Pakistan Journal of Applied Sciences 2(3): 390-399, 2002

A Delay-throughput Performance Improvement to the p_i-Persistent Protocol

[1]Nurul Sarkar and [2]Krzysztof Pawlikowski
[1]Auckland University of Technology, Auckland, New Zealand
[2]University of Canterbury, Christchurch, New Zealand

Abstract: The p_i-persistent medium access protocol is an attractive solution for high-speed fiber-optic unidirectional bus networks. Sarkar and Pawlikowski (2001) investigated the delay versus throughput characteristics and fairness performance of p_i-persistent protocol, including the 1-persistent protocol. The main advantage of 1-persistent protocol over the p_i-persistent protocol is the reduced mean packet delay (network-wide and also for individual stations), but the potential drawback of this protocol is the lack of fairness. In contrast, the p_i-persistent protocol can provide good fairness in the sense that mean delays become almost station position independent, but the main drawback of this protocol is the inevitable increase in packet delay. In this paper we propose a new scheme, called p_i-persistent/HH (p_i-persistent protocol with hitch-hiking mechanism for slot pre-use) that can offers the combining advantages of 1-persistent and p_i-persistent protocols (ie. Low mean delay and fairness). The low mean delay as well as fairness is achieved by introducing a special mechanism of slot pre-use, called hitch-hiking (HH). In this paper the p_i-persistent/HH scheme is described and simulation results are presented to verify the projected performance.

Keywords: Packet Delay, Throughput, Poisson Arrival Stream, Fairness

Introduction

In a series of papers by Mukherjee *et al.* (Mukherjee, 1990; Mukherjee, 1991; Mukherjee and Meditch, 1988; Mukherjee and Kamal, 1994) proposed a medium access protocol for high-speed fiber-optic unidirectional bus networks, known as the p_i-persistent protocol. This protocol, with its very simple flow control mechanism allowing station i to access empty slots with station-dependent probability p_i $(0 \le p_i \le 1)$, has been seen as an alternative for solving fairness problems associated with the original IEEE 802.6 standard for the distributed queue dual bus (DQDB) for high-speed networks (IEEE 1990; Filipiak, 1989; Hahne, *et al.*, 1990; Mukherjee, 1992). The delay versus throughput characteristics and fairness performance of p_i-persistent protocol, including the 1-persistent protocol, have been investigated in (Sarkar and Pawlikowski, 2001) through a quantitative simulation study. The advantages of 1-persistent protocol are that slots on the bus are never wasted if a station has packets for transmission, each station experiences the minimum possible mean delay based on its location, and the network-wide delay is minimised. On the other hand, the 1-persistent protocol is maximally unfair as a station's performance strongly location dependent, both in terms of mean delay and delay variance. Therefore this scheme is not suitable for applications requiring long distances and high bandwidth. In contrast the p_i-persistent protocol can offers good fairness in the sense that the mean delays are almost independent from the location of station, which is a desirable feature for networks based on long buses. But, the main drawback of this protocol is that the network performance deteriorates (in terms of packet delays) if one secures its fairness. Therefore, to achieve optimum network performance, both in the sense of lower mean delay and fairness, the p_i-persistent protocol requires improvement. The main purpose of this paper is to extend the p_i-persistent protocol to include a technique of slot pre-use, so that we could eliminate the possibility of 'wasting' slots on the bus.

Pursuing this goal we propose a new scheme, called p_i-persistent/HH, developed as modification of the p_i-persistent protocol that can offers the combining advantages of 1-persistent and p_i-persistent protocols (ie. Low mean delay and fairness). The low mean delay as well as fairness is achieved from the protocol's work conserving property, ie., the slots are never wasted if a station has packets for transmission.

The Basic p_i-Persistent Protocol: The p_i-persistent protocol was originally proposed in (Mukherjee and Meditch, 1988) as a medium access protocol for high-speed fiber-optic unidirectional bus networks in which the stations' loads were assumed to be static. Dynamic versions of the protocol were later presented in (Mukherjee, 1990; Mukherjee, Lantz, Matloff and Banerjee, 1991). Under the p_i-persistent protocol, the head of the bus (HOB) generates empty slots (fixed-sized) which propagate downstream along the bus. If a station has a packet to send, it persists with its attempt to transmit the packet in the next empty slot with probability p_i until the packet is successfully transmitted. The p_i 's are computed in order to satisfy some pre-selected fairness criterion, such as the average packet delay, equal average effective service time, equal blocking probability of buffers, or equal throughput for all stations. It has been shown that each of these fairness criteria is satisfiable for any given load profile.

Shortcomings of p_i-Persistent Protocol: Although the p_i-persistent protocol provides stations service in a fair way in the sense that the mean delays may become independent from the station position, but this is achieved by sacrificing network performance (by increasing the mean packet delay), as stressed in (Mukherjee, 1990). Mathar and Pawlikowski (1997) have shown that the p_i-persistent protocol actually deteriorates delays at all upstream stations (in order to equalising the mean delay of all stations) without improving delays at downstream stations. Therefore, the protocol ends up penalising stations in the name of fairness and consequently increases the mean delays of all stations.

Another deficiency of the p_i-persistent protocol is that some empty slots left by station i $(i = 1, 2, \cdots, N-1)$ remain unused (despite of that station i may be ready to transmit packets), when they travel along the bus from station 1 to station N. Manjunath and Molle (1995) have pointed out that although the p_i-persistent algorithm achieves fairness, it is inherently wasteful. The protocol forces the upstream stations to blindly give up bandwidth that they could have used without knowing whether or not the downstream stations are able to use it.

It is very desirable to offer equal and fair access to network's resources for all stations, regardless of their locations on the bus. In addition, the mean packet delay of all stations as well as the network-wide mean delay should be minimised for any arbitrary choice of p_i. In the next two sections we describe the hitch-hiking mechanism of slot pre-use, and the resulting extension of p_i-persistent protocol, which we call p_i-persistent/HH protocol, that overcomes the above mentioned shortcomings of p_i-persistent protocol.

The Hitch-hiking Mechanism of Slot Pre-use: In the p_i-persistent protocol, the i-th station $(i = 1, 2, \cdots, N-1)$ transmits a packet in an empty slot with probability p_i $(0 \le p_i \le 1)$. This also means that with probability of $1 - p_i$, the station does not transmit a packet, and consequently, some empty slots can leave station i empty even though the station has a packet for transmission. Since station N is the last station on the bus, it transmits with probability 1 (if it has a packet to send). Therefore, under the p_i-persistent scheme some slots may remain unused despite that some stations are ready to transmit packets. The number of unused empty slots become more significant for larger values of λ, where λ is the total offered traffic to the network (in packets/slot). Now the question is how to avoid such loss of transmitting capability.

We propose a scheme, called hitch-hiking (HH) mechanism of slot pre-use, which, if applied in the p_i-persistent protocol, prevents slots from being left unused. The basic idea is that if station i has a packet to send, it first attempts to transmits it in an empty slot with probability p_i. A packet being transmitted following this rule we call the "permit holding" packet. Otherwise, the station transmits a packet even if it is not following the original rule of p_i-persistency. Such a packet transmitted "illegally" we call "hitch-hiking" (meaning free riding) packet, because it is transmitted without a permit. Therefore, under p_i-persistent protocol with the hitch-hiking mechanism (p_i-persistent/HH), slots are fully utilised either to carry permit holding (PH) or hitch-hiking (HH) packets. To take the full advantage of p_i-persistent protocol, the PH packets have higher priority than the HH ones, thus transmission of any HH packet can be interrupted by a PH packet.

Implementation of hitch-hiking mechanism for slot pre-use in the p_i-persistent protocol requires one additional bit in the slot header, called hitch-hiking bit (H). The slot header in the p_i-persistent/HH protocol therefore contains a busy bit (B) and a hitch-hiking bit (H). Table 1 shows the meaning of control bits (B,H). In an empty slot (B, H) = (0, 0). A slot carrying a permit holding packet has (B, H) = (1, X), (X means "arbitrary" or "don't care" value) and a slot carrying a hitch-hiking packet has (B, H) = (0, 1).

Table 1: The Meaning of (B, H) Bits

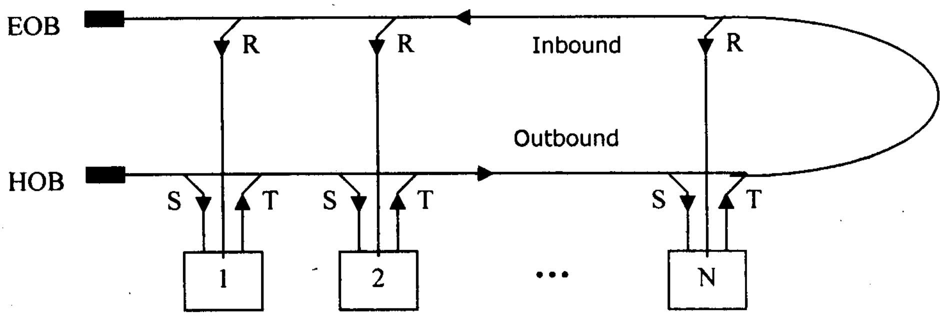
B	H	Contents of the slot
0	0	Empty slots
1	0	Slot contents PH packet
1	1	Slot contents PH packet
0	1	Slot contents HH packet

Description of p_i-Persistent/HH Protocol: The p_i-persistent/HH protocol applies to both folded bus and dual bus topologies. Although all of our work reported in this paper was obtained assuming the single-folded bus topology (in Fig. 1) proposed for D-net (Tseng and Chen, 1983), this work is also directly applicable to double-folded bus topology suggested for Expressnet (Tobagi, *et al.*, 1983). For dual bus topology, a source station must be able to determine the bus on which the destination is downstream from it and then transmit the packet on that bus.

Under the p_i-persistent/HH scheme, each station except station 1 is equipped with a number of transit queues (T-queues) and a local queue (L-queue). At station 1 only L-queue is required. The architecture of a station for the p_i-persistent/HH protocol is shown in Fig. 2.

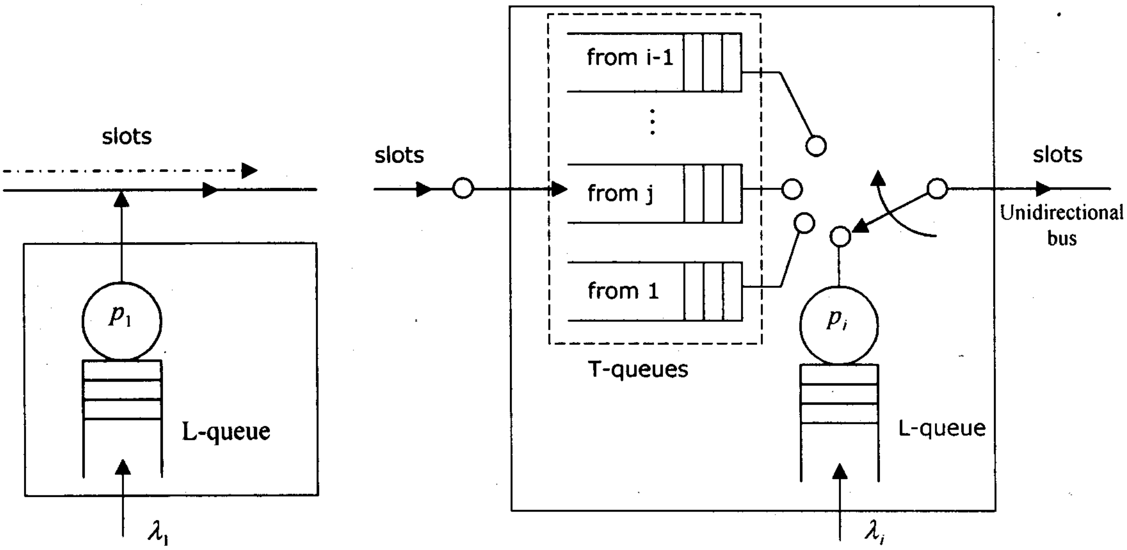
The L-queues are used to store locally generated packets. The T-queues are used for storing packets (both permit holding and hitch-hiking) arriving from upstream stations. The number of T-queues required at each station depends on the station's position on the bus. For example, station i $(i = 1, 2, \cdots, N)$ requires $i-1$ of T-queues to form $i-1$ queues of packets originating from different $i-1$ upstream stations. The packets (both PH and HH) originated from station i are forwarded (according to p_i-persistent/HH access algorithm) to T_i − queue of station $i+1$ in first-come-first-served (FCFS) order.

The flow-chart of p_i-persistent/HH algorithm for station 1 is shown in Fig. 3. Since station 1 is the most upstream station on the bus, naturally it can access any slot. If it has a packet to send, it first



T: transmit tap, R: receive tap, S: sense tap
HOB: Head of Bus, EOB: End of Bus

Fig. 1: A Single-folded Bus Network Topology



(a) Structure of Station 1

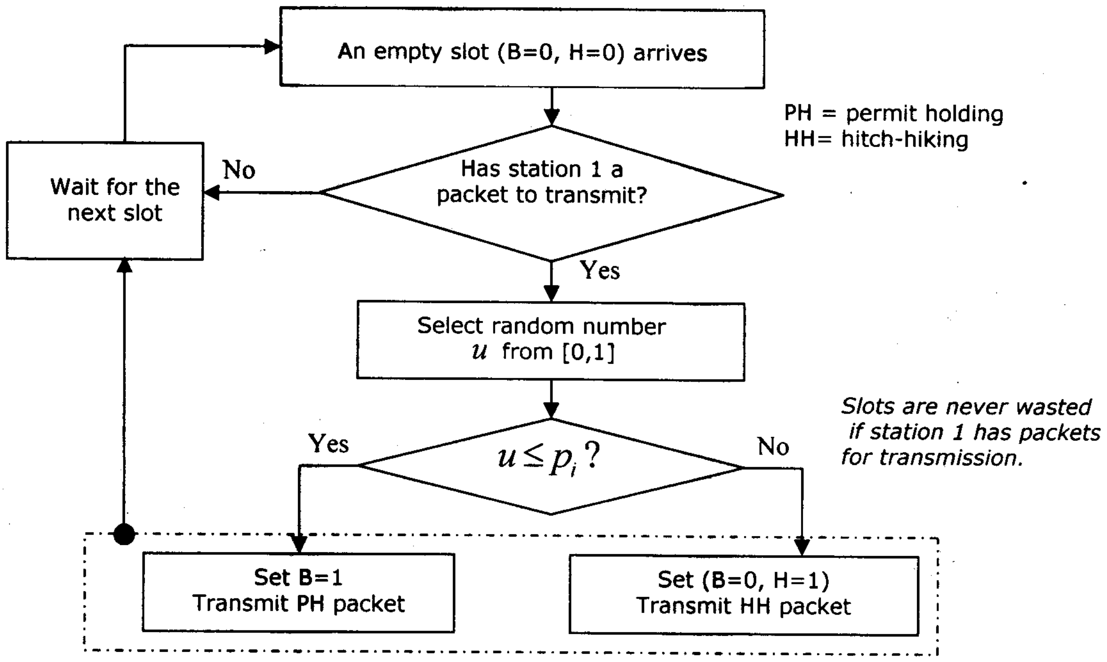
(b) Structure of Station i $(i = 2, 3, \cdots, N)$

Fig. 2: Stations Structure and Bus Interface. p_1 and p_i Represent Decision Rule Deciding about Accessing Slots with Probability p_1 at Station 1, and p_i at Station i $(i = 2, 3, \cdots, N)$, respectively

generates a random number u which is uniformly distributed between 0 and 1. If $u \le p_i$, (p_i is the channel access probability of that station), the packet from its L-queue is transmitted as PH packet, otherwise it transmits the packet as a HH packet. The transmitted packets (both PH and HH) from station 1 will be stored in the next downstream station's (i.e., Station 2) T_1 – queue. Unlike the p_i-persistent scheduling algorithm, the station 1 under the p_i-persistent/HH scheme never leaves an outgoing slot empty, unless it has no packet to transmit.

Fig. 4 shows the flow-chart of p_i-persistent/HH algorithm for station i $(i = 2, 3, \cdots, N)$. The incoming slots at station i can carry either PH, or HH packets, or be empty. If the incoming slot is carrying a PH packet originating from station j $(j = 1, 2, \cdots, i-1)$, then station i inserts the arriving packet at the end of its T_j – queue and replaces it by a packet from the head of T_j – queue by writing it into the same slot. Thus, the original order of packet delivery is maintained. The status of the slot remains unchanged and it leaves the station i as a PH. Therefore, if the incoming slot contains a PH packet originating from station j, then the slot will contain such a packet when leaving this station. The outgoing slot will carry a PH packet which had been found at the top of T_j – queue of station i.

Now we consider the case when the incoming slot carries a HH packet. If the incoming slot contains a HH packet



PH = permit holding
HH= hitch-hiking

Slots are never wasted if station 1 has packets for transmission.

Fig. 3: Flow-chart of p_i-persistent/HH Algorithm for Station 1

originating from station , then like before the packet will be copied and put at the end of $T_j -$queue at station i. Therefore, station i selects a packet from its L-queue (if any are available) to be written in the outgoing slot. If the L-queue is empty, the packet from the head of $T_1 - queue$ will be transmitted. If $T_1 - queue$ is empty, then packet from $T_2 - queue$ queue will be sent, and so on. Note that the packet selection from T-queues at station i starts from station 1 (most upstream station on the bus). We have chosen this order because the packets originating from station 1 visit more T-queues than packets from any other stations. By doing this, we should be able to minimise packet delay. The outgoing slot will contain a PH packet if station i transmits a packet from its L-queue with p_i, otherwise the slot will contain a HH packet. Again the station i never leaves an outgoing slot as empty if there is a packet ready for transmission.

If the incoming slot is empty, then station i selects a packet from its L-queue and write it into that slot for transmission. If the L-queue is empty, the packet from a T-queue is sent. If both L- and T-queues are empty, the slot will leave the station i empty. Notice that the packets generated at N (last station on the bus) can be transmitted directly without visiting any T-queues. If L- and T-queues are empty at station N, then the empty slots will be dropped off the bus at the end. Table 2 shows operations executed by station i during one slot time.

Note that the p_i-persistent/HH access algorithm (described above) is work conserving, since a station never leaves an empty outgoing slot unless all queues are empty.

The Performance of the p_i-Persistent/HH Protocol: To evaluate the performance of p_i-persistent/HH protocol, we developed a simulation model (written in C++) based on AKAROA II[3] system (to obtain results with controlled level of statistical errors). The length of each simulation run and the precision of the final estimates were controlled by the Spectral Analysis in Parallel Time Streams method (Pawlikowski and Yau, 1992; Pawlikowski, *et al.*, 1994) which stops the simulation automatically when the steady-state estimates of performance measures obtain the required relative precision (defined as the relative width of the confidence interval). The simulation results presented in the next section report the steady-state behaviour of network and have been obtained with the relative precision below 0.05, at 0.95 confidence level.

In the p_i-persistent/HH simulation model, we use the same set of $\{p_i\}$ as used in the p_i-persistent protocol. We have chosen this $\{p_i\}$ as the near-optimum values for $\{p_i\}$ for p_i-persistent/HH protocol by searching them empirically. More details about the optimisation of p_i for the p_i-persistent/HH protocol can be found in (Sarkar, 1996).

For infinite buffer case, the p_i-persistent protocol under

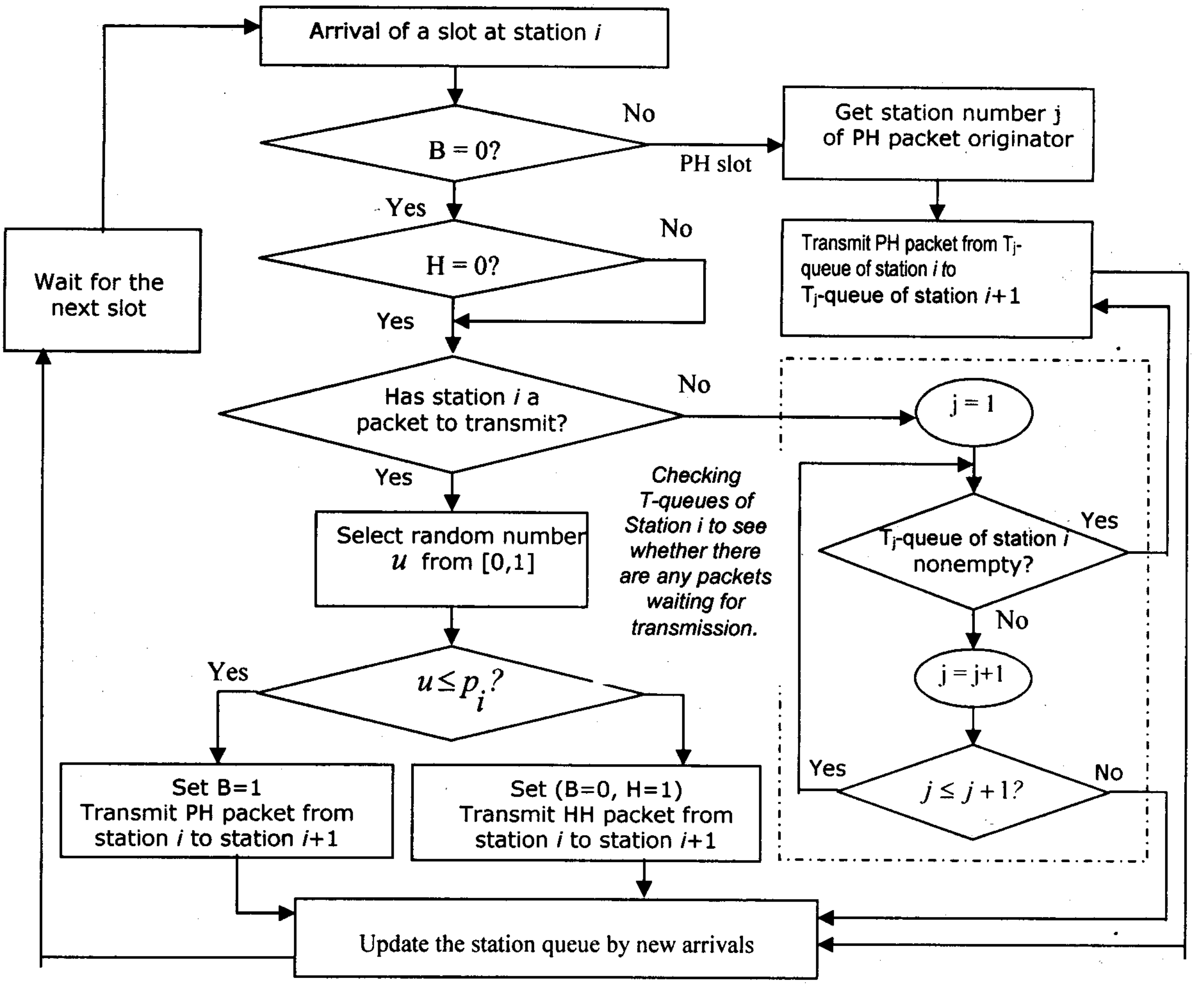


Fig. 4: Flow-chart of p_i-persistent/HH Algorithm for Station i $(i = 2, 3, \cdots, N)$.

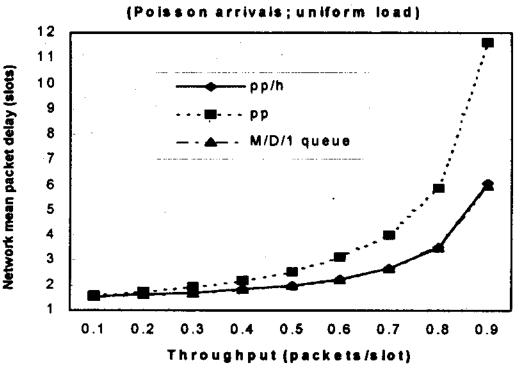


Fig. 5: Comparison of the Network Mean Packet Delay Versus Throughput Performance of the p_i-Persistent/HH Protocol (p/hh) and the p_i-persistent Protocol (pp). Also Shown is the Average System Delay Time for a Slotted M/D/1 queue

the equal average packet delay fairness criterion, should operate using

$$p_i = \frac{2(1-\lambda) + \lambda_i(1+\lambda-\lambda_N)}{(2-\lambda_N)(1-\sum_{j=1}^{i-1}\lambda_j)} \qquad (1)$$

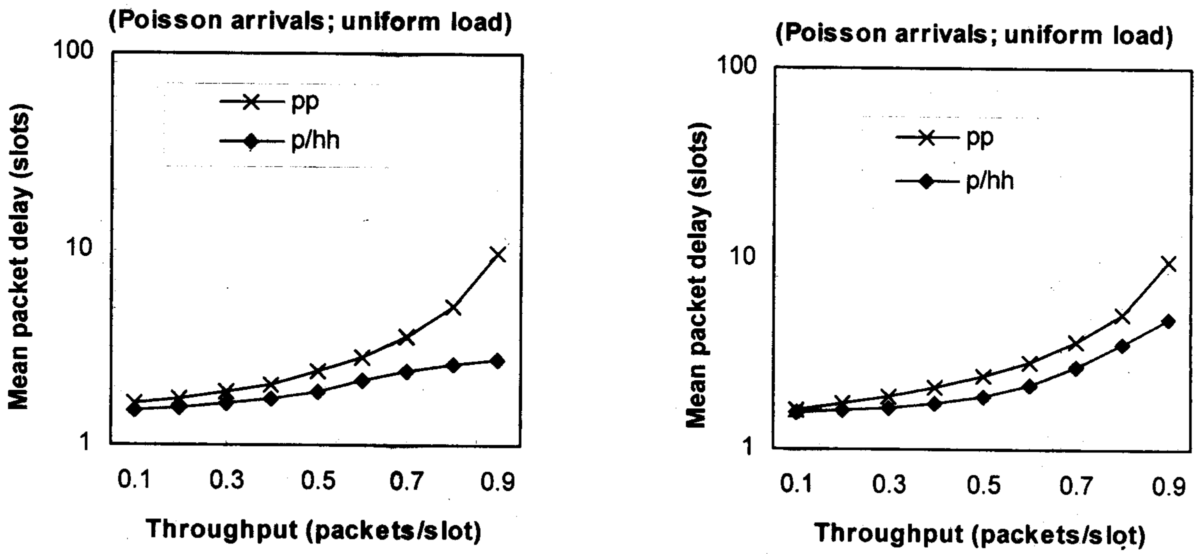
where p_i is the channel access probability at station i $(i = 1, 2, \cdots, N)$, and $\lambda = \sum_{j=1}^{N}\lambda_j$ is the total offered traffic to the network in packets/slot.

The derivation of Equation (1) can be found in (Mukherjee, 1990). Note that (1) is an approximate formula for p_i, and is valid for Poisson arrival processes.

To simplify the simulation model, the following assumptions are made throughout the simulation experiments:

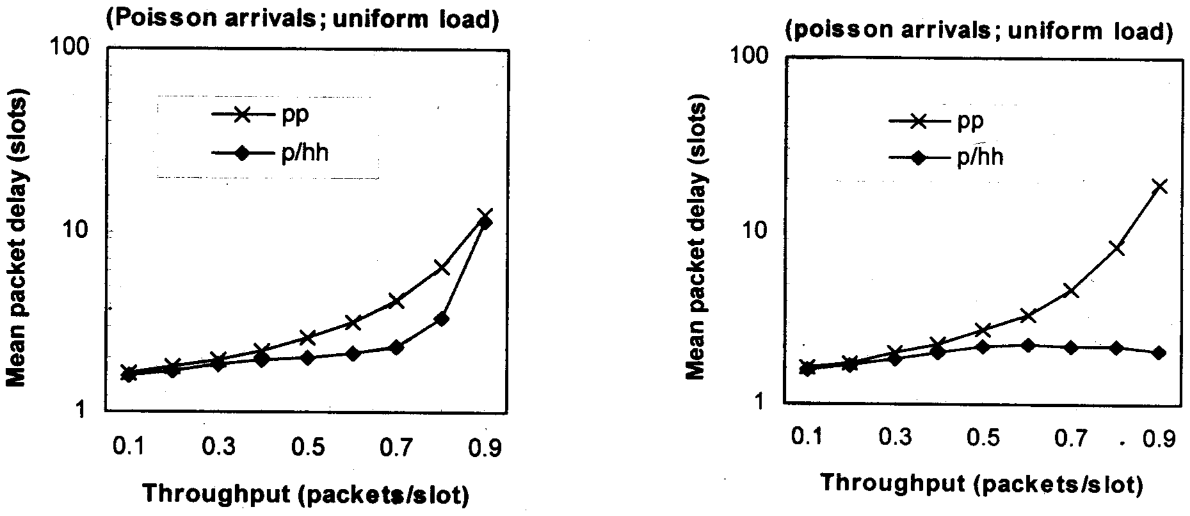
A1. **Traffic:** All traffic is assumed to be of asynchronous type.

A2. **Packet generation:** Streams of data packets generated at stations are modelled as independent Poisson processes, assuming that maximum one packet can arrive during a slot time.



(a) Station 1

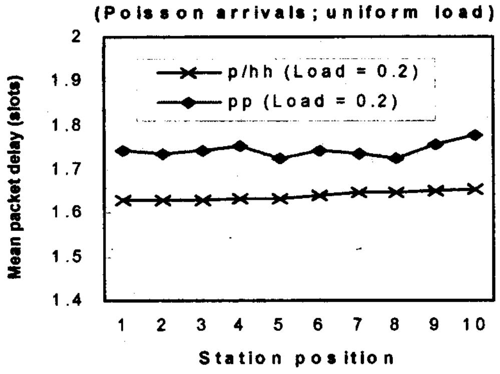
(b) Station 5



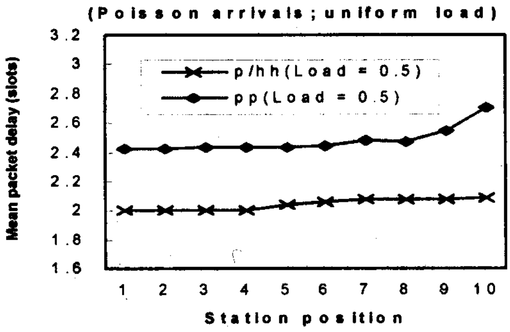
(c) Station 9

(d) Station 10

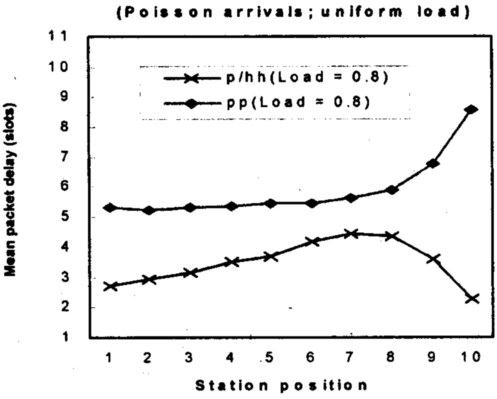
Fig. 6: Mean Packet Delay Versus Throughput Performance of Selected Stations. Comparison of the P_i-Persistent/HH Protocol (P/hh) and the P_i-Persistent Protocol (PP)



(a) $\lambda = 0.2$



(b) $\lambda = 0.5$



(c) $\lambda = 0.8$

Fig. 7: Mean Packet Delay Versus Station Position Performance With *Uniform* Loading. Comparison of The P_i-Persistent/HH Protocol (P/hh), and P_i-Persistent Protocol (PP)

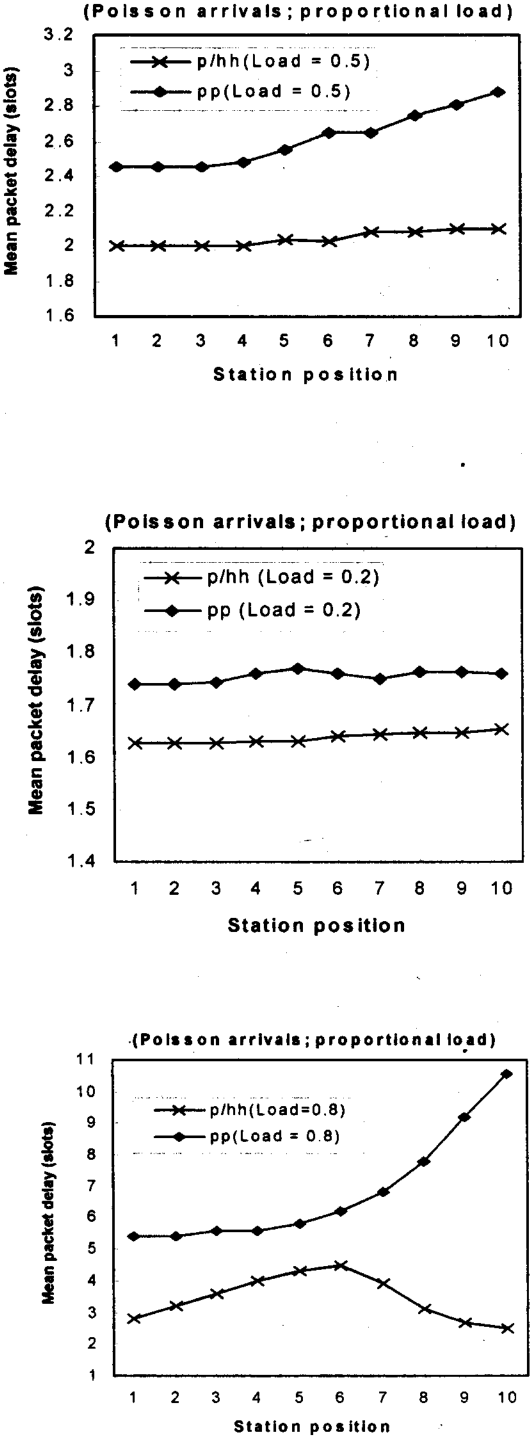


Fig. 8: Mean Packet Delay Versus Station Position Performance with Proportional Loading. Comparison of the P_i-Persistent Protocol (P/hh), and P_i-Persistent Protocol (PP)

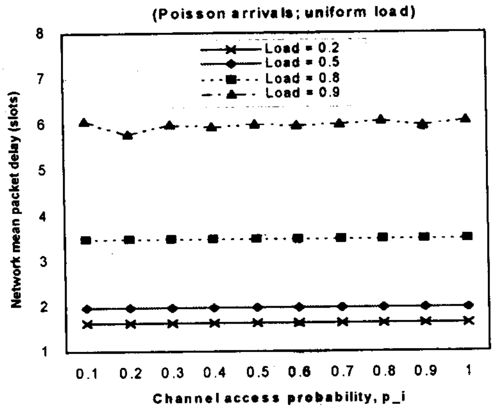


Fig. 9: Effect Of $\{p_i\}$ on Network-Wide Mean Packet Delay of P_i-Persistent/HH Protocol

A3. **Packet size:** Packets are of fixed length. The time axis is divided into slots of equal length, and the transmission of one packet takes one slot time.

A4. **Buffer size:** Each station in the network has a large buffer, modelled as a buffer of infinite size, to store packets. This assumption means that packets cannot be lost due to buffer overflows when the system is under manageable input loads.

A5. **Processing delay:** The station's latency or processing delay is negligible if compared with slot duration. The processing of control data contained in the header can be done in a fraction of a slot time.

A6. **Destination addresses:** We assume that the packets arriving at a station are uniformly destined to $N-1$ other stations in the network.

A7. **Stations spacing:** The stations can be arbitrarily spaced on the bus.

A8. **Analysis:** We study the network performance under steady state conditions.

The performance measures of interest are the mean packet delay and fairness for individual stations and the overall network. The mean packet delay is defined as the average time (measured in slots) from the moment the packet is generated at a given station and joins its local queue (L-queue) until the packet is delivered to its destination. This includes the waiting time in the L-queue, and all T-queues it visits before it is completely shipped out from the bus. It should be noted that the minimum possible mean packet delay is 1.5 slots (half a slot for residual life time of the current slot and one slot for packet transmission). We use simulation to measure the mean packet delay at the stations.

Simulation Results: In this section, we present the experimental results obtain from simulation runs for p_i-persistent/HH protocol (abbreviated to p/hh) and the p_i-persistent protocol ((abbreviated to pp).

Network mean packet delay versus throughput performance ($N = 10$ stations, Poisson arrivals, uniform loading): In this experiment we consider a network with $N = 10$ stations operating under uniform loads (in which the packet arrival rate is the same for all stations). The network mean packet delay versus throughput performance under p_i-persistent/HH protocol (p/hh) and p_i-persistent protocol (pp) is shown in Fig. 5. The corresponding values for a slotted M/D/1 queue (the ideal work conserving FIFO service algorithm) are also shown for comparison. As is evident from the graphs, the p/hh protocol provides significantly better performance than the p_i-persistent protocol. Note that the network-wide mean packet delay of p/hh scheme match the value for the ideal FIFO queue. The main conclusion we may draw from Fig. 5 is that when we employ p/hh instead of pp protocol for a 10-station network, we obtain a significant improvement in mean packet delay across the network, specially for moderate to high loads.

Mean packet delay versus throughput performance of selected stations ($N = 10$ stations, Poisson packet arrivals, uniform loading): In Fig. 6 (a) - (d), we plot the mean packet delay versus throughput of Stations 1, 5, 9, and 10, respectively, under the p/hh and pp schemes.

We observe that the mean packet delay performance of stations 1, 5, 9, and 10 under the p/hh scheme is better (in the sense that they experience lower mean delay) than the pp scheme. The main conclusion we may draw from Fig. 6 (a) – (d) is that stations 1, 5, 9, and 10 under p/hh scheme achieve a substantial improvement in the mean packet delay performance in comparison with under pp scheme, specially for moderate to high loads.

Mean packet delay versus station position ($N = 10$ stations, Poisson packet arrivals): The mean packet delay at each station for a 10-station network with uniform loading for 0.2, 0.5, and 0.8 is shown in Fig. 7 (a), (b), and (c), respectively. The corresponding results for proportional loading are shown in Fig. 8 (a) - (c). By looking Fig. 7 and 8 we find that the stations under the p/hh scheme have lower mean packet delay than under the pp scheme for $\lambda = 0.2$, 0.5, and 0.8.

The influence of traffic pattern on mean packet delay versus station position performance is insignificant under p/hh scheme. In contrast, under the pp scheme delay on average than the corresponding stations under uniform loads, specially as we move further downstream along the bus.

The main conclusion we can draw from Fig. 7 and 8 is that individual station's mean packet delay performance under the p/hh scheme significantly better than under the pp scheme for a 10-station network. The influence of traffic pattern on mean packet delay at each station is found to be insignificant under p/hh scheme, but under the pp scheme, the traffic pattern has some influence on mean packet specially at higher loads.

Effect of $\{p_i\}$ on network mean delay Station 1($N=10$ stations, Poisson arrivals, uniform loading): The effect of $\{p_i\}$ on network mean packet delay is shown in Fig. 9. We observe that the influence of different sets of $\{p_i\}$ on network mean packet delay is insignificant.

In Fig. 9, we plot the network mean packet delay versus p_i $(p_i \in [0.1,1])$ for $\lambda = 0.2$, 0.5, 0.8, and 0.9. As our simulation results show, the variability in network mean packet delay is negligible for $\lambda = 0.2$, 0.5, 0.8, and is below 5% for $\lambda = 0.9$. From Fig. 9, we can draw the conclusion that the network-wide mean packet delay is almost independent of p_i. This is an attractive feature which comes from the protocol's work conserving property (ie., the slots are never wasted if a station has packets for transmission).

Table 2: Operations Executed at Station *i* During one Slot time. Δ is the Processing Time of Slot Header at the Station

In-slot at time t_0	Out-slot at time $t_0 + \Delta$
1) PH	PH (it will carry a packet from T-queues)
2) HH	PH (if station *i* transmits from its L-queue with p_i)HH (otherwise)
3) Empty	Empty (if both L- and T-queues are empty) PH (if station *i* transmits from its L-queue with p_i) HH (otherwise)

Conclusion

In this paper we extended the original p_i-persistent protocol by including a technique of slot pre-use that not only eliminates the possibility of 'wasting' slots but also significantly improve the delay versus throughput performance, as well as can achieved fairness without much increase in latency or complexity at the stations. Through a number of simulation experiments we compared the performance of the p_i-persistent/HH protocol with that of pi-persistent protocol, assuming the average packet delay as the fairness criterion for selecting access probabilities P_i. Under the assumptions made, the results showed that the pi-persistent/HH protocol can offers lower mean delay and better fairness than the p_i-persistent protocol, specially for low to moderate loads. We have also found that the network-wide mean packet delay is almost the same as it would be for a centralised single server queue, and the network mean packet delay is not very sensitive to p_i. These features come from the protocol's work conserving property, in which slots are never wasted if a station has packets for transmission.

In spite of the improved delay-throughput performance of the p_i-persistent/HH protocol reported in this paper, further research has to be done to analyse the latency or processing delay at the stations.

References

Filipiak, J., 1989. Access protection for fairness in a distributed queue dual bus metropolitan area network, In Proceedings IEEE ICC '89, Boston, 635-639.

Hahne, E. L., Choudhury, A. K. and Maxemchuk, N. F. 1990. Improving the fairness of distributed queue dual bus networks, In Proceedings INFOCOM '90, San Francisco, 1990. 175-184.

IEEE Standard 802.6/D15, 1990. IEEE standards for local and metropolitan area networks: distributed queue dual bus (DQDB) subnetwork of a metropolitan area network (MAN).

Mathar, R. and Pawlikowski, K., 1997. Performance analysis of the p_i-persistent protocol in unidirectional bus networks, Computer Communications, 20: 535-543.

Manjunath, D. and Molle, M. L., 1995. The effect of bandwidth allocation policies on delay in unidirectional bus networks, IEEE J. on Selected Areas in Communications, 13: 1309-1323.

Mukherjee, B., 1990. Algorithm for the p_i-persistent protocol for high-speed fiber optic networks, Computer Communications, 13: 387-398.

Mukherjee, B., 1991. Performance of a dual-bus fiber optic network operating under a probabilistic scheduling strategy, Performance Evaluation, 12: 127-139.

Mukherjee, B. and Kamal, A. E., 1994. The continuation-bit approach and the p_i-persistent protocol for scheduling variable-length messages on slotted, high-speed, fiber optic LANs/MANs, Computer Networks and ISDN Systems, 26: 721-744.

Mukherjee, B., Lantz, A.C., Matloff, N.S. and Banerjee, S., 1991. Dynamic control and accuracy of the p_i-persistent protocol using channel feedback, IEEE Transactions on Communications, 39: 887-897, 1991.

Mukherjee, B. and Meditch, J. S., 1988. The p_i-persistent protocol for unidirectional broadcast bus networks, IEEE Transactions on Communications, 36: 1277-1286.

Pawlikowski, K. and Yau, V., 1992. On automatic partitioning runtime control and output analysis methodology for massively parallel simulation, In Proceedings of the European Simulation Symposium, ESS '92, 135-139, Dresden, Germany.

Pawlikowski, K., Yau, V. and McNickle, D., 1994. Distributed stochastic discrete event simulation in parallel time streams, In Proceedings of the 1994 Winter Simulation Conference, Orlando, 723-739.

Sarkar, N. and Pawlikowski, K., 2001. A simulation study of the p_i-persistent protocol in unidirectional bus networks, In Proceedings of the Int. Conference on Computer and Information Technology, ICCIT2000, 84-88, NSU, Dhaka, 25-26.

Sarkar, N., 1996. Probabilistic Scheduling Strategies for Unidirectional Bus Networks, MSc thesis, Uni. of Canterbury, Christchurch, New Zealand.

Tobagi. F.A, Borgonovo, F. and Fratta, L., 1983. Expressnet: A high-performance integrated-services local area network, IEEE J. on Selected Areas in Communications, SAC-1: 898-913, 1983.

Tseng, C-W. and Chen, B., 1983. D-Net, A new scheme for high data rate optical local area networks, IEEE Transactions on Communications, 36: 1277-1286, 1983.