EFFICIENT ADAPTATION OF FUZZY CONTROLLER FOR SMOOTH SENDING RATE TO AVOID CONGESTION IN MULTICAST NETWORKS

Deepa V B [1] and Ushadevi M B[2]

[1]Department of Computer Engineering, Information Science and Engineering, JawaharaLal Nehru National College of Engineering.Shimogga, Karnataka, India
[2]Department of Telecommunication and Engineering, JawaharaLal Nehru National College of Engineering.Shimogga, Karnataka, India

ABSTRACT

This paper prefers a fuzzy-logic-based sending rate adaption scheme named FSR(Fuzzy Sending Rate) intending to improve the evenness of TCPFriendly Multicast Congestion Control (TFMCC). To mitigate fluctuation of sending rate for TFMCC sender, FSR intends, five actions and link utilization for tuning sending rate and uses a fuzzy controller to determine which operation should be reaped according to the feedback information from CLR (current limiting receiver). Asymmetrical membership functions and biased fuzzy inference rules make FSR as friendly to TCP flows as TFMCC. Simulation results show that FSR has exceptional smoothness and fine TCP Friendliness.

KEYWORDS

Tfmcc, Rtt, Link Utilization, Sending Rate, Fuzzy Controller

1. INTRODUCTION

Multicast Congestion Control (MCC) is one of the critical methods to weave congestion and make network perform steadily. It is needed that MCC mechanism must not only assurance the QoS (Quality of Service) of users, but also ensure that multicast flows could share the resource adequately with existing flows specifically TCP flows, which is called TCP-Friendliness[1].

MCC functioning can be divided into two division with relating to the manner of sending rate[1]: single-rate mechanism versus multi-rate mechanism. Although single-rate MCC is poor in performance and expandability, it is easy to achieve, has fine friendliness and suits to the conditions that is not so heterogonous. Some newly proposed multi-rate MCC which are also called hybrid MCC[2-4] make single-rate MCC as building block that each layer applies single-rate MCC.

One of the utilizations that IP multicast transmits most is multimedia applications (video or voice) which have smooth transmitting rate. Repeated changes in the transmission rate may disintegrate the quality of multimedia and import more difficulty and complicatedness to encoder/decoder. How to sustain smooth transmitting rate is a technology obstacle associated with all MCC mechanisms. In this paper, we prefer a sending rate adaption scheme based on Fuzzy-logic intended to smoothen the sending rate of TFMCC 5].The rest of this paper is categorized as follows. Section 2 summarizes related work. Section 3 interprets the excessive decrease

DOI: 10.5121/ijcsit.2018.10606

phenomenon that make sending rate of TFMCC fluctuate and presents Fuzzy-logic based rate adaption scheme. Section 4 elaborates the link Utilization of the network to decrease congestion, Section 5 introduces Fuzzy Controller for multicast congestion control having smooth sending Rate. Section 6 gives results and discussion as well as section 7 conclusion.

2. Related Work

There are different mechanisms have been adopted till now to control the congestion in the network. We know that TFMCC [2] is a steady state equation based multi-cast technique to calculate the throughput of the network. But it has some problems. First, it is slow in identifying the Congestion representative and therefore it is slow in reacting to changes in the congestion condition. Secondly, the CLR drag down the whole TFMCC session. Therefore, some modifications are made to TFMCC using Additive Increase Multiplicative Decrease (AIMD)[7-8]. Once congestion is detected, it is notified by using Implicit Congestion Notification (ICN) signaling. After receiving this signal the intermediate nodes adjust its sending rate so that congestion doesn't occur. Further congestion is implemented using Fuzzy logic controller [9,10].

2.1 Proposed Work

We have seen that many algorithms have been proposed to control the congestion in the multicast network. These algorithms used different protocols to reduce congestion by adjusting the sending rate of the sender and different mechanisms has been proposed to indicate the congestion representative. The heterogeneous behavior of the network leads to the more utilization of bandwidth which results in congestion in the network. We propose an algorithm to improve the utilization by keeping the same sending rate while congestion occurs in the network. And also we adapt efficient fuzzy controller for sending smooth rate to avoid congestion using parameter RTT and Link Utilization as input and Sending Rate as output for Fuzzy Controller.

3. Adaption Scheme For Sending Rate Using Fuzzy (Fsr)

TFMCC has high quality performance including good TCPFriendliness and efficient feedback suppression mechanisms[3].In that idea TFMCC is extensively accepted and preferred as building block in some multi-rate MCC[2]. To be TCP-Friendly, each TFMCC receiver estimates its expected sending rate by a control equation derived from a model of TCP's long-term throughput[6]:

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

where *RTT* is Round-Trip time, *s* is packet size and *p* is packet loss event rate. TFMCC sender regulates sending rate by the predicted throughput of CLR(current limiting receiver), which is the receiver who has the minimum expected throughput of the group. Once the expected throughput of CLR *T*(*k*) is lesser than the current sending rate *S*(*k*) , sender will adopt the new sending rate to the expected rate. Actually, this operation may decrease the sending rate enormously and indirectly make throughput change, which is adverse to undergo for multimedia applications. In the intensification following, *p*(*v*) and *RTT*(*v*) are the loss event rate and RTT correspondingly, at the sending rate with *v* ;*T*(*p*, *RTT*) is the predictable throughput estimated by (1) with the parameters *p* and *RTT* .We consider two consecutive steady states: state1 and state2.Basically, the network is in state1 and at time t_1 the backdrop traffic increases, which means that congestion is more dangerous. In state1 TFMCC has sending rate of $S_1 = v_1$, loss rate of p_1 and RTT of RTT_1 ; correspondingly state2 with $S_2 = v_2 < v$, p_2 and RTT_2 . During each steady state, the three variables fulfill (1). Let p' (v_1) denote the loss rate estimated by the CLR after t_1 but before t_2 , during when the congestion has been more serious but the feedback has not received by the sender, so the sending rate is still v_{1}.. We also further assume that the background traffic doesn't change after t_1. It is obvious that the higher sending rate is, the larger loss rate and RTT will be if the background traffic is steady. So we get

$$\begin{cases} P' > (V_1)\ P(V_2) \\ RTT'(V_1) > RTT'(V_2) \end{cases} \tag{2}$$

At least one of the two inequations above is absolute because of $v_1 > v_2$. Then the calculated predicted throughput is,

$$T(p'(v_1), RTT'(v_1)) < T(p(v_2), RTT'(v_1)) = v_2 \tag{3}$$

Once the feedback with as $T(p'(v_1), RTT'(v_1))$ referred to as T'_1 for short, reaches the sender, the sender decreases the sending rate directly to T'_1,

$$S' = T' < v_2 \tag{4}$$

Afterward, the CLR will estimate new loss rate and *RTT* at the sending rate of S'2< v2:

$$\begin{gathered} p(S'_{2)} \leq p(v_2) \leq p'(v_1) \\ RTT(S'_2) \leq p'(v_1) \leq RTT'(v_1) \end{gathered} \tag{5}$$

Then the new calculated expected throughput will be higher than $S'_{'□}$. When the new feedback arrive at the sender, the sender will increase the sending rate additively. Finally with several step of adjusting, the sending rate will be close to v_2 .

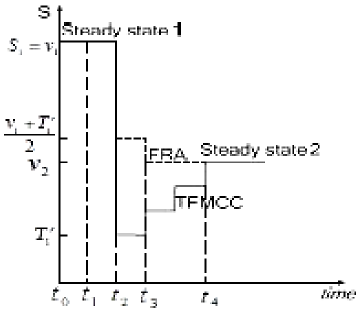


Figure 3.1 Excessive decrease phenomena in TFMCC and Multiplicative Decrease action in FSR.

Now, we can analyze that TFMCC will decrease the sending rate excessively and it will take a long time to converge to the new state as depicted in Figure(3.1). This conclusion will be proved by simulations in section 5.

3.1 Rate Adjusting Actions

n order to alleviate the excessive decrease phenomenon in TFMCC, we introduce five rate adjusting actions into FSR for different congestion degree.

• **Additive Increase (AI):** The additive increase component should be such that at no instant of time should the sending rate undergo an increment greater than about 10% the current size. This serves to distinguish an additive increase form a multiplicative increase. To ensure TCP-Friendliness, AI action is taken when the expected rate is a little higher than sending rate.

• **Additive Decrease (AD**): The additive decrease component should be such that the decrement in the sending rate should never be more 10%. This serves to differentiate it from a multiplicative decrease. Actual, if the expected sending rate is a little lower than the current sending rate, which is meaning congestion is not very serious, there is no need to decrease sending sharply, especially for multimedia applications. In such case, in order to smoothen sending rate we introduce "additive decrease (AD)" action that the sending rate will decrease by one packet per RTT.

• **Multiplicative Increase (MI**): The multiplicative increase component should be large enough so that the increment in size of the sending rate is larger than 10% always. To fully utilize resource and improve response speed, we introduce "MI" into FSR: if the expected T'_1 is much larger than the sending rate, the new sending rate will increase to $S_1 +T'_1/2$.

• **Multiplicative Decrease (MD):** The multiplicative decrease component should be so chosen that the sending rate decrement is never less than 10%. From (3) we can see that the actual rate the receiver can accept is between the two values: $T'_1< v_2 < S_1$ .In FSR, the sender decreases the sending rate to $S_1 + T'_1/2$ instead of T'_1 directly, which is called "MD" action:

3.2 Link Utilization

We are proposing an algorithm to improve the utilization by keeping the same sending rate while congestion occurs in the network. For this we have to first calculate the link utilization using old link utilization method. The proposed algorithm entitles LUMCC is given below[11]:

Algorithm:

Link Utilization Based Multicast Congestion Control (LUMCC)
1) Initialize the total link capacity.
2) Initialize the initial sending rate.
3) Initialize the queue size.
4) Initialize the packet size.
5) Set the session time.
6) Calculate the packet loss ratio on the link.

$$P_{ls=}P_d/P_d+P_s$$

Where P_{ls} is the Packet loss observed on the link, Pd is the number of Packets dropped, Ps is the number of Packets sent on the link.

7) Calculate the link utilization, αij.
Where $\sum bw_{f*} \max(X_{if})^f$ denotes the value of total traffic demand for all flows fЄF that are transmitted through link (i, j), Cij is the link capacity.

8) Setting the threshold values:
a) If (99% of link utilization < αij)
Then, Congestion is very high and we adjust

$$F_{new}= \alpha_{ij}/\beta^{*}2$$

b) If (99% ≤ αij < 90%) Then, Congestion is high and we adjust
$F_{new}= \alpha_{ij}{}^{*}/ 2^{\beta}/2$
If (F_{new} > 99%) Then go to step "a".

c) If (90% ≤ αij ≤ 50%) Then,\

$F_{new}= \alpha_{ij}{}^{*}\ \beta^{*}2$
Congestion is medium and we adjust
If ($F_{new} > 90\%$) Then go to step "b".

d) If ($0 < \alpha ij < 49\%$) Then, Congestion is low and we adjust
$F_{new}= \alpha_{ij}{}^{*}\ \log_2\alpha$
Else increment Fnew till its value reaches to medium value.
We see the example of propose algorithm given as below:

Example: Suppose the Link Capacity is 100 Mbps, Initial Sending Rate is 80 Mbps, Packet size is 300, RTT is 150 ms, decreasing factor, β is 0.65 ($0 < \beta < 1$), increasing factor α = S/RTT is 2 and we vary the Queue size.

Case 1: Queue size = 50 packets
Link utilization,
$\alpha = \sum bw_{f*}\ \max(X_{if})^{f}\ /C_{ij}\ f\text{€}F$
=80*50/100
=40%

the link utilization is 40% means that congestion is low. Then we use third condition and the proposed formula is:
$F_{new}= \alpha_{ij}{}^{*}\ \log_2\alpha$
$=40*\log_2(S/RTT)=40*\log_2(2)=40\%$

Again, the Fnew is 40%, then we go to step "c".
So, our utilization comes to 65%.

Case 2: Queue size = 80 packets
Link utilization
$\alpha = \sum bw_{f*}\ \max(X_{if})^{f}\ /C_{ij}$ for all $f\text{€}F$
=80*80/100=64%

The link utilization is 64% means that congestion is medium. Then we use second condition and the proposed formula is:
$F_{new}= \alpha_{ij}{}^{*}\ \beta^{*}2$
=64*0.65*2
=83%

So, our utilization comes to 83%.

Case 3: Queue size = 120 packets
Link utilization,
$\alpha = \sum bw_{f*}\ \max(X_{if})^{f}\ /C_{ij}$ for all $f\text{€}F$
=80*120/100=96%

As utilization is 96% which shows the high congestion in the network is according to set thresholds. So, we have made the congestion medium. For we use first condition and the proposed formula is:
$F_{new}= \alpha_{ij}\ */\ 2^{\beta}/2$
$96*2^{0.65}*2=5\%$

We conclude that if our link utilization is high then we need more care about the congestion, otherwise regularly needs to increase the flow speed according to low and medium include with medium and high factor of speed respectively. Therefore, utilization of link is very important phenomenon for control the congestion.

4. FUZZY INFERENCE ENGINE

Fuzzy Inference System is the key unit of a fuzzy logic system having decision making as its primary work as shown in figure (4.1). It uses the “IF…THEN” rules along with connectors “OR” or “AND” for drawing essential decision rules[12].

Following are some characteristics of FIS

- The output from FIS is always a fuzzy set irrespective of its input which can be fuzzy or crisp.
- It is necessary to have fuzzy output when it is used as a controller.
- A defuzzification unit would be there with FIS to convert fuzzy variables into crisp variables.

The following five functional blocks describe the construction of FIS

- **Rule Base –** It contains fuzzy IF-THEN rules.
- **Database –** It defines the membership functions of fuzzy sets used in fuzzy rules.
- **Decision-making Unit –** It performs operation on rules.
- **Fuzzification Interface Unit –** It converts the crisp quantities into fuzzy quantities.
- **Defuzzification Interface Unit –** It converts the fuzzy quantities into crisp quantities. Following is a block diagram of fuzzy interference system.

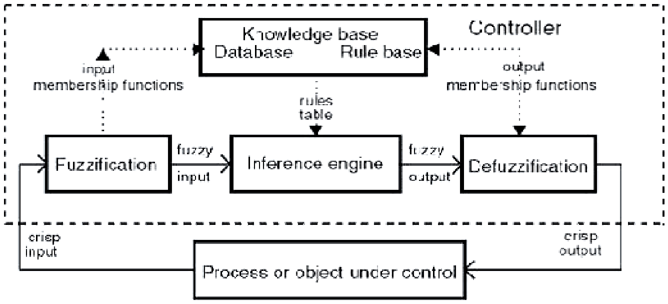


Figure 4.1 Fuzzy Inference Engine

4.1.1 Types of Controller

- Self-organising controller – a selfish creation noticing nothing outside itself and always observing itself only and nothing else.
- Adaptive controller – a system that is just a current situation with neither memory nor recollections about the past and reflections about the future.
- Learning controller – an industrious student constantly developing and expanding his or her knowledge and experience.

4.2 Mamdani Fuzzy Logic Controller

The most commonly used fuzzy inference technique is the so called Mamdani method (Mamdani & Assilian, 1975) which was proposed[12], by Mamdani and Assilian, as the very first attempt to control a steam engine and boiler combination by synthesizing a set of linguistic control rules obtained from experienced human operators. Their work was inspired by an equally influential publication by Zadeh (Zadeh, 1973). Interest in fuzzy control has continued ever since, and the literature on the subject has grown rapidly. A survey of the field with fairly extensive references may be found in (Lee, 1990) or, more recently, in (Sala et al., 2005).In Mamdani's model the fuzzy implication is modeled by Mamdani's minimum operator, the conjunction operator is min, the t-norm from compositional rule is min and for the aggregation of the rules the max operator is used. In order to explain the working with this model of FLC will be considered the example from (Rakic, 2010)[13] where a simple two-input one-output problem that includes three rules is examined:

Rule1 : IF x is A3 OR y is B1 THEN z is C1
Rule2 : IF x is A2 AND y is B2 THEN z is C2
Rule3 : IF x is A1 THEN z is C3.

Step 1: Fuzzification

The first step is to take the crisp inputs, x0 and y0, and determine the degree to which these inputs belong to each of the appropriate fuzzy sets. According to Figure (4.2) one obtains

$$\mu A1\ (x0) = 0.5, \mu A2\ (x0) = 0.2, \mu B1\ (y0) = 0.1, \mu B2\ (y0) = 0.7$$

Step 2: Rules evaluation

The fuzzified inputs are applied to the antecedents of the fuzzy rules. If a given fuzzy rule has multiple antecedents, the fuzzy operator (AND or OR) is used to obtain a single number that represents the result of the antecedent evaluation. To evaluate the disjunction of the rule antecedents, one uses the OR fuzzy operation. Typically, the classical fuzzy operation union is used :

$$\mu A\cup B(x) = \max\{\mu A(x), \mu B(x)\}.$$

Similarly, in order to evaluate the conjunction of the rule antecedents, the AND fuzzy operation intersection is applied:

$$\mu A\cap B(x) = \min\{\mu A(x), \mu B(x)\}.$$

The result is given in the Figure (4.3).

Now the result of the antecedent evaluation can be applied to the membership function of the consequent. The most common method is to cut the consequent membership function at the level of the antecedent truth; this method is called clipping. Because top of the membership function is sliced, the clipped fuzzy set loses some information. However, clipping is preferred because it involves less complex and generates an aggregated output surface that is easier to defuzzify. Another method, named scaling, offers a better approach for preserving the original shape of the fuzzy set: the original membership function of the rule consequent is adjusted by multiplying all its membership degrees by the truth value of the rule antecedent. Figure (4.4).

Step 3: Aggregation of the rule outputs

The membership functions of all rule consequents previously clipped or scaled are combined into a single fuzzy set as shown in Figure(4.5).

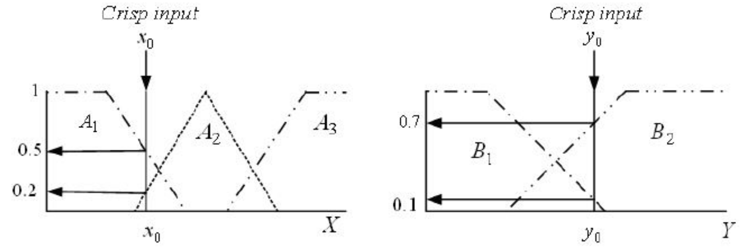
Step 4: Defuzzification

The most popular defuzzification method is the centroid technique. It finds a point representing the center of gravity (COG) of the aggregated fuzzy set A, on the interval [a, b]. A reasonable estimate can be obtained by calculating it over a sample of points. According to Figure(3.6), in our case results,

$$\text{COG} =(0 + 10 + 20) \times 0.1 + (30 + 40 + 50 + 60) \times 0.2 + (70 + 80 + 90 + 100) \times 0.5$$
$$0.1 + 0.1 + 0.1 + 0.2 + 0.2 + 0.2 + 0.2 + 0.5 + 0.5 + 0.5 + 0.5= 67.4$$

4.3 Universal approximators

Using the Stone-Weierstrass theorem, Wang in (Wang, 1992) showed that fuzzy logic control systems of the form ,, Ri : IF x is Ai AND y is Bi THEN z is Ci , i = 1, ..., n With,



Figure(4.2) Fuzzification

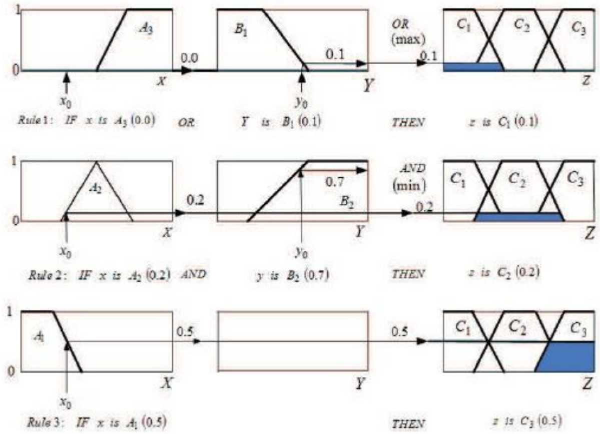
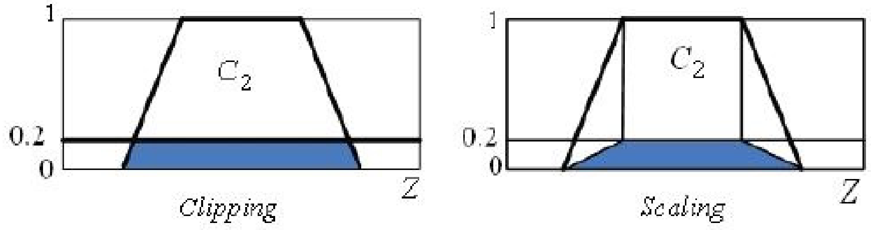
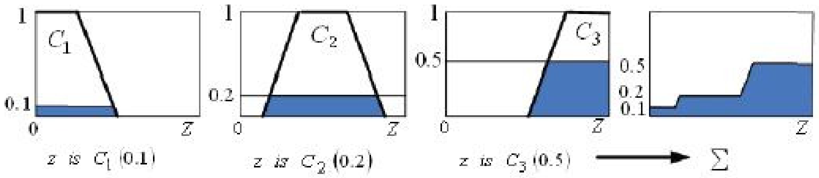

1
0
A_3
0.0
X
x_0
B_1
0.1
Y
y_0
OR (max)
0.1
C_1
C_2
C_3
Z
Rule 1: IF x is A_3 (0.0) OR Y is B_1 (0.1) THEN z is C_1 (0.1)
1
0
A_2
0.2
X
x_0
0.7
B_2
Y
y_0
AND (min)
0.2
C_1
C_2
C_3
Z
Rule 2: IF x is A_2 (0.2) AND y is B_2 (0.7) THEN z is C_2 (0.2)
1
0
A_1
0.5
X
x_0
Y
0.5
C_1
C_2
C_3
Z
Rule 3: IF x is A_1 (0.5) THEN z is C_3 (0.5)

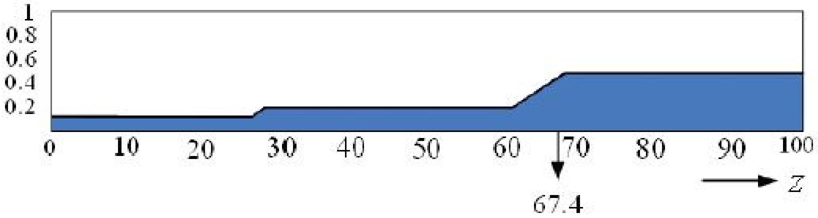
Figure (4.3) Rules Evaluation



Figure(3.4) Clipping and Scaling



Figure(4.5) Aggregation of the Rules output



Figure(4.6) Defuzzification

4.2.1 Mamdani fuzzy logic controller

- Gaussian membership functions

$$\mu_A(x) = \exp[-\frac{1}{2}(\frac{x-x_0}{\sigma})^2]$$

where x0 is the position of the peak relative to the universe and σ is the standard deviation

- Singleton fuzzifier

fuzzifier(x) = x

- Fuzzy product conjunction

μAi (u) AND μBi (v) = μAi (u)μBi (v)

- Larsen (fuzzy product) implication

[μAi (u) AND μBi (v)] → μCi (w) = μAi (u)μBi (v)μCi (w)

- Centroid deffuzification method

$$z = \frac{\sum_{i=1}^{n} c_i \mu_{A_i}(x)\mu_{B_i}(y)}{\sum_{i=1}^{n} \mu_{A_i}(x)\mu_{B_i}(y)}$$

where ci is the center of Ci , are universal approximators, i.e. they can approximate any continuous function on a compact set to an arbitrary accuracy.

4.4 Fuzzy Based Congestion Estimation

Fuzzification: The mapping from a real-valued point to a fuzzy set is known as Fuzzification which receives other robots information in order to convert it into fuzzy linguistic variable inputs. The fuzzy logic is chosen based upon the following two reasons: a) In between the normal and abnormal events, clear boundaries are not present, b) Fuzzy rules should level the normality and abnormality separation. The fuzzy set can be represented using the mathematical formation known as membership function.

Rule Definition: Conditional statements are used to implement a membership function which characterizes a fuzzy set A in x. When the fuzzy statement in an antecedent is true to some degree of membership, the consequent of the same degree also proves to be true.

Rule structure: If antecedent then consequent. The rule, When both the variables have different values high and low, then we can get a generous output otherwise a malicious output is detected. For a fuzzy classification system, the case or an object can be classified by applying the set of fuzzy rules which depend upon the linguistic values of its attributes. The rule is functioned at the number given by the antecedent which has a value between 0 and 1. The input can be fuzzified by evaluating the antecedent and then essential fuzzy operators can be applied. The consequent obtains this result as the inference.

We will now describe our methodology for fuzzy logic approach to control congestion in the network. In controlling congestion, the three most important variables are the RTT, Link utilization, Sending Rate. With fuzzy logic, we assign grade values to our three variables. Our fuzzy set therefore consists of three fuzzy variables.

Fuzzy set = {R,L,S } (R –RTT, L-LinkUtilization, S- SendingRate)
Fuzzy logic implements human experiences and preferences via membership functions and fuzzy rules. In this work, the fuzzy if-then rules consider the parameters: R –RTT, L-Link Utilization, S-Sending Rate.

The fuzzy logic uses two input variables and one output variable. The two input variables to be fuzzified are RTT and Link Utilization. The inputs are fuzzified, implicated, aggregated and defuzzified to get the output as Sending Rate. The linguistic variables associated with the input variables are Low (L), Medium(M) and high (H). The output variables use three linguistic variables H, M, and L where H denotes high Sending Rate, M denotes Medium Sending Rate and L denotes Low Sending Rate. The rules for the FIS are shown below as shown in Table1. They utilize the AND method which is based on the min function. The FIS rules of the Fuzzy Inference System are:

Table 1 Fuzzy Rules (LU-Link Utilization, SR-Sending Rate) (L-Low, M-Medium, H-High)Fuzzy Rules

If RTT is Less	**and Lu is less**	**then SR is High.**
If RTT is Less	and Lu is Medium	then SR is Medium.
If RTT is Less	and Lu is High	then SR is Low.
If RTT is Medium	and Lu is less	then SR is Medium.
If RTT is Medium	and Lu is Medium	then SR is Medium.
If RTT is Medium	and Lu is High	then SR is low.
If RTT is High	and Lu is less	then SR is Medium.
If RTT is High	and Lu is Medium	then SR is Low.
If RTT is High	and Lu is High	then SR is Very Low.

Control Action to be taken after defuzzification as shown in table 2,and description is given in section 3.1 Rate Adjusting Action. (MIAD-Multiplicative Increase and Additive Decrease, AIAD-Additive Increase and Additive Decrease, AIMD-Additive Increase and Multiplicative Decrease).

Table 2 (Control Action for smooth Sending Rate)

Sending Rate		**Link Utilization**		
		L	**M**	**H**
RTT	**L**	**MIAD**	**AIAD**	**AIMD**
	M	**AIAD**	**AIMD**	**AIMD**
	H	**AIAD**	**MD**	**MD**

5. Simulation Results And Discussion

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 Sending Rate and FSR (Fuzzy Sending Rate). Topology is as shown in figure(5.1). The network topology for single multicast. is below of 15 receivers: Simulation parameters are as shown in Table 3.

Table 3 Simulation Parameters

Parameters	Value
Link Bandwidth	15-100Mbps
Link Delay	25ms
Queue Size	60-99 packets
Sending Rate(initial)	85Mbps
No.of Groups	5
No.0f Receivers	15
RTT	150ms
Packet Size	300
Session Time	500ms
Congestion Status	High, Medium, Low

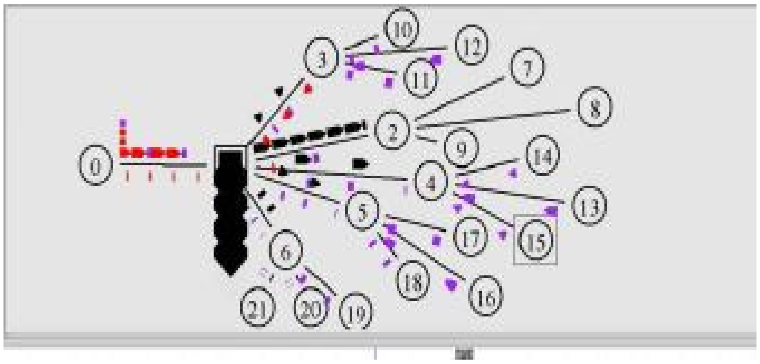


Figure (5.1) Multicast Topology

5.1 Results and Comparisons

5.1.1 Packet Delivery Ratio: Number of packets sent to the recovers is more than the TFMCC comparing FSR(Fuzzy sending Rate as shown in below. The sending rate of FSR sender which has less monitor and notch is smoother than that of TFMCC sender. In such dynamic network, the tight track to expected rate calculated by CLR will make multicast flow unstable. On the contrary, FSR with fuzzy controller can adjust sending rate adaptively with the knowledge of feedback information, so FSR has smoother sending rate than TFMCC as shown in Figure (5.2) and (4.3).

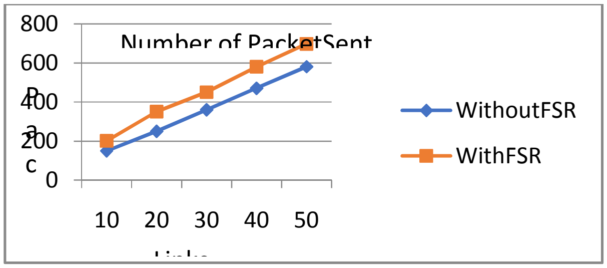


Figure (5.2) Packet Delivery Ratio

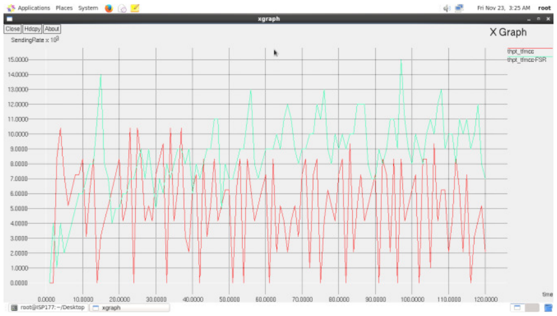


Figure (5.3) Throughtput TFMCC vs TFMCC-FSR

5.1.2 Packet loss Ratio:The variation is due to the link utilization of the network. The new proposed sending rate (*FSR*) shows the better result than the existing TFMCC link utilization strategy because of the less packet loss ratio. This is because of efficient link utilization using Fuzzy logic Controller as shown in figure (5.4).

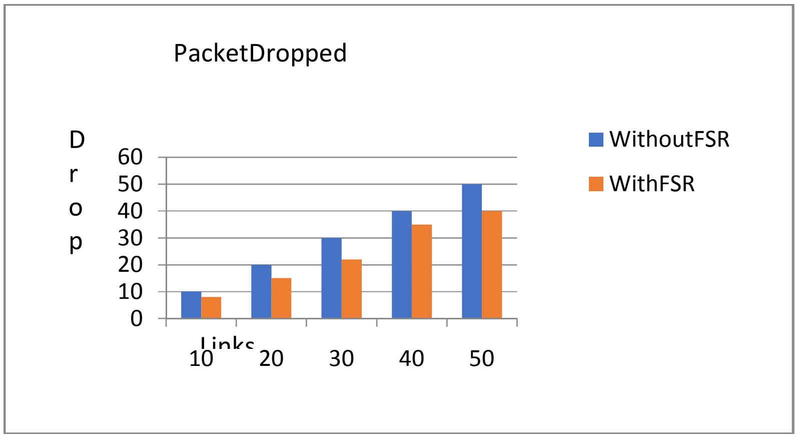


Figure (5.4) Packet loss Ratio

5.1.3 Throuhput Comparing TFMCC vs TFMCC-FSR

Figure(5.5) shows the variation of throughput with the time. It shows the maximum time needed for the multi-cast source till reaching a steady state throughput. It is clear that proposed approach Outperforms existing approach.

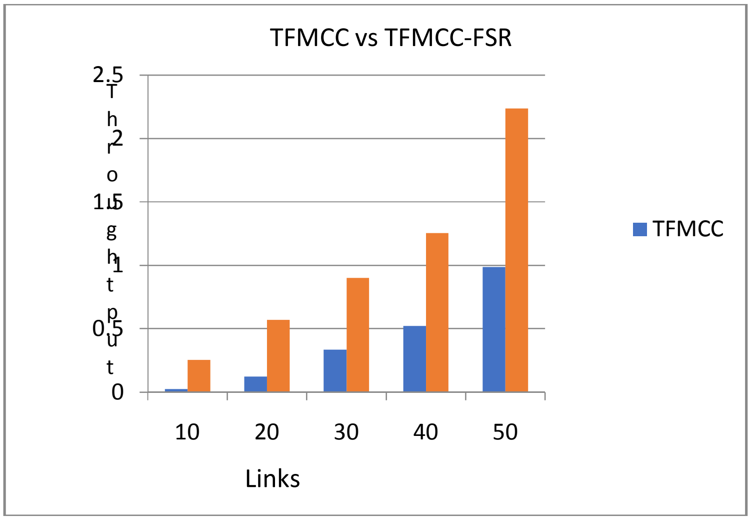
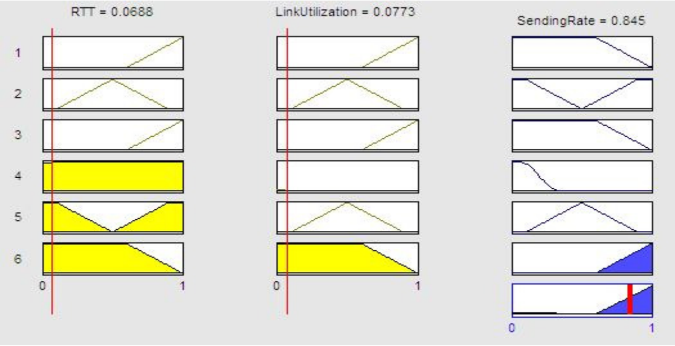


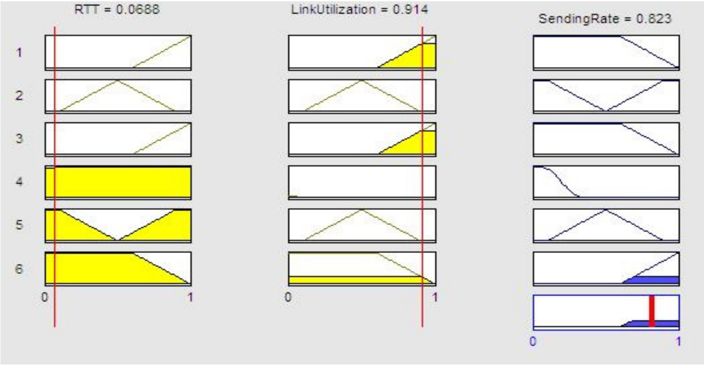
Figure (5.5) Throughput TFMCC vs TFMCC-FSR

5.1.4 Fuzzy Controller operations

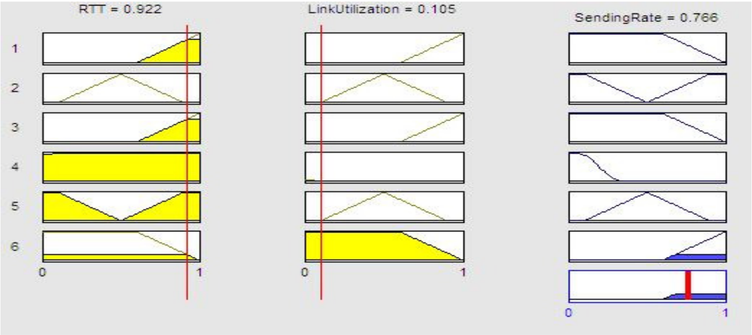
a) RTT is low and LinkUtilization is Low then sending rate High



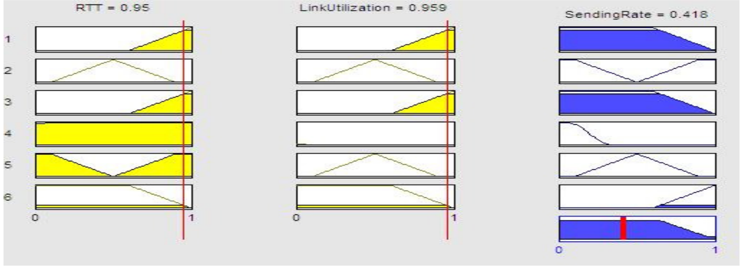
b) RTT is low and LinkUtilization is High then Sending Rate is Medium



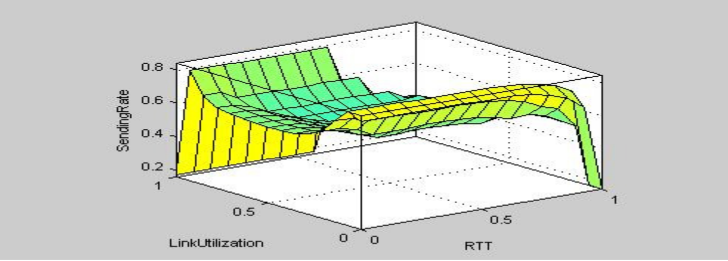
c) RTT is high and LinkUtilization is Low Then Sending rate is Medium



d) RTT is High and LinkUtilization is High Sending Rate is low



Fuuzy Graph



Figure(5.6) Fuzzy Sending Rate

Control surface of the Fuzzy Congestion Controller is shown in Figure (5.6). The control surface is shaped by the rule base and the linguistic values of the linguistic variables. By observing the progress of simulation, and modifying the rules and definitions of the linguistic values, FSR can be tuned to achieve better Link utilization, and smoothened Sending Rate.

6. Conclusion

For the requirement of multimedia application based on IP multicast, we have proposed an improved rate adaption scheme named FSR to smoothen the sending rate of TFMCC sender. FSR introduces four actions to adjust sending rate and uses a fuzzy controller for making decision to choose one of the four actions adaptively. In dynamic network environment, fuzzy controller uses the difference between expected rate and sending rate to reflect the congestion degree, as well as the difference between two latest consecutive expected rates to predict the trend of network. Under the fuzzy controller, MIAD,AIAD,AIMD actions eliminate the "sawtooth" phenomenon in TFMCC, which is crucial for smoothing sending rate. When the available bandwidth is turning abundant, an algorithm for congestion control which based on utilization of link and taking decision according to high, medium and low congestion.

REFERENCES

[1] J. Widmer, R. Denda, and M. Mauve, "A Survey on TCP-friendly Congestion Control,"IEEE Network, vol. 15, no. 3, 2001, pp.28-37.

[2] Gu-In Kwon and J. W. Byers, "Smooth Multirate Multicast Congestion Control," Proc.of IEEE INFOCOM 2003, San Francisco: IEEE Communications Society, 2003,pp.1022-1032.

[3] J. Li, M.Yuksel, X. Fan, and S. Kalyanaraman, "Generalized Multicast Congestion Control". Computer Networks. 51(2007), pp.1421-1443.

[4] M. Rodríguez Pérez, S. Herrería Alonso, M. Fernández Veiga, and C.López García. "An Adaptive Multirate Congestion Control Protocol for Multicast Communications".Computer Communications, 29 (2006), pp.2247–2260.

[5] J. Widmer and M. Handley, "Extending Equation-based Congestion Control to Multicast Applications," Proc. Of ACM SIGCOMM'01,August 2001, San Diego, California,USA.

[6] J. Padhye, V. Firoiu, D. F. Towsley, and J. F. Kurose. "Modeling TCP Reno Performance: A Simple Model and Its Empirical Validation," IEEE/ACM Transactions on Networking, vol. 8,no. 3, April 2000, pp.133–145.

[7] W. Kammoun and H. Youssef, "Improving the Perfor-mance of End-to-End Single Rate Multicast congestion.

[8] G.-I. Kwon and J. W. Byers, "Leveraging Single Rate Schemes in Multiple Rate Multicast congestion Control Design," IEEE Journal on Selected Areas in Communica-tions, Vol. 22, No.10, 2004, pp. 1975-1 1986.

[9] R. Chakravarthi and C. Gomathy, "A Fuzzy Approach To Detect And Control Congestion In Wireless Sensor Net-works," Indian Journal of Computer Science and Engi-neering (IJCSE),Vol. 3, No. 3, 2011, pp. 476-483.

[10] J. Bian, Z.-Q. Wang, H.-H. Ke and G.-Z. Zhang, "Con-gestion Control Protocol Based on Delay Parameters for Layered Multicast Communication," ICCT International Conference on communication Technology, 10 April 2007.

[11] Manisha Manjula,Rajesh Mishra,Joysna ,"Link Utilization Based Multicast Congestion Control",Published On Communication and Networks,2013,5,649-653

[12] "Fuzzy Controller",Leonid Reznic Book, ISBN 0 7506 3429 4An imprint of Butterworth-Heinemann Linacre House, Jordan Hill, Oxford OX2 8DP A division of Reed Educational and Professional Publishing Ltd

[13] Rakic, A. (2010). Fuzzy Logic. Introduction 3. Fuzzy Inference, ETF Beograd. URL: http://www.docstoc.com/docs/52570644/Fuzzy-logic-3

Authors

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.