Prof. Ravi Verma, Manoj Kumar Khinchi / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 3, May-Jun 2012, pp.1207-1212

Congestion control in network management system

*Prof. Ravi Verma, **Manoj Kumar Khinchi
*Department of CSE,Mahakal Institute of Technology & Sciences, Ujjain India
**CSE Dept, Faculty of Engineering and Technology, Mody Institute of Technology and Science ,Lakshmangarh India

I.ABSTRACT

As we know that computing work is growing around the world so that it is obvious that traffic in network is also having very largely as result network are suffering from various problems related to the Data like Data lose, Data duplication and Delay. Time management are very necessary in all the organization but at the traffic load will be higher ,channel capacity are not able to carry the desired data to its desired destination so that it is necessary to maintain load of data packet at communication channel for this objective we have various protocol implementing at network layer that all protocol are working for MAC and LLC layer in order to Detect and Correct the Congestion but still Congestion is there that they are just controlling the congestion but not eliminating the congestion and then the result will be lose of data .

We are introducing a new protocol that will detect the congestion but it will not happen in all network.

Time Slotted Protocol gives opportunity to avoid congestion from the network completely so that best network services can be achieved with greater reliable services.

Keywords : Data lose , Congestion, Time Slotted Protocol, MAC,LLC

II. INTRODUCTION

As we have discussed that available network services provides the services related to the congestion detection and correction. In coming session we will discussed about all those protocol which has been working in network to reduce the data loss or to maintain all the

Problem related with congestion.

Data communication required at least two devices working together one to send and other to receive it need a greater idea for coordination for an intelligent exchange of Information .The most responsibility of Data Link Layer is to maintain Flow and Error control[9] . Data communication and computer network provide lots of protocols and algorithms at Data link layer to make the proper Error and Flow control communication as all Know that **flow control refers to a set of procedures used to restrict the amount of data that the sender can be send before waiting for acknowledgment.**

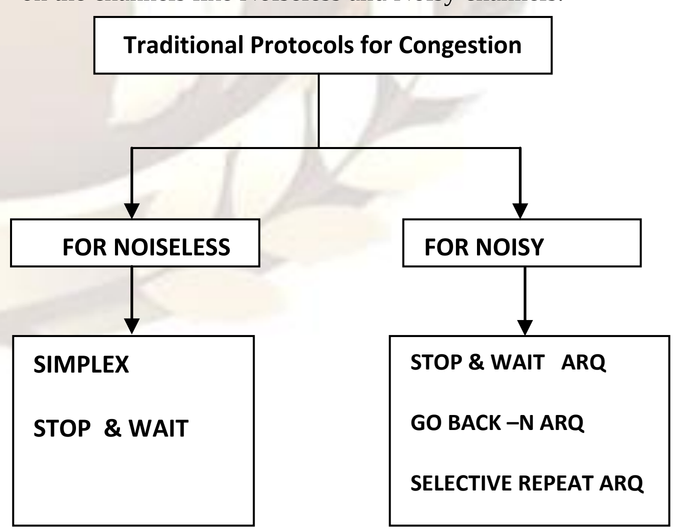
On the other hand error control is both the error detection and error correction .It allows the receiver to inform the sender of any frames lost or damaged in transmission and coordinates the retransmission of those frames [1]. **Error control at Data link layer is based on automatic repeat request, which is the retransmission of Data [2].**

Proposed Model over Existing work: our proposed model i.e. **"Time Slotted Protocol"** Will be the greater solution in order to reduce data flow and error control services complexity our proposed model uses a symbolic time period for each successive delivery of data. Networking provides many protocols as we have discussed above but no one will provides the guaranteed reliable congestion less delivery

III.RELATED WORK

Data communication and Computer Network has been working with many traditional protocols all has been defined at the data link layer for error, flow control.

All this protocol is worked well for both Error and Flow control but no one will be used to eliminate the complexity of Error and Flow Control. Data communication and computer network has been classified all this protocols based on the channels like Noiseless and Noisy channels.



this traditional protocols are working for both Noisy and Noiseless channels, Network are making very useful with the implementation of such protocols at large traffic area, they all are provides congestion less environment with full duplex communication in the network with low risk of data loss .but when we are individually analyze the performance of all protocols we seen that QoS is not available properly for the point of view of congestion.

Now we are considering the simplex protocol for noiseless communication it will provides the functionality for simplex communication system as the algorithm works as:

Stop and Wait Protocol: This is the simplest file control protocol in which the sender transmits a frame and then waits for an acknowledgement, either positive or negative, from the receiver before proceeding [8]. If a positive acknowledgement is received, the sender transmits the next packet; else it retransmits the same frame. However, this protocol has one major flaw in it. If a packet or an acknowledgement is completely destroyed in transit due to a noise burst, a deadlock will occur because the sender cannot proceed until it receives an acknowledgement [10]. This problem may be solved using timers on the sender's side. When the frame is transmitted, the timer is set. If there is no response from the receiver within a certain time interval, the timer goes off and the frame may be retransmitted.

Sliding Window Protocols: Instead of the use of timers, the stop and wait protocol still suffers from a few drawbacks. Firstly, if the receiver had the capacity to accept more than one frame, its resources are being underutilized. Secondly, if the receiver was busy and did not wish to receive any more packets, it may delay the acknowledgement. However, the timer on the sender's side may go off and cause an unnecessary retransmission. These drawbacks are overcome by the sliding window protocols[5].

In sliding window protocols the sender's data link layer maintains a 'sending window' which consists of a set of sequence numbers corresponding to the frames it is permitted to send. Similarly, the receiver maintains a 'receiving window' corresponding to the set of frames it is permitted to accept. The window size is dependent on the retransmission policy and it may differ in values for the receiver's and the sender's window [15]. The sequence numbers within the sender's window represent the frames sent but as yet not acknowledged. Whenever a new packet arrives from the network layer, the upper edge of the window is advanced by one. When an acknowledgement arrives from the receiver the lower edge is advanced by one. The receiver's window corresponds to the frames that the receiver's data link layer may accept [3]. When a frame with sequence number equal to the lower edge of the window is received, it is passed to the network layer, an acknowledgement is generated and the window is rotated by one. If however, a frame falling outside the window is received, the receiver's data link layer has two options [11]. It may either discard this frame and all subsequent frames until the desired frame is received or it may accept these frames or buffer them until the appropriate frame is received and then pass the frames to the network layer in sequence [7].

Go **Back 'n':** If a frame is lost or received in error, the receiver may simply discard all subsequent frames, sending no acknowledgments for the discarded frames. In this case the receive window is of size 1. Since no acknowledgements are being received the sender's window will fill up, the sender will eventually time out and retransmit all the unacknowledged frames in order starting from the damaged or lost frame [4]. The maximum window size for this protocol can be obtained as follows. Assume that the window size of the sender is n. So the window will initially contain the frames with sequence numbers from 0 to (w-1) [16]. Consider that the sender transmits all these frames and the receiver's data link layer receives all of them correctly. However, the sender's data link layer does not receive any acknowledgements as all of them are lost. So the sender will retransmit all the frames after its timer goes off. However the receiver window has already advanced to w. Hence to avoid overlap , the sum of the two windows should be less than the sequence number space.

w-1 + 1 < Sequence Number Space

i.e., w < Sequence Number Space

Maximum Window Size = Sequence Number Space - 1

Selective Repeat: In this protocol rather than discard all the subsequent frames following a damaged or lost frame, the receiver's data link layer simply stores them in buffers. When the sender does not receive an acknowledgement for the first frame its timer goes off after a certain time interval and it retransmits only the lost frame. Assuming error - free transmission this time, the sender's data link layer will have a sequence of a many correct frames which it can hand over to the network layer [13]. Thus there is less overhead in retransmission than in the case of Go Back n protocol. In case of selective repeat protocol the window size may be calculated as follows. Assume that the size of both the sender's and the receiver's window is w. So initially both of them contain the values 0 to (w-1). Consider that sender's data link layer transmits all the w frames; the receiver's data link layer receives them correctly and sends acknowledgements for each of them. However, all the acknowledgements are lost and the sender does not advance it's window. The receiver window at this point contains the values w to (2w-1) [14]. To avoid overlap when the sender's data link layer retransmits, we must have the sum of these two windows less than sequence number space. Hence, we get the condition [9].

Maximum Window Size = Sequence Number Space / 2

Accordingly if we are analyzing the data rate on the basis of the entire above mention algorithm we are getting the following result on the basis of the data transmission by placing different protocols as:

- Numbering. on 32 bit
- As a function of link speed we have a different wrapping time…….
- The same application may have problem if sequence wrap around or if successive connection have overlapping sequences [12].

Network Speed	Wrap around times
T1(1.5 Mbps)	6.4h
Ethernet (10 Mbps)	57m
T3(45Mbps)	13m
FDDI(100Mbps)	6m
STS-3(155Mbps)	4m
STS-12(622Mbps)	55s

IV.TIME SLOTTED PROTOCOL

our proposed protocols playing very important role in high traffic congestion network it is not necessary that every network protocol provides the robust network on large data traffic all of the available protocol for noisy and noiseless channels are never provides the guaranteed communication as long as the traffic growing into the internet when the data packets traffic grow then we need to have new mechanism that we take the responsibility of maintaining QoS in such a engaged data packet network we have to implements the following algorithm that provides the address of making the solution of high data traffic network route complications with good performance of that transmission i.e. QoS.

Protocol Definition:

```
#define MAX_PKT 1024

typedef enum {false, true} boolean;
typedef unsigned int seq_nr;
typedef struct {unsigned char data[MAX_PKT];}
typedef enum {data, ack, nak} frame_kind;
```

Figure 1.1

As the definition say that how the data packet has been Initialized at the data link layer for the successful transmission of data packet to the receiver site to make the algorithm time slotted we need the actual timer _event to be set to the flag bit as start _event=0; but obviously they will be by default initialize with the start_event=null ; as the first data packet has been introduce for the typedef struct {unsigned char data [MAX_PKT];} used for the all level of the MXA_PKT transmitted data type frames should be prepared and acknowledged by the actual synchronization of that time RR_Session timer transmission EXITS such a scenario is working until all packet of the connection will not
be terminated by both of the user who are involved in the communication channel such sachem is not only working for a similar kind of network but for all the noisy and noiseless channels are also can be able to make the same communication channel in a regular timer_event with the maintenances of the QoS at all the network level [6] for making the communication stronger for the point of view of the reliability as well as making the performance excellent by implementing the following prototypes:

```
typedef struct {
frame_kind kind;
seq_nr seq;
seq_nr ack;
packet info;
}frame;
If(event(timeout))
{
Initialize_timer start_event-0;
Start_bit=0;
Starttimer();
Temp=total_initialized_frame;
T=Temp;
While(temp<total_initialized_frame)
{
Sendframe(T);
T=t+1;
}    }
```

TSP Algorithm: 1.1

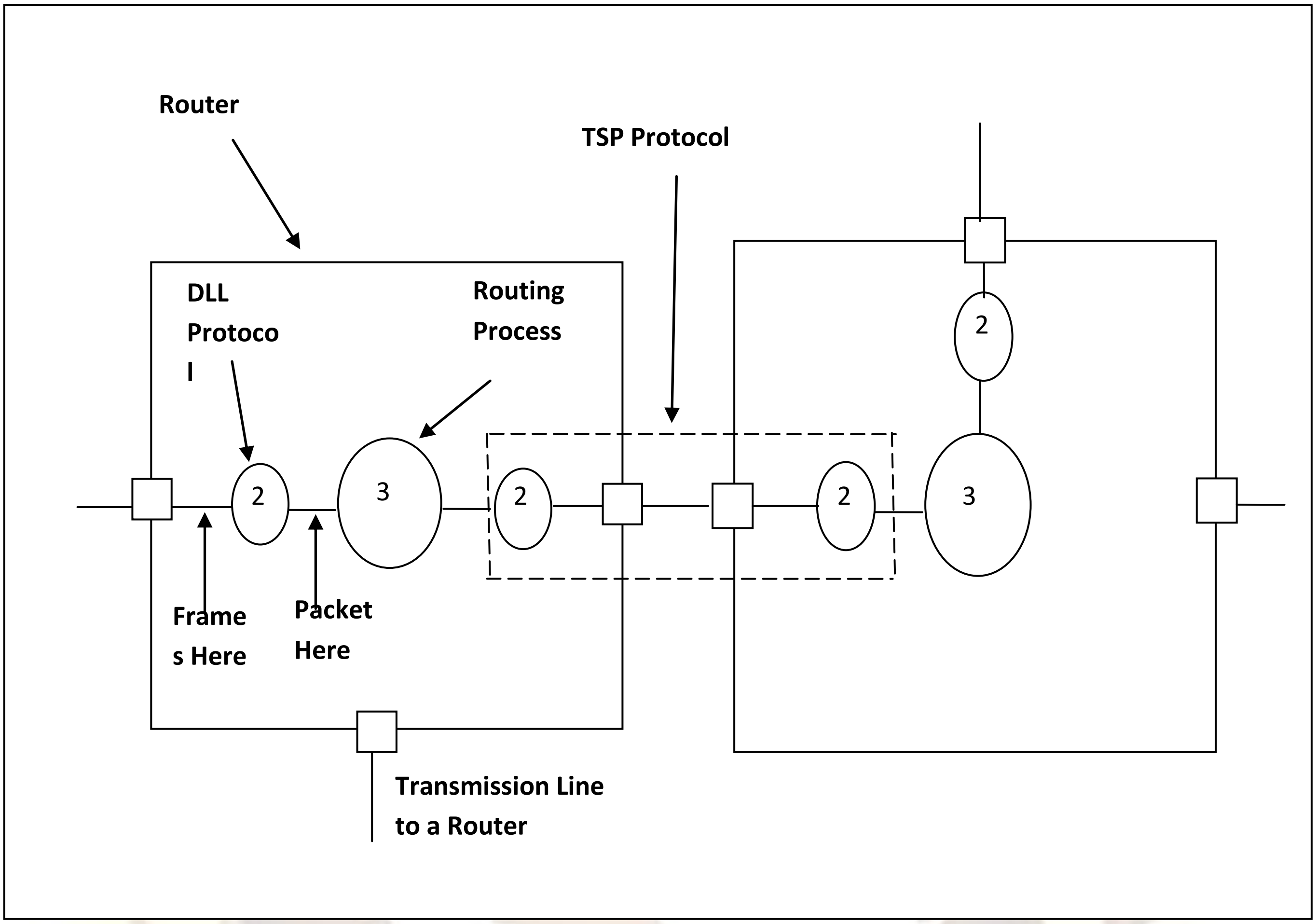


Figure1.2: Implementation of TSP Protocol at Data Link Layer

V. IMPLEMENTATION POLICY

The stations on wireless or wired network are a maximum of 600 km apart. If we

- Network layer on A gives packet to its DLL. B receives packet correctly and is sent to its Network layer. B sends ack frame back to A.
- Ack frame gets lost.
- DLL on A times out. A takes lost ack to mean that frame never made it to B. Sends frame containing packet 1 again.
- B receives packet 1 twice and may process it twice. Protocol fails.

VI. IMPLEMENTATION OF TSP

As the following diagram show that how the new protocols has been working with the upcoming data frames for the Congestion less communication system it follow the steps like:

- Firstly the Proposed TSP protocols have making the connection with the available transmission line to a protocol.
- All the necessary function has already being working there for the process of DLL.
- As soon as the frames are arrived it will be treated as frames for the further DLL processing.
- Now frames have been sanded for the routing process in the form of data packets.
- The similar processes have also been done at the other end of router that will be the intermediate node between the processes.
- On basis of the upcoming and outgoing delivery of data packet we are getting the worst case complexity of O (log n).

- Average case complexity seems on the simulator that show much better performance than other protocols i.e. o(n.)
- Figure 1.2 has been showing the implementation of the TSP protocols for congestion less data control environment.

VII. CONCLUSION

As we have been studied yet that there are so many technology has available for the purpose of data and flow control in the network but yet the network has not available properly in the higher traffic circumstances it goes down to the weaken so that our proposed protocol provides the address of making the network healthy by the way of implementing the new protocol at the data link layer i.e. working as a protocol process layer in our TSP Module explaining in figure1.2 and as the algorithm 1.1 we can initialize the protocol with protocol definition mention as figure 1.1 For the point of view of complexity simulator also showing the co coordinator of worst case and average case complexity scenario as upcoming and outgoing delivery of data packet we are getting the worst case complexity of O (log n). Average case complexity seems on the simulator that show much better performance than other protocols i.e. o (n.).

VIII. REFERENCES

[1] H. Suzuki and F. A. Tobagi, “Fast bandwidth reservation scheme with multilink and multipath routing in ATM networks,” in *Proc. IEEE INFOCOM*, Mar. 1992, pp. 2233–2240.

[2] R. G. Ogier, V. Rutenburg, and N. Shacham, “Distributed algorithms for computing shortest pairs of disjoint paths,” *IEEE Trans. Inform. Theory*, vol. 39, pp. 443–455, Mar. 1993.

[3] K. Bharath-Kumar and J. M. Jaffe, “Routing to multiple destinations in computer networks,” *IEEE Trans. Commun.*, vol. COM–31, pp. 343–351, 1983.

[4] B. Dahill, B. N. Levine, E. Royer, and C. Shields, “A secure routing protocol for ad hoc networks,” in Proceedings of the International Conference on Network Protocols (ICNP), pp. 78-87, 2002.

[5] Y. Hu, A. Perrig and D. Johnson, Ariadne: A Secure On-demandRouting Protocol for Ad Hoc Networks, in Proceedings of ACM MOBICOM’02, 2002

[6] Z. J. Haas, M. Perlman, “The Performance of Query Control Schemes of the Zone Routing Protocol” IEEE/ACM Transactions on Networking, vol. 9, no. 4, pp. 427-438, Aug 2001.

[7] R. Vaishampayan and J. Garcia-Luna-Aceves, “Protocol for unifiedmulticasting through announcements(PUMA),” in *Proc. 1st IEEE Int.Conf. Mobile Ad-Hoc and Sensor Syst. (MASS ’04)*, 2004.

[08]E. M. Royer and C. E. Perkins, “Multicast operation of the Ad-hoc on-demand distance vector routing protocol,” in *Proc. 5th ACM/IEEE Int. Conf. Mobile Comput. Netw. (MobiCom’99)*, Sep. 1999, pp. 207–218.

[09]J. G. Jetcheva and D. B. Johnson, “Adaptive demand-driven multicast routing in multi-hop wireless Ad hoc networks,” in *Proc. 2th ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc’01)*, 2001, pp. 33–44.

[10]M. Transier, H. Fü□ ler, J. Widmer, M. Mauve, and W. Effelsberg, “Scalable position-based multicast for mobile ad-hoc networks,” *First Int. Workshop on Broadband Wireless Multimedia: Algorithms, Architectures and Applications (BroadWim ’04)*, Oct. 2004

[11]P. Tournois, J.-L. Vernet, and G. Bienvenu, “Sur l’analogie optique de certain montages e´lectroniques:Formationd’imagestemporelles de signaux electriques,” *C. R. Acad. Sci. Paris*, vol. 267, pp. 375–378, 1968.

[12] B. H. Kolner, “Space-time duality and the theory of temporal imaging,” *IEEE J. Quantum Electron.*, vol. 30, no. 8, pp. 1951–1963, Aug. 1994.

[13]A. A. Godil, B. A. Auld, and D. Bloom, “Picosecond time-lenses,” *IEEE J. Quantum Electron.*, vol. 30, no. 3, pp. 827–837, Mar. 1994.

[14]M. T. Kaufman, W. C. Banyai, A. A. Godil, and D. M. Bloom, “Timeto- frequenc converter for measuring picosecond optical pulses,” *Appl. Phys. Lett.*, vol. 64, pp. 270–272, 1994.

[15]L. K. Mouradian, F. Louradour, V Messager, A. Barth´el´emy, and C. Froehly, “Spectro-temporal imaging of femtosecond events,” *IEEE J. Quantum Electron.*, vol. 36, no. 7, pp. 795–801, Jul. 2000.

[16]C. Dorrer, “Single-shot measurement of the electric field of optical waveforms by use of time magnification and heterodyning,” *Opt. Lett.*, vol. 31, pp. 540–542, 2006.

Er. Ravi Verma: He has completed his MCA from Vikram University Ujjain India and M. Tech from Singhaniya University Jaipur Rajasthan , India in Computer Science and Engineering Now working as a research scholar in Computer Science and Engineering for doing PhD in future My interested research area is Ad Hoc Network. Presently He is working as a lecturer at MITS Ujjain India.