be avoided, that is to decide whether an interface leads to any downstream members of a particular group. In favour of the simplification and flexibility gained by not depending on a specific topology discovery protocol, but considering additional overhead is to be chosen .

Source host sent a multicast datagram. It creates an (S,G) entry, if a receiving router has no forwarding cache state for the source sending to group G. The RPF lookup in the unicast routing table is determined by incoming interface for (S,G). The (S,G) outgoing interface list contains dense-mode configured interfaces, that have PIM routers present or host members for group G. A PIM-Prune message is triggered when an (S,G) entry is built with an empty outgoing interface list is as shown in fig(3.1).

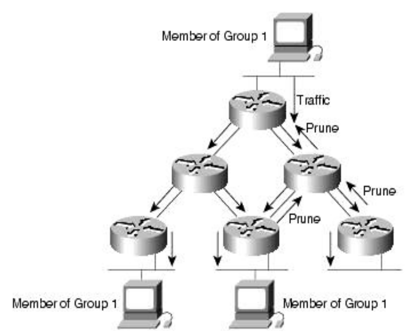


Figure 3.1 PIM-DM

This type of entry is called a negative cache entry. This can occur when a leaf router has no local members for group G or a prune message was received from a downstream router which causes the outgoing interface list to become NULL. PIM-Prune messages are never sent on LANs in response to a received multicast packet that is associated with a negative cache entry.

PIM-Prune messages received on a point to point link are not delayed before processing as they are in the LAN procedure. If the prune is received on an interface that is in the outgoing interface list, it is deleted immediately. Otherwise the prune is ignored. When a multicast datagram is received on the incorrect LAN interface (i.e. not the RPF interface) the packet is silently discarded. If it is received on an incorrect point-to-point interface, Prunes may be sent in a rate-limited fashion. Prunes may also be rate-limited on point-to-point interfaces when a multicast datagram is received for a negative cache entry.

IV. TFMCC (TCP FRIENDLY MULTICAST CONGESTION CONTROL)

TFMCC is intended to be a congestion control scheme that can be used in a complete protocol for reliable content delivery and streaming of multimedia information. TFMCC is most applicable for sessions where to deliver a substantial amount of data (i.e., in length from hundreds of kilobytes to many gigabytes) and whose duration is on the order of tens of seconds or more. TFMCC is intended for multicast delivery as shown in figure(4.1). There are currently two models of multicast delivery, the Any-Source Multicast (ASM) model and the Source-Specific Multicast (SSM) model. TFMCC works with both multicast models, but in a slightly different way. ASM is used, where feedback from the receivers is multicast to the sender, as well as to all other receivers. Feedback can be received, either from multicast on the same group address used for sending data or on a separate multicast feedback group address. This is similar to PIM-SM model. For SSM, the receivers must unicast the feedback directly to the sender. Feedback from a receiver will not be received by other receivers. This is similar to PIM-DM model. All types of networks that allow bi-directional communication, including LANs, WANs, Intranets, the Internet, asymmetric networks, wireless networks, and satellite networks TFMCC works with inherently. In some network environments varying the sending rate to the receivers may not be advantageous (e.g., for a satellite or wireless network, there may be no mechanism for receivers to effectively reduce their reception rate since there may be a fixed transmission rate allocated to the session).

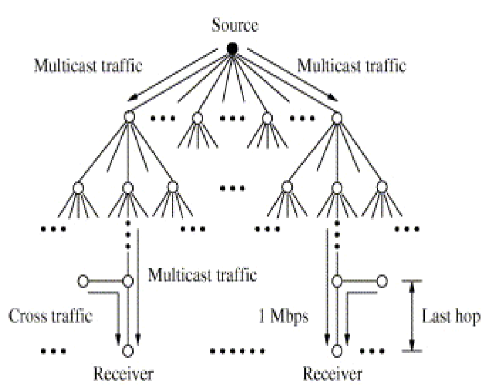


Figure 4.1 TFMCC Model

The TFMCC Protocol

Building an equation-based multicast congestion control mechanism requires that the following problems be solved:

- A control equation must be chosen that defines the target throughput in terms of measurable parameters, in this case loss event rate and RTT.
- The loss event rate measured by each receiver. Thus a filter for the packet loss history needs to be chosen that is a good stable measure of the current network conditions, but is sufficiently responsive when those conditions change.
- The RTT is measured by each receiver to estimate time to the sender. Designing a way to do this without causing excessive network traffic is a key challenge.
- Each receiver uses the control equation to calculate an acceptable sending rate from the sender to itself.
- A feedback scheme must be so devised that feedback from the receiver calculating the slowest transmission rate always reaches the sender, but feedback implosions do not occur when network conditions change.
- A filtering algorithm needs to be devised for the sender to determine which feedback it should take into account as it adjusts the transmission rate.

All these parts are closely coupled. For example, altering the feedback suppression mechanisms will impact how the sender deals with this feedback. Many of our design choices are heavily influenced by TFRC, as these mechanisms are fairly well understood and tested. In this paper we will expend most of our efforts focusing on those parts of TFMCC that differ from TCP.

DETERMINING AN ACCEPTABLE SENDING RATE

TFRC and TFMCC is derived from a model for long-term TCP throughput in bytes/sec by the control equation [8].

$$Ttcp = S/(tRTT\sqrt{}(2p/3 + (12\sqrt{}\ 3p/8)p(1 + 32p\char`^2))) \qquad \text{--------1}$$

The expected throughput T_{tcp} of a TCP flow is calculated as a function of the steady-state loss event rate *p* the round-trip time t_{RTT} and the packet size *s*. TFMCC receiver measures its own loss event rate and estimates its RTT to the sender. It then uses Equation (1) to calculate T_{tcp}, which is an estimate of the throughput a TCP flow would achieve on the network path to that receiver under the same network conditions. If the sender does not exceed this rate *for any receiver* then it should be TCP-friendly, in that it does not affect a TCP flow through the same bottlenecks more than another TCP flow would do.

ADJUSTING THE SENDING RATE

The receiver will continuously send feedback to the sender. If a sender receives feedback that indicates a rate that is lower than the sender's current rate, the sender will immediately reduce its rate to that in the feedback message. In order to reduce a large number of unnecessary messages, receivers will not send feedback unless their calculated rate is less than the current sending rate. There exists a problem how do we increase the transmission rate?. To increase the transmission rate in the absence of feedback not affordable, as the feedback path from the slowest receiver may be congested. As a solution the concept of the *current limiting receiver* (CLR) exists. The CLR is the receiver that the sender believes currently has the lowest expected throughput of the group. The sender can use the CLR's feedback to to increase the transmission rate. The CLR will change if another receiver sends feedback indicating that a lower transmission rate is required. It will also change if the CLR leaves the multicast group – this is normally signalled by the CLR, but an additional timeout mechanism serves as a backup in case the CLR crashes or becomes unreachable. Normally the way loss measurement is performed limits the possible rate increase to roughly 0.3 packets per RTT, as shown in [12]. However, if the CLR leaves the group, the new CLR may have a significantly higher calculated rate. We cannot afford to increase directly to this rate, as the loss rate currently measured may not be a predictor of the loss rate at the new transmission rate. Instead we then impose a rate increase limit of one packet per RTT, which is the same as TCP's additive increase constant, so that the rate gradually increases to the new CLR's rate.

MEASURING THE LOSS EVENT RATE

The loss event rate can measured at the receivers only for the scalably. The measurement mechanism closely matches that used for TCP. A receiver aggregates the packet losses into *loss events*, defined as one or more packets lost during a round-trip time. The number of packets between consecutive loss events is called a *loss interval*. The average loss interval size can be computed as the weighted average of the m most recent loss intervals.

$$l_{avg}(k)=\frac{\sum_{i=0}^{m-1} wilk-1}{\sum_{i=0}^{m-1} wi}$$

The weights w_i are chosen so that very recent loss intervals receive the same high weights, while the weights gradually decrease to 0 for older loss intervals. For example, with eight weights we might use {5, 5, 5, 5, 4, 3, 2, 1}. This allows for smooth changes in l_{avg} as loss events age. While large values for m improve the smoothness of the estimate, a very long loss history also reduces the responsiveness and thus the fairness of the protocol. Values around 8 to 32 appear to be a good compromise.

The loss event rate *p* used as an input for the TCP model is defined as the inverse of $_{lavg}$.. The interval since the most recent loss event does not end with a loss event and thus may not reflect the loss event rate. This interval is included in the calculation of the loss event rate if doing so reduces *p*.

$$P = \frac{1}{\max\,(lavg\,(k),lavg\,(k-1))}$$

For a more thorough discussion of this loss measurement mechanism see [8].

ROUND-TRIP TIME MEASUREMENTS

TFMCC is for each receiver to be able to measure its RTT to the sender without causing excessive traffic at the sender. The problem is primarily one of getting an initial RTT measurement as, with the use of timestamps in the data packets, a receiver can see changes in the delay of the forward path simply from the packet's arrival time. A receiver is to be able to initialize its RTT measurement without having to exchange any feedback packets with the sender. This is possible if the sender and receiver have synchronized clocks, which might be achieved using GPS receivers. Less accurately, it can also be done using clocks synchronized with NTP[9]. In another case, the data packets are time stamped by the sender, and the receiver can then compute the one-way delay. The RTT is estimated to be twice the one-way delay. In the case of NTP, the errors that accumulate between the stratum-1 server and the local host must be taken into account. An NTP server knows the RTT and dispersion to the stratum-1 server to which it is synchronized. The sum of these gives the worst case $\dot{\varepsilon}$ in synchronization. To be conservative:

$$t_{RTT}=2(d_{s\text{->}R}+\dot{\varepsilon}_{sender}+\dot{\varepsilon}_{receiver})$$

Each receiver must then initialize its RTT estimate to a value that should be lager than the highest RTT of any receiver.

RTT MEASUREMENT

A receiver gets to measure the instantaneous RTT by sending times stamped feedback to the sender, which then echoes the timestamp and receiver ID in the header of a data packet. If more feedback messages arrive than data packets are sent, Priority is given to the sender's report echoes in the following order:

1. A receiver whose report causes it to be selected as the newCLR.
2. Receivers that have not yet measured their RTT.
3. Non-CLR receivers with previous RTT measurements.
4. The existing CLR.

Ties are broken in favour of the receiver with the lowest reported rate. Normally the number of data packets is larger than the number of feedback packets, so the CLR's last report is echoed in any remaining data packets. To prevent a single spurious RTT value from having an excessive effect on the sending rate we smooth the values using an exponentially weighted moving average (EWMA).

$$t_{RTT}=\beta.t_{RTT}^{inst}+(1-\beta).t_{RTT}$$

For the CLR we set β_{CLR}=0.05. Given that other receivers will not get very frequent RTT measurements and thus old measurements are likely to be outdated, a higher value of β non-CLR=0.05 used for them.

V. RESULTS AND DISCUSSIONS

Simulations were carried out using Network Simulation (ns-2.35). We patch new agent TFMCC algorithm in transport layer ns-allinone-2.35. Varying links and packet size and bandwidth calculated throughput with comparing TFMCC and TCP agents. Comparison of TFMCC and TCP flows of multicast routing strategies is as follows:

5.1 Throughput: It is the ratio of packets delivered successfully from source to destination per unit time. TFMCC has higher throughput value as compared to TCP in PIM-DM modes in varying bandwidth as shown in fig(5.1).

Figure 5.1 Throughput TFMCC and TCP

5.2 Packet Delivery Ratio

It is the ratio of number of received data packets to the number of generating data packets TFMCC shows better performance than. TCP. Packet Sent by TCP and TFMCC varying number of links as shown fig(5.2) and fig(5.3).

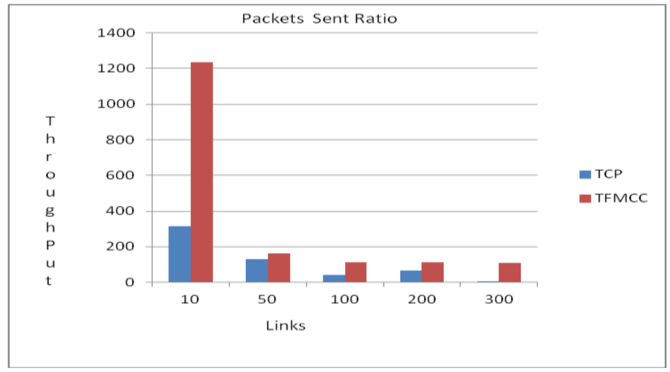


Figure 5.2 Packet Sent by TCP and TFMCC

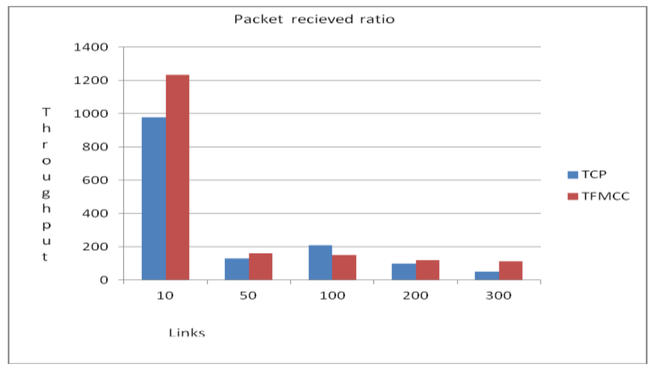


Figure 5.3 Packet Received by TCP and TFMCC

Throughput achieved by varying packet size, TFMCC shows better performance than TCP as shown in fig(5.4).

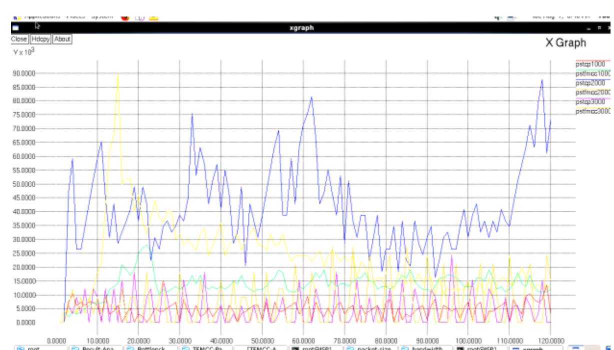
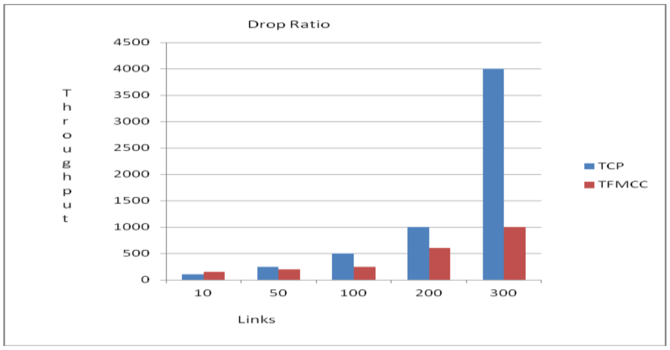


Figure 5.4 Throughput varying Packet Size

5.3 Drop Ratio: It is the ratio of packets lost to the packets sent. TFMCC has better (lower) drop ratio as compared to TCP in PIM-DM Mode of Multicast Routing Strategies as shown fig(5.5).



Figure(5.5) Drop Ratio TFMCC vs TCP

The average packet loss ratio between TFMCC and TCP is evaluated. It is observed that the average packet loss ratio of TFMCC is 0.52% and the average packet loss ratio of TCP is 0.59% in varying number of links. So the average packet loss ratio of TFMCC has improved by about 11.9% at the same network circumstance. This is because TFMCC can respond to the network congestion conditions and adjust its sending rate in time by using the average delay before packet loss occurs. Comparison of TCP and TFMCC is shown in table 1

	TCP	TFMCC
TCP Friendliness	Good	Average
Stable transmission Rate	Good	Very Good
Convergence Time	Average	Very Good
Scalability	Average	Very Good

VI. CONCLUSION

TFMCC, a single-rate multicast congestion control mechanism intended to scale to groups of several thousand receivers. Performing multicast congestion control whilst remaining TCP-friendly is difficult, in particular because TCP's transmission rate depends on the RTT, and measuring RTT in a scalable manner is a hard problem. Given the limitations of end-to-end protocols, we believe that TFMCC represents a significant improvement over previous work in this area. The implication is therefore that single-rate multicast congestion control mechanisms like TFMCC are only really well-suited to relatively long-lived data streams. Fortunately it also appears that most current multicast applications such as stock-price tickers or video streaming involve just such long-lived data-streams. Simulation results show a remarkable improvement in throughput performance, while maintaining the TCP- fairness property of TFMCC, an essential property that must be possessed by any multicast protocol in order to fairly coexist in the Internet with TCP-flows.

REFERENCES

[1] Mohammed A. Al-Jendary1, Maher A. Al-Sanabani2 "Congestion Avoidance in Wireless Mesh Networks Based On Multicast Routing Protocols" IJCSE vol-6 Issue-8 Aug 2018 E-ISSN: 2347-2693

[2] Christos Bouras1, postolos Gkamas1 and Georgios Kioumourtzis2. " Smooth Multicast Congestion Control for Adaptive Multimedia Transmission" IEEE Infocom 2003.Twenty-second Annual Joint Confenence Society (IEEE Cat.No.03CH37428)

[3] "TCP Friendly Rate Control (TFRC): Protocol Specification" S. Floyd M. Handley University College of London J. Padhye Microsoft J. Wider DoCoMo September 2008.

[4] S. Floyd, M. Handley, J. Padhye, and J. Widmer. "Equation-Based Congestion Control for Unicast " Pages 49-60 ACM SIGCOMM Computer Communication Aug 2005

[5] Smith, H., Mutka, M., Rover, D. A "Feedback based Rate control Algorithm for Multicast Transmitted Video Conferencing," Accepted for publication in the Journal of High Speed Networks,vol 7 no 3,4,pp 259-279,1998

[6] Sisalem D., Wolisz A. "LDA + TCP - Friendly Adaptation: A Measurement and Comparison Study," in the 10th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'2000), June 25 - 28,2000, Chapel Hill,NC, USA.

[7] Protocol Independent Multicast (PIM), Dense Mode Protocol. Specification. Deborah Estrin. omputer science Department/ISI University of Southern california. Los Angeles, CA 90089 estrin@usc.edu. Dino Farinacci. Cisco Systems Inc. 170 West Tasman Drive,. San Jose, CA 95134 dino@cisco.com

[8] JOR¨ G WIDMER AND MARK HANDLEY "EXTENDING EQUATION-BASED CONGESTION CONTROL TO MULTICAST APPLICATIONS" UNIVERSITAT¨ MANNHEIM PRAKTISCHE INFORMATIK IV L15, 16 D-68131 MANNHEIM.

[9] D.L.Mills,A.Thyagarajan,and B.C Huffman. "Internet Timekeeping around the globe". *Proc Precision Time and Time Inerval(PTTI)* Application and Planning Meeting, pages 365-371,Dec 1997.

Authors Profile

Mrs Deepa V B working as Assistant Professor in Department of Information Science and Engineering at JawaharaLal Nehru National College of Engineering Shimoga Karnataka. Perusing Ph.D in Computer Network in the area of Multicast Congestion Control.

Dr Ushadevi M B working as Prof and Head of Dept of Telecommunication Engineering at JawaharLal Nehru National College of Engineering Shimlogga Karnataka. She completed Ph.D Degree in 2010 at Kuvempu University Shimogga. Her research area is Switching Networks, interested areas are Wireless Sensor Networks, Mobile Networks. She published many conference ,journal papers.