Full Utilization, Fairness and Bounded Access Delay on High Speed Bus Networks *

Angela L. Chiu and Robert G. Gallager
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139

Abstract

The purpose of this paper is to understand the relationship between utilization, fairness and access delay in high speed slotted bus networks. We illustrate this relationship by means of a protocol called FUFA (fully utilized and fair). We define full utilization, and fairness precisely, and show that both are achieved together in the FUFA protocol. In addition, the protocol provides bounded access delay that is linear in the round trip propagation delay, and at most a constant away from its minimum possible value for any bus protocol that is both fully utilized and fair. The main idea is that each station takes account of the idle slots propagated previously to interpret the information from downstream (i.e., estimated aggregate number of data segments in queue downstream and estimated number of active downstream stations). This allows the active downstream stations to be served in a round robin fashion according to the updated information.

Key words: dual bus network, propagation delay, full utilization, fairness, bounded access delay

1 Introduction

High speed slotted bus networks [6, 7, 9, 11] are attractive candidates for metropolitan area networks (MANs), backbone networks for LANs, and feeder networks for ATM. One such network is the well known distributed queue dual bus (DQDB) network with bandwidth balancing (the IEEE 802.6 standard for MANs) [5]. Although this has not been a great commercial success, it has many interesting features, and has generated a large literature of suggested modifications and improvements. See [8] for a good survey. Currently, there are attempts to build slotted bus networks at much higher speeds, in the multiple Gbps range [1, 3, 11].

In all of these network proposals, there are tradeoffs between utilization, fairness, access delay, overhead, and complexity. Our objective here is to understand the relationship between utilization, fairness, and access delay. We illustrate this relationship by means of a protocol called FUFA (fully utilized and fair) that shows that full utilization, fairness, and bounded access delay are in fact compatible with each other. The protocol has somewhat greater overhead and complexity than most of the above proposals, and thus is not intended as a practical protocol for very high speed applications, but rather as a means to demonstrate the above compatibility and to understand the kind of inter-station communication required to achieve it.

We define full utilization for a slotted bus network as the property that a station with data segments to send never releases an idle slot unless that idle slot is used by some further downstream station. As an example of an algorithm with full utilization, consider the greedy algorithm, i.e., the algorithm in which each station with traffic to send fills each passing idle slot. The algorithm is not fair, since the station at the head-end can monopolize the bus. Similarly there is no bound on access delay, since downstream stations can wait forever for an idle slot. Therefore, in order for an algorithm to be fully utilized and be fair or have bounded access delay, it must be non-greedy.

We now show that for a non-greedy algorithm to be fully utilized, it must have a certain kind of feedback information. Suppose the bus is idle and that at a certain time, an upstream station receives an unlimited number of data segments to send and that at close to the same time, a downstream station receives a single segment to send. The downstream station must send some type of request upstream (otherwise the upstream node, in ignorance of the downstream segment, must

*This research is supported in part by ARPA under grant MDA972-92-J-1038 and in part by Army Research Office under grant ARO-DAAH04-94-G-0221 (ASSERT).

0-8186-7453-9/96 $05.00 © 1996 IEEE

fill every idle slot to achieve full utilization.) When the upstream station receives the request, in order to make a decision on whether to propagate the next idle slot without wasting it, it needs to check how many idle slots it has already propagated during the last round trip propagation delay to the downstream station. These are the idle slots that the downstream station would receive before the next slot arrives there. Although many existing protocols [3, 5-7, 9-11] (including the modifications of DQDB surveyed in [8]) can provide high utilization, none of them incorporates these recently transmitted idle slots in their decision, and thus none achieves full utilization as defined here. In the FUFA protocol, each station takes account of the idle slots propagated previously to interpret the information from downstream (i.e., estimated aggregate number of data segments in queue downstream and estimated number of active downstream stations). This allows the active downstream stations to be served in a round robin fashion according to the updated information. We will see later that the FUFA protocol actually achieves full utilization.

Besides the full utilization property, FUFA provides fairness in the following sense. If a subset of the stations are very active and the rest are idle, the FUFA protocol distributes idle slots fairly among active stations in a round robin fashion, and the cycle is established within one round trip propagation delay between the most upstream and most downstream active stations.

A protocol is defined to have the bounded access delay property if the access delay of the first data segment in queue at each station is bounded. We show that the FUFA protocol provides a bounded access delay that is linear in the round trip propagation delay, and only a constant away from its minimum value for any bus protocol that is both fully utilized and fair.

The remainder of this paper is organized as follows. In section 2 we describe the basic dual bus topology. In section 3 we define what we mean by full utilization, fairness, and bounded access delay. In section 4, we describe the FUFA protocol. We start with the basic concept, and then give a full description of the FUFA protocol, followed by some basic properties of the protocol. In section 5 we state the full utilization, fairness and bounded access delay properties of FUFA. Finally we conclude our results in section 6.

2 Basic dual bus network

The dual bus topology we consider here is identical to that used in DQDB (see Figure 1).

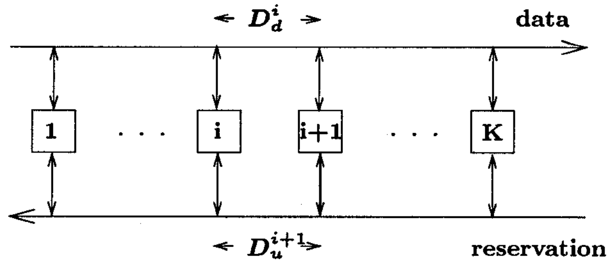

Figure 1: dual bus topology

The two buses support unidirectional communications in opposite directions. Stations are connected to both buses and communicate by selecting the proper bus. A special unit at the head-end of each bus generates one slot at each unit of time. The stations are numbered from left to right as stations 1, 2, ..., K. Because of the symmetry of the dual bus topology, we can consider only transmission on one bus. The bus from station 1 to station K is used to transfer data and is referred to as the data bus or downstream bus. The bus from station K to station 1 is used to make reservations and is referred to as the reservation bus or upstream bus. Therefore station 1 is the most upstream station, and station K is the most downstream station. For each station $i \in \{1, 2, ..., K\}$, denote D_u^i as the propagation delay measured in slots between station i and its upstream station $i-1$ and D_d^i as the propagation delay between station i and its downstream station $i+1$, where D_u^i and D_d^i are integers. (The integer assumption is for notational simplicity. The generalization to the non-integer case is in [2].) Each station has a local FIFO (first-in-first-out) queue to store data segments from local users while these segments wait for assignment to appropriate idle slots on the data bus. Note that the protocol also works in principle on a folded bus, where one fold of the bus can be viewed as the date bus and the other fold as the reservation bus.

3 Definitions of full utilization, fairness, and bounded access delay

Definition 1: A protocol has *full utilization* if whenever a station with a non-empty queue propagates an idle slot, that idle slot is used by some further downstream station.

That is, full utilization means that an idle slot is never wasted. A slot must be used if it passes any station with a non-empty queue. It is shown in the Appendix that any protocol with full utilization provides the minimum system queueing delay among all protocols.

Definition 2: Let $S_n = \{i_1 < i_2 < ... < i_n\}$ be a set of some n stations that have been "very active" since t_0, where being "very active" is defined in section 5.2. A protocol is *fair* if each station $i_k \in S_n$, $k = 1, ..., n$, starting from $t_0 + \sum_{h=i_1}^{i_k-1} D_d^h$, transmits one data segment in every n time slots for as long as S_n remains the set of "very active" stations and all the other stations remain idle (i.e., with empty queues).

Definition 3: Let P_i be a data segment at station i, $i \in \{1, 2, ..., K\}$, denote t_a^i as the time that P_i becomes the first segment in the queue, and t_d^i as the time that P_i departs from the queue. Then an algorithm has the bounded access delay property if for each i, there is some constant B_i such that $t_d^i - t_a^i \leq B_i$ for each P_i .

Note that full utilization and bounded access delay, as defined here, are for general, arbitrarily varying traffic conditions.

4 Fully utilized and fair (FUFA) protocol

4.1 Basic Concept

Since all the idle slots on the data bus are generated from the head-end, the most upstream station, station 1, has first access to idle slots. The basic concept of the protocol is to give equal access to all the stations, according to the most updated information available through the reservation bus. In particular, according to the information available, each station estimates the number of active downstream stations and uses a counter to serve them in a round robin fashion. The novel feature of this protocol is that each station takes account of the idle slots propagated previously to interpret the information from downstream (i.e., estimated aggregate number of data segments in queue downstream and estimated number of active downstream stations).

4.2 Parameters

The following parameters are used in the protocol. At time t, the information available at station $i \in \{1, 2, ..., K\}$ is as follows:

- $n_i(t)$: number of idle slots propagated by station i during the past $D_u^{i+1} + D_d^i$ time slots. $D_u^{i+1} + D_d^i$ is assumed to be integer here. See Figure 2.
- $Q_i(t)$: number of data segments in the FIFO queue of station i.

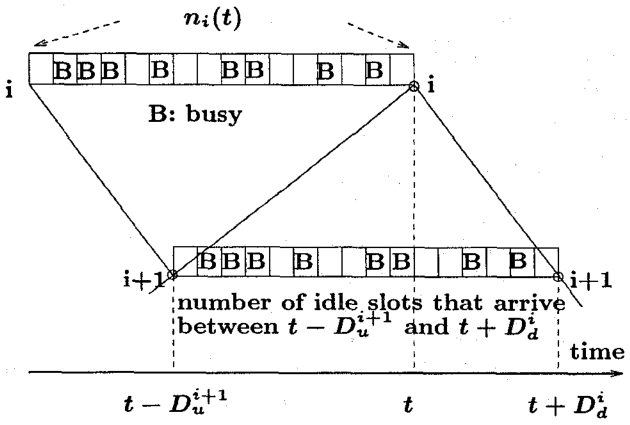

Figure 2: the idle slots counted in $n_i(t)$ arrive at station $i+1$ between $t - D_u^{i+1}$ (the time of the most recent feedback from $i+1$) and $t + D_d^i$ (the time at which the next slot from i reaches $i+1$)

- $I_i(t)$: indicator function whether station i is active or not, i.e.,

$$I_i(t) \triangleq 1 \text{ if } Q_i(t) > 0, \ 0 \text{ otherwise.}$$

The information sent by station i to the upstream station $i-1$ is,

- $M_i(t)$: estimated current number of active stations downstream from station i (including i).
- $m_i(t)$: estimated aggregate number of data segments in the FIFO queues of all stations downstream from station i (including i).

4.3 Distributed algorithm

The algorithm is described in discrete time with the assumption of zero processing delay. ([2] covers the case with non-discrete time and non-zero processing delay.) At time t, the information available at station i, $i \in \{1, 2, ..., K\}$, is $Q_i(t)$, $I_i(t)$, and $n_i(t)$. Before station i receives any information from downstream, it uses idle slots whenever it can with round robin counter $C_i(t)$ being 0. This is also the algorithm for the most downstream station K at all t, and,

$$m_K(t) = Q_K(t),$$
$$M_K(t) = I_K(t),$$
$$C_K(t) = 0.$$

In general, for time t, and station $i \in \{1, 2, ..., K-1\}$, the algorithm runs as follows:

1. Receive $m_{i+1}(t - D_u^{i+1})$ and $M_{i+1}(t - D_u^{i+1})$ sent by station $i+1$ at $t - D_u^{i+1}$.

2. Calculate $m_i^s(t)$ and $M_i^s(t)$ as follows,

$$m_i^s(t) = [m_{i+1}(t - D_u^{i+1}) - n_i(t)]^+,$$
$$M_i^s(t) = \min\{M_{i+1}(t - D_u^{i+1}),\ m_i^s(t)\}. \quad (1)$$

Here $m_i^s(t)$ is an estimate of the aggregate number of queued data segments strictly downstream of station i that will be seen by a downstream slot passing i at time t, and $M_i^s(t)$ is a similar estimate of the number of active stations strictly downstream from station i.

3. Update the counter and make a decision as follows:

$$C_i(t) = \min\{\tilde{C}_i(t-1),\ M_i^s(t)\}, \quad (2)$$

- if $I_i(t) = 1$, the passing slot is idle, and $C_i(t) = 0$, then occupy it, and set $\tilde{C}_i(t)$ to $K - i$,
- if $I_i(t) = 1$, the passing slot is idle, and $C_i(t) > 0$, then propagate it, and set $\tilde{C}_i(t)$ to $C_i(t) - 1$,
- if $I_i(t) = 1$, and the passing slot is busy, then propagate it, and set $\tilde{C}_i(t)$ to $C_i(t)$,
- if $I_i(t) = 0$, then propagate the passing slot, and set $\tilde{C}_i(t)$ to $K - i$.

4. Obtain $M_i(t)$ and $m_i(t)$ as below, and send them to station $i - 1$,

$$m_i(t) = Q_i(t) + m_i^s(t),$$
$$M_i(t) = I_i(t) + M_i^s(t).$$

The core of the algorithm is the second step, where station i uses the extra piece of information $n_i(t)$ to update $m_{i+1}(t - D_u^{i+1})$ and $M_{i+1}(t - D_u^{i+1})$. As an example, consider Figure 3. At time t, station i receives

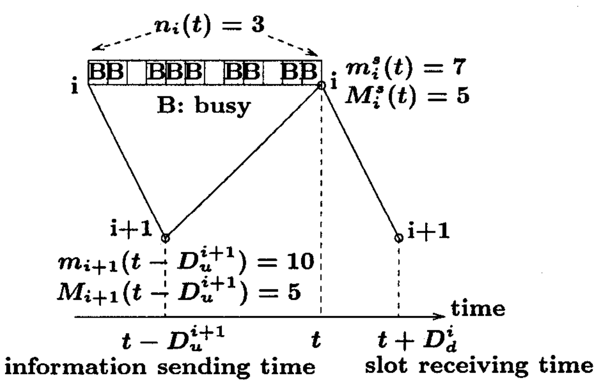

Figure 3: an example for the information updating step

the information that there are 10 data segments and 5 active stations downstream ($m_{i+1}(t - D_u^{i+1}) = 10$ and $M_{i+1}(t - D_u^{i+1}) = 5$). On the other hand, station i needs to take consideration of $n_i(t) = 3$. In the absence of new arrivals, station i knows that the next slot will see 7 data segments remaining at the queues of at most 5 downstream stations (i.e., $m_i^s(t) = 7$ and $M_i^s(t) = 5$). Consider the same example except that $n_i(t) = 7$. Again, in the absence of new arrivals, station i knows that the next slot will see only 3 data segments remaining at the queues of at most 3 downstream stations (i.e., $m_i^s(t) = 3$ and $M_i^s(t) = 3$).

In order to guarantee the full utilization property, the decision made on idle slots should be based solely on the information received, not the probabilistic estimates of future arrivals. As a consequence, downstream stations still suffer from propagation delays. In order to compensate for this disadvantage, the protocol is designed with a bias towards downstream stations in the updating equation (1), where $M_i^s(t)$ takes its maximum possible value in the absence of new arrivals. This can be seen in the second example above. The 3 data segments remaining can be distributed at one station, or at most 3 stations, and $M_i^s(t) = 3$. On the other hand, the estimate $m_i^s(t)$, the aggregate number of data segments downstream, is the true value in the absence of new arrivals. This ensures the full utilization property.

4.4 Properties of the FUFA protocol

In order to describe some basic properties, we first define the following parameters, for time t, s, and station i, $k \in \{1, 2, ..., K\}$,

- $A_i[t,\ t+s]$: number of arrivals at station i during the interval $[t,\ t+s)$; This is the time interval starting at the t-th time slot and ending right before the $(t+s)$-th time slot.
- $n_i[t,\ t+s]$: number of idle slots that station i propagates during $[t,\ t+s)$; Thus $n_i(t)$, defined earlier, is $n_i[t - D_u^{i+1} - D_d^i,\ t]$.
- $N_i[t,\ t+s]$: number of idle slots that station i uses during $[t,\ t+s)$; thus,

$$n_{i+1}[t,\ t+s] = n_i[t - D_d^i,\ t+s - D_d^i] - N_{i+1}[t,\ t+s].$$

- $\tau_k^i(t)$: time when an upstream slot passing station i at t passed station $k > i$, or the time when it will reach station $k < i$, i.e.,

$$\tau_k^i(t) \triangleq t - \sum_{h=i+1}^{k} D_u^h, \quad \text{or } \tau_k^i(t) \triangleq t + \sum_{h=k+1}^{i} D_u^h.$$

- $T_k^i(t)$: time when a downstream slot passing station i at t reaches station $k > i$,

$$T_k^i(t) \triangleq t + \sum_{h=i}^{k-1} D_d^h.$$

See Figure 4 for an illustration.

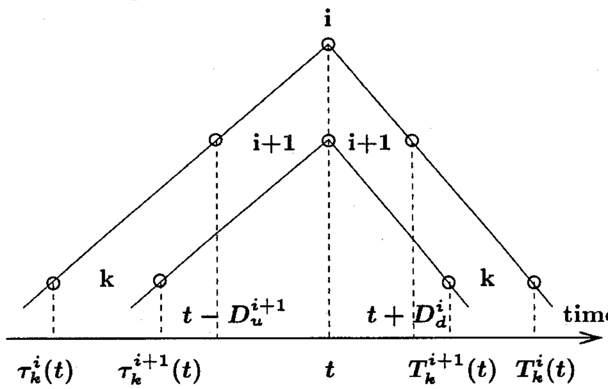

Figure 4: illustration of $\tau_k^i(t)$ and $T_k^i(t)$ for $k > i$

Proposition 1 to 5 below are valid for all t, s, and all $i \in \{1, 2, ..., K-1\}$. The proofs for all but 3 are straightforward consequences of the definitions. (See [2] for details.)

Proposition 1: $M_i^s(t) \geq C_i(t) \geq 0$.

Proposition 2: $n_{i+1}[t,\ t+s] = n_i[t - D_d^i,\ t + s - D_d^i] - Q_{i+1}(t) - A_{i+1}[t,\ t+s] + Q_{i+1}(t+s)$.

Proposition 3: $\sum_{k=i+2}^{l} N_k[T_k^{i+1}(t - D_u^{i+1}),\ T_k^i(t)] = n_{i+1}[t - D_u^{i+1},\ t + D_d^i] - n_l[T_l^{i+1}(t - D_u^{i+1}),\ T_l^i(t)]$.

Proposition 4: If station i has a non-empty queue and propagates all the idle slots arriving during $[t,\ t+s)$, then

$$C_i(t) - C_i(t+s) \geq n_i[t,\ t+s] \geq 0.$$

Proposition 5: $M_i^s(t) \geq [M_{i+1}(t - D_u^{i+1}) - n_i(t)]^+$.

Due to the space limitation, we only state the lemmas and theorems here. Readers are referred to [2] for proofs. First, we state two lemmas here and three later in section 5 which are useful in the proofs of full utilization, fairness and bounded access delay.

Lemma 1: For any given $i \in \{1, 2, ..., K-1\}$ and any t,

$$m_i^s(t) \leq \sum_{k=i+1}^{K} Q_k(T_k^i(t)).$$

Remark. The main point of Lemma 1 is as follows. At time t, station i decides whether to use or propagate a passing slot based on the estimated number $m_i^s(t)$ of data segments that will be waiting at downstream stations $i+1, i+2, ..., K$. Then the aggregate number of data segments that the slot sees when it passes at each downstream station $k \in \{i+1, ..., K\}$ at $T_k^i(t)$ should be at least as large as the estimation $m_i^s(t)$. It might be larger due to the new arrivals at all the downstream stations $i+1, i+2, ..., K$. Thus, the lemma is useful in the proof of full utilization.

Lemma 2: For all t, and all $i \in \{1, 2, ..., K-1\}$, the following two statements are true:

$$m_i^s(t) > 0 \text{ iff } M_i^s(t) > 0,$$
$$m_i(t) > 0 \text{ iff } M_i(t) > 0.$$

Remark. Lemma 2 formalizes the intuitive fact that the estimated aggregate number of downstream data segments is positive if and only if the estimated number of active downstream stations is positive.

5 Full utilization, fairness and bounded access delay

5.1 Full utilization

Theorem 1: The protocol FUFA has full utilization according to Definition 1 in section 3.

5.2 Fairness

The fairness property of the FUFA protocol depends on Lemmas 3 and 4 below.

Lemma 3: For any given $i \in \{1, 2, ..., K-1\}$, any t,

$$m_i^s(t) \geq [\ \sum_{k=i+1}^{K} (Q_k(T_k^i(t)) - A_k[\tau_k^i(t),\ T_k^i(t)])\ ]^+.$$

Remark. The main point of Lemma 3 is as follows. At time t, station i decides whether to use or propagate the passing slot based on the estimated number $m_i^s(t)$ of data segments that will be waiting at downstream stations $i+1, i+2, ..., K$. Then the aggregate number of data segments that the slot sees when it arrives at each downstream station $k \in \{i+1, ..., K\}$ at $T_k^i(t)$ should be no larger than that estimate $m_i^s(t)$ plus all the new arrivals.

Lemma 4: For any $i \in \{1, 2, ..., K-1\}$, and any t,

$$\sum_{k=i+1}^{K} I_k(\tau_k^i(t)) \geq M_i^s(t).$$

Remark. Lemma 4 is intuitive based on the fact that taking extra idle slots into consideration in the information updating can only reduce the estimated number of active downstream stations among $i+1$, $i+2$, ..., K.

Definition 4: The set $S_n = \{i_1 < i_2 < ... < i_n\}$ has been "very active" since t_0 if, for each $t \geq t_0$, and each station $i_k \in S_n$, the queue length at time $T_{i_k}^{i_1}(t)$ exceeds the number of new arrivals in the interval from $T_{i_k}^{i_1}(t)$ back to the past round trip delay between i_k and i_1 (i.e., some packet in queue at $\tau_{i_k}^{i_1}(t)$ is still in queue at $T_{i_k}^{i_1}(t)$), i.e.,

$$Q_{i_k}(T_{i_k}^{i_1}(t)) > A_{i_k}[\tau_{i_k}^{i_1}(t),\ T_{i_k}^{i_1}(t)].$$

See Figure 5 for an illustration of the timing.

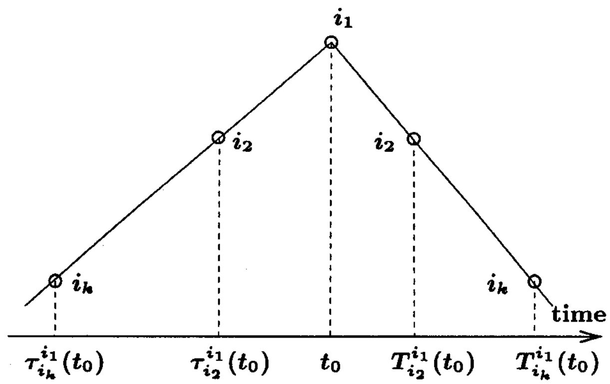

Figure 5: illustration of $\tau_{i_k}^{i_1}(t_0)$ and $T_{i_k}^{i_1}(t_0)$ for $i_k \in S_n$

Theorem 2: The protocol FUFA is fair according to Definition 2 in section 3.
Remark. This means that the network converges to a fair state under the condition in Definition 2 in at most one round trip propagation delay between the most upstream and the most downstream active stations.

5.3 Bounded access delay

In order to prove the bounded access delay property, we first state Lemma 5 as follows:
Lemma 5: For any $i \in \{1,\ 2,\ ...,\ K\}$, any segment P_i, any k satisfying $1 \le k < i$, and any t, denote t_a^i and t_d^i as the time that P_i becomes the first segment in the queue and departs from the queue, respectively. Let $t_k \triangleq \tau_k^i(t_a^i)$ be the time when this information propagates to station k, and $r_k \triangleq \min\{t \ge t_k \mid C_k(t) = M_k^s(t)\}$ be the time that counter $C_k(t)$ is set to $M_k^s(t)$ by (2) for the first time since t_k. Then we have

$$N_k[r_k,\ T_k^i(t_d^i) + 1] = 0.$$

Proof: See the proof of Statement 2 in [2].
Remark. Lemma 5 concerns the number of slots that a station k can occupy before propagating an idle slot to a downstream station i. If station k is active at t_k, it can occupy at most one slot before propagating the idle slot used by i, and if station k is idle at t_k (i.e., $r_k = t_k$), it occupies no slot before propagating the idle slot used by i. Hence,

$$N_k[t_k,\ T_k^i(t_d^i) + 1] \le I_k(t_k). \qquad (3)$$

Theorem 3: The protocol FUFA has the bounded access delay property; for each station $i \in \{1,\ 2,\ ...,\ K\}$ and each first data segment P_i in queue, the access delay $t_d^i - t_a^i \le B_i = \sum_{k=1}^{i-1}(D_u^{k+1} + D_d^k) + K - 1$.
Proof: Note that

$$\begin{aligned} t_d^i - t_a^i \ \le\ & \sum_{k=1}^{i-1}(D_u^{k+1} + D_d^k) + n_i[T_i^1(t_1),\ t_d^i + 1] \\ & + \sum_{k=1}^{i-1} N_k[T_k^1(t_1),\ T_k^i(t_d^i) + 1] \end{aligned}$$

Based on (3), for each $k \in \{1,\ 2,\ ...,\ i-1\}$, we have

$$N_k[T_k^1(t_1),\ T_k^i(t_d^i) + 1] \le N_k[t_k,\ T_k^i(t_d^i) + 1] \le I_k(t_k).$$

On the other hand,

$$\begin{aligned} n_i[T_i^1(t_1),\ t_d^i + 1] \ \le\ & n_i[t_i,\ t_d^i + 1] \le C_i(t_i) \le M_i^s(t_i) \\ \le\ & \sum_{k=i+1}^{K} I_k(t_k) \end{aligned}$$

where the second and last inequality are based on Proposition 4 and Lemma 4 with $t_k \triangleq \tau_k^i(t_a^i)$ for $k > i$, respectively. Combining all the inequalities above, we have

$$t_d^i - t_a^i \le \sum_{k=1}^{i-1}(D_u^{k+1} + D_d^k) + \sum_{k=1,\ k \ne i}^{K} I_k(t_k)$$

where $\sum_{k=1,\ k\ne i}^{K} I_k(t_k)$ is bounded by $K - 1$. ■

Remark. Theorem 3 states that the access delay for the first data segment in queue at station i is upper bounded by $\sum_{k=1}^{i-1}(D_u^{k+1} + D_d^k) + K - 1$, the round trip propagation delay between station i and the most upstream station 1, plus a constant $K - 1$, where K is the total number of stations in the network. For any general bus protocol that is both full utilized and fair, let B_i^G be an upper bound on the access delay for the first data segment at station i. Due to the full utilization property, idle slots are propagated by a non-empty station based on only the information that has been received. Therefore, the access delay of the first data segment at station i can be as large as the round trip propagation delay between station i and the most upstream station, station 1, i.e.,

$$B_i^G \ge \sum_{k=1}^{i-1}(D_u^{k+1} + D_d^k) \qquad (4)$$

Besides the round trip propagation delay in (4), the round robin cycle under the condition in the definition of "fairness" can result in extra delay for a station to get access to the idle slot. Consider the following scenario. The most upstream station 1 is always active with a long queue. All the other stations $i > 1$ stay idle until $\tau_i^1(t)$, when many data segments arrive at the same time. Based on the full utilization property, station 1 will not propagate any idle slot until time t when the information of downstream stations being active first arrives. Therefore, idle slots will not arrive at station $i > 1$ earlier than $T_i^1(t)$, a round trip propagation delay away from $\tau_i^1(t)$. Hence, (as already seen in (4)),

$$B_i^G \geq \sum_{k=1}^{i-1}(D_u^{k+1} + D_d^k)$$

Notice that starting from t, all the stations are in the set of "very active" stations. According to the definition of fairness, a round robin cycle starts at $T_i^1(t)$ at each station $i \in \{1, 2, ..., K\}$. This extra delay varies between 0 and $K-1$, depending on the position of the station in the cycle. Therefore, there must exist a station i' which is at the end of the cycle, i.e.,

$$B_{i'}^G \geq \sum_{k=1}^{i'-1}(D_u^{k+1} + D_d^k) + K - 1 \tag{5}$$

Comparing (4) and (5) with the upper bound on the access delay in FUFA, $B_i = \sum_{k=1}^{i-1}(D_u^{k+1}+D_d^k)+K-1$, we can see that with FUFA, station i' has exactly the minimum possible value of the maximum access delay, while each of the other stations has a maximum access delay that is at most $K-1$ away from its minimum possible value.

6 Conclusion

In this paper, we have designed and analyzed a fully utilized and fair (FUFA) dual bus protocol to illustrate the relationship between utilization, fairness and access delay. The basic concept of the protocol is to give equal access to all the stations according to the most updated information available through the reservation bus. In particular, according to the information from downstream and the idle slots propagated previously, each station computes the latest estimate on the number of active downstream stations, and serves them in a round robin fashion. It was shown that FUFA achieves fairness with full utilization, where a fair round robin cycle for distributing idle slots among the set of "very active" stations (with other stations idle) is established within one round trip propagation delay between the most upstream and most downstream active stations. Additionally, the protocol provides a bounded access delay which is linear in the round trip propagation delay, and at most a constant $K-1$ away from its minimum possible value for any bus protocol that is both fully utilized and fair. To the best of our knowledge, we believe that this is the first protocol that provides full utilization, fairness and bounded access delay. The following issues warrant further research.

- Simulation results would be useful to analyze both steady state and transient state behaviors.
- The fairness defined here is for steady state behavior. It is desirable to analyze the fairness property in transient states, where a protocol is defined to be "fair" if the number of idle slots used by any heavily loaded station during some interval T is at least as large as the number used by any other station, less some constant independent of T.
- Modifications of the protocol to make it more practical should be investigated.

References

[1] Barry, R. A. *et al.*, "All-Optical Network Consortium–Ultrafast TDM Networks," *IEEE J. Select. Areas Comm.*, SAC-14:999-1013, 1996.

[2] Chiu, A. L., and Gallager, R. G., "Full Utilization, Fairness and Bounded Access Delay on High Speed Bus Networks," LIDS-P-2321, Laboratory for Information and Decision Systems, MIT, 1996.

[3] Finn, S. J., "HLAN-An Architecture for Optical Multi-access Networks," *IEEE 1995 Dig. LEOS Summer Top. Mtg.*, p. 45-47, 1995.

[4] Gallager, R. G., *Discrete Stochastic Processes*, Kluwer Academic Publishers, 1995.

[5] Hahne, E. L., Choudhury, A. K., and Maxemchuk, N. F., "Improving the Fairness of Distributed-Queue-Dual-Bus Networks," *Infocom*, San Francisco, Ca., pp. 175-184, 1990.

[6] IEEE 802.6 Working Group, "Proposed IEEE Standard 802.6 - Distributed Queue Dual Bus (DQDB) - Metropolitan Area Network (MAN)," Draft versions: June 1988, Nov. 1989, Spring 1989, August 1989.

[7] Limb, J. O. and Flores, C., "Description of Fasnet – A Unidirectional Local-Area Communications Networks," *Bell Syst. Tech. J.*, 61:1413-1440, 1982.

[8] Mukherjee, B. and Bisdikian, C., "A Journey through the DQDB Network Literature," *Performance Evaluation*, 165:129-158, 1992.

[9] Nassehi, M. M., "CRMA: An Access Scheme for High-Speed LANs and MANs," *IEEE ICC*, Atlanta, Ga., pp. 1697-1702, 1990.

[10] Pillai, R. R. snd Mukherji U., "Analysis of a Variation of the Distributed Queueing Access Protocol," *IFIP TC-6 Networks*, 1994.

[11] Watson, G. C. and Tohmé, S., "S++ – A New MAC Protocol for Gb/s Local Area Networks," *IEEE J. Select. Areas Comm.*, SAC-11(4):531-539, 1993.

Appendix: minimum system queueing delay

Let the queueing delay of a segment be the interval between its entrance and departure into the queue. Due to the distributed nature of the system, we use a time reference $T_k^1(t)$ which is the time when a downstream slot starting from station 1 at time t passes station k as it travels down the bus, for $k = 1, 2, ..., K$. Since the first idle slot arrives at station 1 at time 0, no departure from station k occurs before time $T_k^1(0)$. Therefore, without loss of generality, we can assume that no arrival occurs at station k before $T_k^1(0)$, for all k. For $k \in \{1, 2, ..., K\}$, $t \geq 0$, denote respectively, $A_k(T_k^1(t))$ and $D_k(T_k^1(t))$ as the number of arrivals and departures from station k between $T_k^1(0)$ and $T_k^1(t)$. Define the system arrival and departure processes $\{A(t) : t \geq 0\}$ and $\{D(t) : t \geq 0\}$ as follows,

$$A(t) = \sum_{k=1}^{K} A_k(T_k^1(t)),$$
$$D(t) = \sum_{k=1}^{K} D_k(T_k^1(t)).$$

The system is said to be "empty" at t if $L(t) \triangleq A(t) - D(t) = 0$. That is, the system is "empty" if an idle slot that starts from station 1 at t sees an empty queue at each station as it travels down the bus. We define the ith data segment to be the data segment that causes $A(t)$ to be incremented for the ith time. That is, if this ith increment occurs at time t due to the arrival of a data segment at station k, then that data segment actually arrives at station k at $T_k^1(t)$. Therefore, W_i is the queueing delay of that ith data segment in the system. This system can be viewed as a non-FCFS system in the proof of Little's Law (see, for example, section 3.6 of [4]). Letting t be any time when the system is empty, $\bar{W}(t)$ is defined as the system queueing delay up to time t, hence,

$$\bar{W}(t) \triangleq \frac{\sum_{i=1}^{A(t)} W_i}{A(t)} = \frac{\int_{s=0}^{t} L(s)ds}{A(t)}. \tag{6}$$

Theorem 4: Any protocol with full utilization provides the same system queueing delay $\bar{W}(t)$ for any $t \geq 0$ when the system is empty, and it is the minimum system queueing delay provided by any protocol.

Proof: Based on (6), we can prove the first half of the theorem by showing that for all $t \geq 0$, $\int_{s=0}^{t} L(s)ds$ is the same for all protocols with full utilization.

According to the definition of $L(s)$, $L(s) \triangleq A(s) - D(s)$, all we need to show is that $D(s)$ is the same for all $s \geq 0$ among all protocols with full utilization. This follows from the definition of full utilization which implies that for all $s \geq 0$,

$$\begin{aligned} D(s) &= D(s-1)+1 \quad \text{if } A(s) - D(s-1) > 0, \\ D(s) &= D(s-1) \qquad \text{otherwise} \end{aligned} \tag{7}$$

with the same initialization $D(-1) = 0$ for all protocols with full utilization.

From (6), in order to show that $\bar{W}(t)$ provided by any protocol with full utilization is the minimum system queueing delay provided by any protocol, it is sufficient to show that for all s, $D(s)$ is at its maximum for any protocol with full utilization. Let $\{D^A(s) : s \geq 0\}$ be the departure process for an arbitrary protocol. D^A has the following two constraints, for all $s \geq 0$,

$$D^A(s-1) \leq D^A(s) \leq D^A(s-1)+1, \tag{8}$$
$$L^A(s) = A(s) - D^A(s) \geq 0. \tag{9}$$

The initialization is $D^A(-1) = 0$. Based on the fact that $D^A(s)$ is a non-decreasing function of s, (9) implies that,

$$A(s) - D^A(s-1) \geq 0. \tag{10}$$

Now, we use induction on time $s \geq 0$ to show that for all $s \geq 0$,

$$D(s) \geq D^A(s).$$

1. Let $s = 0$. If $A(0) = 0$, $D(0) = D^A(0) = 0$. If $A(0) > 0$, then from (7), $D(0) = 1$, and from (8), $D^A(0) \leq 1$. Thus we have established the basis.

2. For an arbitrary s, assume that $D(s-1) \geq D^A(s-1)$. We need to show that $D(s) \geq D^A(s)$. Comparing (7) with (8), the only case we need to consider is when $D(s) = D(s-1) = D^A(s-1)$ with the condition that $A(s) - D(s-1) = 0$. This means that $A(s) - D^A(s-1) = 0$ based on (10) with the induction assumption. Combining with the constraint (9), we have

   $$D^A(s) = D^A(s-1).$$

   Therefore, $D(s) = D^A(s)$ in this case.

We have completed the proof. Note that with the assumption of statistical stationary, the results can be extended to the time average and ensemble average. ■

Architectural Concepts in Implementation of End-system Protocols for High Performance Communications

K. Ravindran*
Department of Computer Science
The City University of New York
New York, NY 10031,U.S.A
ravi@cs-mail.engr.ccny.cuny.edu

G. Singh
Department of Computing and Information Sciences
Kansas State University,
Manhattan, KS 66506, U.S.A
singh@cis.ksu.edu

C.M. Woodside
Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario K1S 5B6, Canada
murray.woodside@sce.carleton.ca

Abstract

The paper presents a functional view of end-system protocol implementations whereby the protocol is decomposed into multiple functions, each causing a change in the protocol state. This view makes the interactions and the relationships among the various functional modules explicit. In terms of this view, currently prevalent architectural optimizations for performance improvement (such as 'parallel executions' and 'integrated layer processing') can be easily described as a set of control flow relationships among modules. If a protocol implementation is analyzed using our functional model, the possible architectural optimizations in the protocol can be easily identified and implemented without violating correctness. Thus, our approach can be used to optimize existing implementations by casting the underlying protocols in our framework, and it is particularly useful in developing implementations for new protocols.

1. Introduction

The search for generality, flexibility and standardization has led to bulky implementations of end-systems. Examples are the TCP and the ISO TP4 based transport systems. The implementations of such systems often conform to the OSI layered architecture. However, the slowness of execution of the protocol implementations, which is essentially due to sequential processing of the complex protocol procedures for each data unit in various layers, is becoming a limiting factor in some emerging applications which require high bandwidth and large volume data exchanges between various application devices through a high speed backbone network. For example, the transfer of images and live video information may involve transporting hundreds of gigabytes of data at high rates, typically in the range of 150 to 200 *mbps* [1]. Such a capability cannot be met by existing protocol implementation structures, which typically support transfer rates ranging between 750 *kbps* to 6 *mbps* [2, 3]. This warrants high performance implementations of end-systems that can provide high transfer rates meeting the application needs, limited only by the network speeds.

The various processing activities on a application-specific data unit (or, packet) such as scheduling control, multiplexing and presentation level processing of data are part of the end-system protocol. The protocol can also include lower level functions such as rate control and error recovery on data. Figure 1 illustrates the placement of these functions in an end-system node with respect to application entities and the backbone transport network attached to this node.

The ways in which various functions influence the overall performance of end-systems are often difficult to be analyzed in a systematic way and generalized for broad usage. This difficulty can often obscure many performance engineering aspects that may be inherently possible. For instance, the communication level processing of a video picture data can proceed in parallel with the compression/decompression of this data, provided the presentation and communication activities on the video data are carefully separated. This parallelism can be obscured if, for instance, a conventional layered implementation of various protocol

*Part of this work was performed when the author was at Kansas State University.

0-8186-7453-9/96 $05.00 © 1996 IEEE