Title
Collision Avoidance Tree networks

Authors
Huang, Hung Khei
Suda, Tatsuya

Publication Date
1992

Peer reviewed

Collision Avoidance Tree Networks [1]

Technical Report 92-54

Hung Khei Huang, Tatsuya Suda
Department of Information and Computer Science
University of California, Irvine
Irvine, CA 92717
Phone: (714) 856-5474
FAX: (714) 856-4056
e-mail: huang@ics.uci.edu, suda@ics.uci.edu

Abstract

The Collision Avoidance Tree is a new local area network based on a hardware device called collision avoidance switch, which arbitrates random access to a shared communications channel. Collision Avoidance Tree combines the benefits of random access (low delay when traffic is light; simple, distributed, and therefore robust, protocols) with concurrency of transmission, excellent network utilization and suitability for the domain of high-speed, optical networking.

The Collision Avoidance Tree is classified in two classes: the Collision Avoidance Single Broadcast (CASB) Tree and the Collision Avoidance Multiple Broadcast (CAMB) Tree. The CASB Tree allows only a single transmission on the network at a given time, while the CAMB Tree is more general and allows concurrent transmissions on the network.

This paper describes network architectures (e.g., station and switch protocols) and designs and implementations of the CASB and CAMB Trees. Performance results derived from analyses, simulations, measurements of experimental networks are also presented.

[1]This material is based upon work supported by the National Science Foundation under Grant No. NCR-8907909. This research is also in part supported by the University of California MICRO program.

1 Introduction

The Collision Avoidance Tree is a network based on a hardware device called collision avoidance switch, which arbitrates random access to a shared communications channel [1, 2, 3, 4]. In a Collision Avoidance Tree, the collision avoidance switches and stations are organized according to a tree topology. The switches are the internal nodes, and the stations are the leaves of the tree. Stations use simple, random access protocols to transmit packets. Packets climb the tree and are broadcast to all the stations. The collision avoidance switches arbitrate access to the transmission channel so that while the channel is being used by one station, other stations are blocked from using it, and thus, collisions are prevented. The Collision Avoidance Tree provides low packet delay under light load, and the switches avoid waste of channel utilization due to collision resolution and the retransmission of collided packets.

The Collision Avoidance Tree is proposed and implemented to provide an alternative to existing random access based LANs (such as the Ethernet) and controlled access based LANs (such as the Token Ring). In random access based LANs, transmission rights are simultaneously offered to a group of stations in the hope that exactly one of the stations has a packet to send [5]. However, if two or more stations send packets simultaneously on the channel, these packets interfere with each other and none of them are correctly received by the destination stations. Controlled access based LANs avoid collisions by coordinating access of the stations to the channel by imposing either a predetermined or dynamically determined order of access. Access coordination may be done by use of a token as in the Token Ring. This consumes some channel capacity regardless of whether stations require access or not.

While controlled access based LANs are efficient when traffic is heavy, under light traffic conditions they result in unnecessary packet delays as stations that have a packet to transmit have to wait their turn. In contrast, random access based LANs exhibit small packet delays under light traffic conditions. Stations transmit as soon as they want access to the channel, and the probability of a collision is low when traffic is light. However, when traffic is heavy, a large number of collisions occur resulting in a loss of channel utilization from the transmission of colliding packets and from collision resolution via retransmission.

In the Collision Avoidance Tree, no access coordination is required. Stations transmit a packet as soon as one is available. Thus, the disadvantage of a control access based protocol is eliminated. Also, there is no collision even under heavy traffic conditions, since the collision avoidance switches resolve contentions among packets. Therefore, the disadvantage of random access protocol is eliminated.

In addition to these overall architectural advantages, several features of the Collision Avoidance Tree suggest that it may be good for high speed optical networks [6, 7, 8, 9, 10] and that emerging optical technology may be readily applied to its implementation [11]. First, the point-to-point physical architecture of the network makes it compatible with fiber optic technology. Second, the simplicity of the collision avoidance switches suggests that current device-count constraints of photonic switching technology do not pose a problem for a photonic implementation of the collision avoidance switch. Third, there is no requirement of a minimum value in the ratio of packet transmission time to the end-to-end-propagation delay (that is, a minimum packet length requirement) in the Collision Avoidance Tree as there is in the Ethernet. The Collision Avoidance Tree is therefore not adversely affected by an increase in channel speed (that is, a shorter packet transmission time, while the end-to-end propagation delay remains constant). Finally, the use of high channel speeds in the Collision Avoidance Tree reduces contention in the network, which contributes to a large reduction in the average transmission delay of packets.

The Collision Avoidance Tree is classified in two classes: the Collision Avoidance Single Broadcast (CASB) Tree and the Collision Avoidance Multiple Broadcast (CAMB) Tree. The CASB Tree allows only a single transmission on the network at a given time, while the CAMB Tree is more general and allows concurrent transmissions on the network. Many researchers have investigated various issues (such as station and switch protocols, switch designs, performance, network implementation) in the Collision Avoidance Tree [1, 2, 3, 4, 11, 12, 13]. An experimental CAMB Tree [12] and a commercial CASB Tree networks are also available [3].

In this paper, we present the network architectures, protocols, implementations of the Collision Avoidance Tree, as well as its performance characteristics. The rest of the paper is organized in the following way. Architectures, protocols, designs and implementations of the CASB Tree and of the CAMB Tree are described in sections 2 and 3, respectively. Performance characteristics of the CASB and CAMB Trees are presented in section 4. Reliability issues are discussed in section 5. Variations of the Collision Avoidance Tree are presented in section 6.

2 Collision Avoidance Single Broadcast (CASB) Tree

The Collision Avoidance Single Broadcast (CASB) Tree is a simple form of the Collision Avoidance Tree where packets reach their destinations by climbing the tree and being broadcast by the root switch.

2.1 Network Architecture and Switch Architecture

The simplest form of the CASB Tree is the Broadcast Star [1, 2, 3, 4]. In the Broadcast Star, all stations are directly connected to a central (collision avoidance) switch by full duplex channels. Each of these channels consist of an uplink and a downlink. See Fig. 1. The switch may be viewed as functionally containing two components: the Selector and the Broadcaster. The Selector selects one packet from one of its uplinks and forwards this packet on a single output line to the Broadcaster. The Broadcaster receives the packet from the Selector's output line and broadcasts it via each of its downlinks. The Selector can be in one of two states, busy or idle. It is busy from the time it has selected a packet to the time it has finished forwarding the packet to the Broadcaster. Otherwise the Selector is idle. While the Selector is busy, all packets arriving on uplinks are ignored in their entirety. Upon becoming idle, the Selector selects the next newly arriving packet.

An important feature of a collision avoidance switch is that when two or more packets contend for the output line, it is guaranteed that one of the packets acquires the line and is successfully transmitted on it. Thus, no channel time is wasted in the transmission of collided packets. Collision avoidance can be implemented with very little circuitry. Switch implementations are discussed in section 2.3.

The CASB (Collision Avoidance Single Broadcast) Tree extends the Broadcast Star and has a general rooted tree topology [1, 3]. The nodes of the CASB Tree consist of switches similar to the Broadcast Star switch: they each contain a Selector and a Broadcaster. However, these two components are not linked. Instead, the Selector passes its output to its parent switch, and the Broadcaster receives its input from its parent switch as shown in Figure 2. The Selector and Broadcaster are only connected in the root switch.

The stations are connected as children of the leaf switches of the tree. A packet transmitted by a station climbs the tree by being selected by each switch along the packet's path to the root. The root switch serves the same function as the switch in the Broadcast Star. Any packet selected by the root's Selector is broadcast to the children of the root. Each child repeats the broadcast to its children until the broadcast reaches every station. Once a packet is selected by the root, it is assured of being broadcast in its entirety to the whole tree.

The functionality of the CASB Tree is similar to the Broadcast Star. The root node in the CASB Tree corresponds to a central switch in the Broadcast Star. The selection of packets is distributed over several levels of switches, as is the propagation of packet broadcasts. Three factors would lead to choice of the CASB Tree instead of a Broadcast Star.

- Due to hardware limitations, the central switch of a Broadcast Star may accommodate only a limited number of stations; The CASB Trees allow for easy expansion when the fanout of a switch is fully occupied. Because all switches in the CASB Tree are identical, expansion entails simply connecting a new switch as a child of a fully loaded switch.
- The CASB Tree provides some fault tolerance that is not present in the Broadcast Star, because one switch failure does not disable the entire CASB Tree.
- The cabling cost of a Broadcast Star may be excessive. The intermediate switches in the CASB Tree allow cables to be shared instead of running a separate cable for each station to a central switch. This may be particularly attractive if some cable runs are long or are difficult to install.

2.2 Station Protocol

A simple, pure random access protocol is used as the station protocol without the need for contention resolution subprotocol. The station protocol for the CASB Tree (and for the Broadcast Star) is the following.

(1) A station transmits a packet as soon as one is available.

(2) The station monitors its downlink for the start of its packet.

(3) If the station does not see the start of its packet after a propagation delay to and from the switch, then it retransmits the packet immediately,

(4) otherwise the station does see its packet and knows that the packet has won the switch and will be broadcast in its entirety.

In this protocol, stations resubmit their packets as soon as they learn of transmission failure. This lack of deferment in retransmissions does not hurt performance because no collision with an ongoing transmission can occur in the CASB Tree.

Some alternative station protocols are available and investigated in [14, 15]. In one protocol [14], different retransmission time intervals are assigned to different stations (or different traffic classes) to achieve priority services. In another protocol [15], packets that arrive at the root switch (refered to as the Hub in [15]) while it is busy transmitting another packet are buffered and transmitted at a later time. This protocol leads to a smaller number of packet retransmissions at the expense of the increased complexity of the switch.

2.3 Designs and Implementations

Both the Broadcast Star and the CASB Tree have been designed and implemented [1, 2, 3]. A design for a Broadcast Star switch is presented in [2]. In this design each station is connected to a central switch through an independent user-network interface circuit which resides within that central switch. A number of interface circuits are connected to a central bus (internal to the switch) through an arbitrator circuit which resolves any contention among packets from different stations. It is shown in [2] that the user-network interface circuits can be implemented by simple flip-flops and logic gates.

An implementation of the CASB Tree called the Hubnet is presented in [3]. The Hubnet uses fiber optics and operates at 50 Mbits/sec. The CASB switches are called Hubs in this network. The transmission medium access units (called network access controllers (NACs)) form the leaves of the tree. A packet sent from a host (through a NAC) transverses the tree until it reaches the central Hub which broadcasts the packet to all of the NAC's in the network. The hosts follow the CASB station protocol described in Section 2.2. The packets may be any length up to a 4096 byte hardware implementation limit. The NAC consists of a MC68000 microprocessor, buffers for packets, control code storage and network interfaces.

The implementation of the Hubnet follows the ISO-OSI reference model. The OSI Physical Layer is implemented in the network interface unit. The OSI Data Link Layer and a part of the OSI Network Layer are implemented in hardware within the NAC. The remainder of the OSI Network Layer and the OSI Transport Layer is implemented in the NAC microprocessor software. The OSI Transport Layer protocol is implemented in two sublayers. The lower layer implements a reflective datagram (used to negotiate the establishment of a virtual circuit) and a blind datagram (used to provide reliable transmission of packets over virtual circuit). The upper layer provides a number of different transport layer protocols. Finally the OSI Session and Presentation Layers are also implemented within the NAC software. Performance measurements of the Hubnet are discussed in section 4.1.

An alternative design of a CASB Tree switch is presented in [1]. CASB switch designs and implementations are also given in [12]. The designs and implementations in [12] support a more general form of the Collision Avoidance Tree network called the Collision Avoidance Multiple Broadcast (CAMB) Tree. They are discussed in detail in section 3.4.

3 Collision Avoidance Multiple Broadcast (CAMB) Tree

The Collision Avoidance Multiple Broadcast (CAMB) Tree is a Collision Avoidance Tree which supports concurrent transmissions [4, 11]. The CAMB Tree is a CASB Tree with switch modifications that allow any switch to broadcast to the subtree below it. A packet reaches its destination by climbing the tree and being broadcast by its proper ancestor. The proper ancestor of a packet is the switch that roots the minimal subtree containing both the source and destination stations of the packet. Each source station places in the packet header the address of the packet's proper ancestor switch.

3.1 Switch Architecture

The CAMB switch may be decomposed functionally into three sections: the Uplink Selector (US), the Address Recognizer (AR), and the Downlink Selector (DS) as illustrated in Figure 3. The US is identical to the Selector of a CASB switch. The only difference is that it sends a selected packet to the AR, not to the parent uplink as in the CASB switch. The AR determines if the switch is the proper ancestor of the selected packet. The DS is a slightly more complicated version of the Broadcaster in the CASB switch. In the following we describe the AR and DS in more detail.

The AR knows the unique address of its switch and executes the following protocol:

(1) The AR checks the header of each packet passed to it by the US to see if the switch is the proper ancestor.

(2) If the switch is not the proper ancestor, the AR transmits the packet to the parent uplink,

(3) otherwise the switch is the proper ancestor, and the AR checks the status of the DS,

- **(3-1)** if the DS is busy broadcasting a packet from the parent downlink, then the AR discards the packet it is receiving. (This is called a broadcast preemption.)
- **(3-2)** otherwise the DS is idle, and the AR forwards the packet to the DS and to the parent uplink. (We will explain in section 3.2 why the packet is also passed to the parent uplink even though it has been selected by its proper ancestor for broadcast.)

The DS receives packets from either the AR or the parent downlink. It broadcasts each packet it receives on all the children downlinks. The DS uses the following protocol:

(1) If the DS is idle when it starts receiving a packet from either the AR or the parent downlink, then it starts broadcasting that packet.

(2) If the DS is busy broadcasting a packet from the AR when it starts receiving a packet from the parent downlink, the AR-broadcast is terminated and the packet from the parent is broadcast instead. (This is called a broadcast abortion).

Packets received by the DS from the parent downlink are given priority over those received from the AR. If instead, the DS were designed so that it blocked the packet received from the parent downlink, then it would be possible for a source station to receive the broadcast of its packet without the broadcast being received by the destination station. However, this situation is incompatible with the station protocol, in which the source station uses the receipt of its own packet as an indication that the packet was also broadcast to the destination station. The station protocol is described further in Section 3.3.

It should be noted that the switch does not store and forward packets, although small amounts of memory might be necessary to support functions such as header processing. The amount of logic required by the switch is quite small and the processing performed by the switch is simple (see section 3.4). This simplicity may be beneficial as link channel speeds increase to the Gbits/sec level and beyond through the use of optical transmission lines. Because of the small logic device count, it may be feasible for this architecture to take advantage of photonic technologies. Photonic design of a CAMB switch is presented in [16].

3.2 Concurrency: Effects of Climbing Packets

A major advantage of the CAMB Tree is the possibility of concurrent transmissions. This concurrency is made possible by the point-to-point, segmented nature of the CAMB Tree which allows broadcasting to subsets of the tree.

The CAMB switch protocol is designed such that, when a packet is selected by the US of its proper ancestor, in addition to being passed to the DS, the packet is also transmitted on the parent uplink. The following is the motivation for having a packet climb the tree above its proper ancestor. Every time the climbing packet busies the US of a switch above its proper ancestor, it prevents stations beneath that switch from using the switch as a broadcast point. This makes the packet's own broadcast less vulnerable to abortion. If the packet can busy the US of every switch up to and including the root switch, then the packet's broadcast is completely shielded from abortion. To the extent that the packet is unsuccessful in winning switches contentions above its proper ancestor, the packet simply "takes its chances" and hopes that it won't be aborted.

More generally, the effect of climbing packets is the partitioning of the tree into broadcast domains. Every time a packet busies the US of any switch, it creates partitions that are the children subtrees of that switch. Within each of these partitions, broadcasts may occur; but while they proceed, they are vulnerable to abortion by broadcasts originating higher in the tree. Any attempt to transmit from within a partition to a destination outside the partition does not succeed. This is illustrated in Figure 3. Each transmission indicated is possible except for the one marked with the cross. It fails because the parent of its partition is busy.

3.3 Station Protocol

The CAMB station protocol is based upon a station monitoring its downlink for the completed broadcast of its packet. In the CASB Tree, because there is only one broadcaster, when a station sees the start of the broadcast of its packet, it is assured that the packet's transmission will not fail due to contention. This is not the case, however, in the CAMB Tree. Because of the possibilities of broadcast abortion, the CAMB Tree station protocol requires that a station see the broadcast of its entire packet.

The CAMB station protocol is as follows:

(1) A station transmits a packet as soon as one is ready, starting at time t.

(2) The station monitors the downlink for the broadcast of the packet.

(3) If the station does not see the start of its packet by time $t + R_{pd}$ (where R_{pd} is the round trip propagation delay between the station and the proper ancestor), then it retransmits the packet immediately,

(5) otherwise (the start of the packet is seen within this time)

- **(5-1)** if the station sees the broadcast of the packet truncated by the broadcast of another packet, then it retransmits the packet immediately,
- **(5-2)** otherwise the station sees the broadcast of the entire packet (and the transmission is successful and the station can transmit a new packet).

In this protocol, the station times out at time $t + R_{pd}$ if it has not seen the start of its packet. This means that each station must keep a table of round trip delays for its proper ancestors. This information could be provided when the network is initialized. Alternatively, at network initialization a station could assume the value of R as the worst case round trip delay between any station and the root. The station can subsequently learn an accurate value for each R_{pd} by timing the round trip delay for each proper ancestor. This sampling could be done using a special control packet or using a regular packet from the station. This approach of timing packets is feasible because, for a packet that is transmitted successfully, there are no non-deterministic delays introduced by the network.

A simpler approach is to assume $R_{pd} = R$ for every packet transmission in the network, and not consider variations in round trip delays to and from different proper ancestors. In this case, the value of R may be set in the station's interface hardware.

Note that if channel speeds are high, packet transmission times may be small compared to propagation delays. It is then possible for a station to receive the broadcast of other packets before seeing the broadcast of its own packet. Therefore, even though a station sees the start of another packet, it waits until the time $t + R_{pd}$ before retransmitting its own packet.

The CAMB station protocol described above requires that a station must succeed in the transmission of one packet before it attempts the transmission of the next packet in its queue. In the case where the channel speed increases and the ratio of the packet transmission time to R_{pd} decreases, this protocol might not be ideal since the idle time of links tends to become large.

Station protocols that allow a station to transmit subsequent packets before knowing the success or failure of prior transmissions and use resequencing of the packets at the destination station are feasible to increase link utilizations. This protocol modifications may be a good answer for future regimes of optical communication in which channel speeds are high and the ratio of the packet transmission time to the propagation delay is small.

3.4 Designs and Implementations

Complete CAMB Tree network design and implementation (including switch, station/network interface and protocols) using standard electronic devices are presented in [12]. This CAMB Tree is an experimental network that operates at 10 Mbits/sec. The CAMB Tree implementation follows the protocol layering architectures of the IEEE 802 local area networks as shown in Figure 4. The Physical Layer consists of CAMB switches, transmission cables, and transceivers located on the station/network interface board. It is responsible for bit level transmission. The MAC Layer and a part of the LLC Layer are implemented in the station/network interface board installed in the station. The MAC Layer is responsible for the station protocol described in Section 3.3. This experimental CAMB Tree network also supports the TCP/IP protocol suite (implemented in the workstation) on top of the IEEE 802 protocol stack.

Figure 5 illustrates the switch design and implementation, which follows the functional decomposition described in Section 3.1. In the switch, each transmission link (either up- or down-link) consists of a control path and a data path. The control path signals the beginning and the end of each packet. The data path carries the packet sent to the switch. In the US section, the *synchronizer* receives packets from uplinks; the *start recognizer* detects the beginning of the packets, activating the *priority resolver*, which will randomly select one of the packets based on the control lines; the selected packet is allowed to go through the *multiplexer*. In the AR section, the *shift register* stores the destination address for one clock cycle, allowing the *comparator* to check it with the switch's address; based on the result of this comparison, the *routing logic* routes the packet to DS section and/or to parent switch; *end recognizer* detects the end of a packet transmission and resets the US, enabling the US to select a new packet. In the DS section, the *Selector* broadcasts a packet either from the parent switch or from the AR; the *flip-flop* forces an end of AR packet transmission in case of an abort.

The station/network interface board connects a station to a CAMB Tree. The interface board consists of an 80186 microprocessor, buffer memory for packets, control code storage, and interface circuitry to the station and to the network. Detailed design and implementations of the switch and the interface board, as well as MAC Layer, are presented in [12].

A VLSI design of the CAMB switch is presented in [12]. A switch design based on photonic devices is also available [11]. The photonic switch design in [11] uses directional couplers (an electronically controlled, optical 2×2 switch) to provide optical paths through the CAMB switch. Both VLSI and photonic switch designs also follow the section decomposition and functionality described in Section 3.1.

4 Performance of the Collision Avoidance Tree

The performance of the Collision Avoidance Tree has been studied through queueing analysis, simulations, and measurements of experimental Collision Avoidance Tree networks. Throughout the following numerical examples, the following assumptions and notations are employed, unless otherwise stated. The packet length is assumed to be constant (P bits per packet). The channel speed is denoted as C (bits/second). The packet transmission time (P/C) is used as the unit time. R denotes the propagation delay (measured in the packet transmission time), and α denotes the ratio of the round trip propagation delay (R) to the packet transmission time (P/C). There are N number of stations in the network. Packets arrive at a station according to a Poisson process with a rate λ (packets/unit time). The aggregated arrival rate of packets to the network is $N\lambda$.

4.1 Performance of the Collision Avoidance Single Broadcast (CASB) Tree

In this subsection, we present the performance of the Broadcast Star and the CASB Tree through analysis [17, 14, 13, 4], simulations [18, 16], and measurements [19, 12].

Fig. 6 and 7 show the average transmission delays in a Broadcast Star. These figures are based on queueing analysis [17]. Fig. 6 assumes a Broadcast Star with an infinite number of stations which collectively generate packets according to a Poisson process. This figure shows the average transmission delays as a function of the throughput for different values of α. The average transmission delays for various values of α have similar behavior, and they grow rapidly when the throughput exceeds 0.8. The maximum throughput for all cases is very close to 1.0. In the case of $\alpha = 0$, the maximum throughput value is actually 1. Under heavy traffic conditions it is very likely that all the channels are always busy. Therefore, there is always a successful transmission, and a throughput of 1 is achieved.

Fig. 7 assumes a Broadcast Star network with N number of stations and shows the average transmission delays as a function of the aggregated arrival rate $N\lambda$ to the network. The horizontal axis is logarithmic. Each station has a single buffer to store arriving packets, and packets which arrive when the station buffer is full are lost. α is assumed to be 0.05. Networks with the larger values of N show larger average transmission delays. This is because that the probability of two or more stations attempting to send a packet at the same time is greater when there are a larger number of stations in the network. The average transmission delay curves become flat as the network traffic load ($N\lambda$) increases to 1. This shows that there is an upper bound in the average transmission delay even in heavy load conditions.

Fig. 8 compares the performance of a Broadcast Star with the performance of a Token Ring and an Ethernet through simulations. All three networks simulated consist of eight stations with a single buffer in each station (as in Fig. 7). Packets which arrive when the station buffer is full are lost. This figure shows the average transmission delays as a function of the packet arrival rate at a station (λ) for different values of α. The results show that the Broadcast Star has the smallest transmission delay for all traffic loads. This is because in the Broadcast Star, there is no waste of channel capacity due to collisions of simultaneously transmitted packets, nor due to transmission of a token (i.e., no control overhead). It is also observed that the Ethernet provides smaller delays than the Token Ring when the traffic load is light, while the Token Ring provides smaller delays when the traffic load is heavy. The difference in the transmission delays between the Broadcast Star and the Token Ring and the Ethernet is more accentuated when α is larger. This fact shows that the Broadcast Star is more suitable than the Ethernet or the Token Ring in high speed network environments, since the value of α is larger in high speed networks than in conventional networks.

The average transmission delays in the Broadcast Star, the Token Bus and the Ethernet are compared through an approximate analysis in [13]. The results in [13] confirm the observation made in Fig. 8. The Broadcast Star exhibits the smallest transmission delay for all traffic loads among the three networks.

Experimental CASB Tree networks have been built, and their performance measurements are available in [19, 12]. [19] presents the performance measurements of a Broadcast Star (refered to as Hubnet in [19]) with one station and a traffic generator. Measured performance statistics include the number of retransmissions for different network traffic loads and for different packet sizes. The results show that the network effectively handles the traffic load up to 0.75, without significantly increasing the number of retransmissions required to transmit a packet successfully. [12] also describes the performance measurements of an experimental CASB tree (as well as an experimental CAMB tree). The results are discussed in the following section.

[16, 18, 14] also present performance study of the CASB Tree.

4.2 Performance of the Collision Avoidance Multiple Broadcast (CAMB) Tree

In this subsection we present the performance of the CAMB tree and compare it with the CASB Tree performance. Results presented in this subsection are through simulations and measurements of an experimental CAMB Tree. Due to the complexity, there is no analytical results available for the performance of the CAMB Tree.

Fig. 9 and 10 present simulation results on the performance of the CAMB Tree and the Broadcast Star [11]. These figures compare the performance of these two types of the Collision Avoidance Tree and also show the impact of the traffic locality on the performance of the CAMB Tree. The simulated networks are an eight station Broadcast Star and an eight station two-level CAMB Tree run with various traffic localities. The Broadcast Star has one switch with eight stations beneath it. The two-level CAMB Tree has a root switch with two switches beneath it. Each of these two children switches has a cluster of four stations beneath it. Each station has infinite buffer. Simulations are performed for $\alpha = 0.05$ (in Fig. 9) and $\alpha = 5$ (in Fig. 10), where α is the ratio of the propagation delay to the packet transmission time. A higher α value corresponds to a higher speed network. For instance, $\alpha = 5$ corresponds to a 1 Gbits/sec. network with 1000 bits packets, while $\alpha = 0.05$ corresponds to a 10 Mbits/sec. network with the same packet length. To study the effects of traffic locality in the two-level CAMB Tree, the simulations are done also with different Origin/Destination traffic pattern: uniform, 75% and 100% localities. With a uniform O/D matrix, packets at each station are destined to the other stations with a uniform probability of 1/7, whereas with the 75% locality O/D matrix, 75% of the traffic goes to a station within the local cluster and only 25% of the traffic passes through the root switch. The horizontal axis shows the aggregated packet arrival rate to the network.

The simulation results show the effects of locality of transmission on performance in the two-level CAMB Tree. When all packets are destined to stations located in the other subtree, and hence, all packets must be broadcast by the root switch, the average delay in a two-level eight-station CAMB Tree is the same as that in an eight-station Broadcast Star. This case gives the upper bound on the delay in a two-level eight-station CAMB Tree because no concurrent transmissions are possible. On the other hand, if all communications are local within each cluster, the CAMB Tree behaves like two independent four-station Broadcast Stars. This case gives the lower bound on the delay in a two-level eight-station CAMB Tree. In between these two extremes, varying the degree of locality of communication yields different performance for the CAMB Tree. From Fig. 9 and 10 we see that a CAMB Tree with a higher degree of locality yields smaller delay. This reflects a decrease in the preemption and abortion of packets at the lower level tree switches, and thus an increase in the concurrency of successful transmissions in the network.

Fig. 9 and 10 also compare the effects on the average transmission delay in a CAMB Tree if one increases the channel speed from 10 Mbits/sec to 1 Gbits/sec. For a 10 Mbits/sec, 1000 bit packet, 500 m network, the unit of time (packet transmission time) in Fig. 9 is 10^{-4} sec. For a 1 Gbit/sec network, the unit of time in Fig. 10 is 10^{-6} sec. Therefore, in this example, there is a multiple factor of 100 between the packet arrival rate in Fig. 9 and 10. For instance, the packet arrival rate of 0.01 in Fig. 10 corresponds to the packet arrival rate of 1.0 in Fig. 9. In the 1 Gbit/sec 2-level CAMB tree with uniform traffic, the delay for a 0.01 arrival rate is 5×10^{-6} sec. The delay at the corresponding arrival rate for the 10 Mbits/sec network (arrival rate of 1.0 in Fig. 9) is 9.5×10^{-4} sec. Essentially, the increase in channel speed has eliminated collisions in the 1 Gbit/sec network, with the result that the transmission delay of packets approaches the packet transmission time plus a round trip propagation delay.

Fig. 10 can be compared to Fig. 9 in another way. Fig. 10 shows what happens if propagation delays of the networks represented in Fig. 9 are multiplied by 100. In other words, both the packet length and the channel speed, and thus the packet transmission time, are kept fixed. Only the propagation delay is changed. In this case, the unit of time in both figures are identical. Note that the transmission delays in Fig. 10 ($\alpha = 5$) is larger than those in Fig. 9 ($\alpha = 0.05$). Fig. 10 then shows two effects. First, increasing the distances between nodes in the network results in increasing the average transmission delay of packets. This is because propagation delays of the packets are increased, and thus each transmission attempt is more time consuming. Second, saturation of the networks occurs at lower packet arrival rates. Because stations must wait for a longer round trip delay before they can retransmit blocked packets or transmit new packets, stations can not offer as great an intensity of traffic to the network.

The performance of the CAMB tree is also compared with that of the Hubnet (i.e., CASB Tree) through simulations for various traffic configurations in [20]. Results show that the CAMB tree presents a smaller transmission delay and a higher maximum throughput than Hubnet.

Fig. 11 shows performance measurements on an experimental two-level CAMB tree network [12]. This experimental network has a root switch with two switches beneath it. One of the child switches has a station and a traffic generator (used to create traffic on the network) beneath it, while the other child switch has only a traffic generator beneath it. Packets of constant length are generated at each traffic generator at an exponential rate. Fig. 11 shows the average transmission delays as a function of the arrival rate for different traffic localities. The horizontal axes gives the arrival rate (λ) at each traffic generator. From this figure, we observe that a CAMB Tree with a higher degree of locality yields smaller delay (same observation made in Fig. 9 and 10).

5 Reliability of Collision Avoidance Tree

With respect to reliability, the tree topology is susceptible to a node or link failure which may result in blocking communications within (potentially) the entire tree. To avoid this reliability problem, modifications may be introduced in in the Collision Avoidance Tree design.

One approach is to modify the switch design so that the switch short-circuits the Selector and the Broadcaster (Fig. 2) and becomes the root of its subtree, when it does not detect any returning packets from the parent switch. This approach increases the CASB Tree reliability by segmenting the tree. A design of a switch with this fault tolerance mechanism is presented in [1]. In the design, additional circuitry is introduced to monitor whether packets return from the parent switch or not. The time intervals between packets are measured. When a switch detects a time interval that exceeds a certain value, a short burst is sent to the parent switch. If the broadcast of the burst is not received within a delay interval, the switch simply connects its Selector directly to its Broadcaster (loop-back), disconnecting the switch itself from the parent switch. While in loop-back mode, the switch continues to check the downlink from the parent switch. A packet arrival from the parent switch restores the switch to normal operation.

Another approach is to introduce additional links in the CAMB Tree [21]. In this approach, the nodes (switches or stations) are connected to the parent switch (as in the original CAMB Tree architecture) and to the sibling switches (not present in the original CAMB Tree architecture). The station protocol remains the same as in the CAMB Tree. A slight modification is introduced in the switch protocol. At the AR, if the switch is the proper ancestor of a packet, the packet is passed to the DS for broadcast and to the parent switch (same as in the original protocol). Otherwise, the packet is transmitted to the parent switch and *to the sibling switches.* This increases the reliability by providing multiple paths from the station to the common ancestor of a packet. However, this approach is still vulnerable to a switch failure.

Yet another approach is to introduce redundant switches as well as links [22]. This approach introduces a second root switch and links between the original and alternative root switches. The alternative root switch may have additional links to (some of) the switches and the stations on the tree. A timer based mechanism is built in an alternative root switch to activate it in case of a failure. Although this approach is the most costly, it provides best fault tolerance against a link and a switch failure.

6 Variations of the Collision Avoidance Tree

Changes in the CAMB switch protocol may be introduced to improve performance under certain conditions [23]. For instance, in an optical CAMB Tree where channel speeds are high, the packet transmission time may be small compared to propagation delays, and whole packets may exist in transit on a network link. In this case, it may be better not to send packets above their proper ancestor. Sending a packet above its proper ancestor might result in the blocking of packets that would not have aborted the broadcast of the packet anyway.

Additionally, the CAMB Tree requires that packets win complete paths from a source station to a destination station for successful transmission to occur. Whenever a packet fails due to contention, it loses the investment it has in its partially established path. Clearly, as the ratio of the length of the complete transmission path to the packet transmission time increases, the loss of partially established paths becomes more serious. In such a regime, it may be better to buffer whole packets within switches and use a store-and-forward protocol (similar to the semaphore-based protocol described in section 2.2).

Another variation is a Collision Avoidance Non-Broadcast Tree called the Tinker Tree [24]. This is proposed to increase the concurrency in a CAMB Tree. Unlike the other collision avoidance networks, the Tinker Tree is not a broadcast network. The switches in the Tinker Tree route packets along a unique path between source and destination stations.

In the Tinker Tree, switches are connected to each other using full duplex connections same as the uplink-downlink pairs in the CAMB Tree. With a tree topology there is only one path between any pair of source and destination stations. The switches in the Tinker Tree route packets along this unique path using address information contained in packet headers. The Tinker Tree allows for concurrent packet transmissions to the extent that transmission paths do not overlap. Contention along overlapping path segments is resolved by the collision avoidance circuitry in the switches. When two or more packets try to use the same link in the tree, one packet is allowed access and the other packets are blocked. Packets that are blocked are lost (since no buffer is provided at switches) and must be retransmitted by their source stations. A possible Tinker Tree switch architecture is proposed in [23].

Although packet transmissions may be blocked in the Tinker Tree, there are no aborted transmissions. When a station starts seeing a packet arrive, it will see the whole packet arrive. Although [24] does not address the issue of how a Tinker Tree station learns of its packet being blocked, one method that may be used is to have the destination station send a short acknowledgement packet to the source station. The source station then learns of contention failure by timing out on the receipt of an acknowledgment. This acknowledgement could be generated at a higher protocol layer in the network.

In both the CAMB Tree and the Tinker Tree, stations use time-outs to detect transmission failures (i.e., blocking, abortion) due to contention. The CAMB stations use shorter time outs and thus receives faster feedback on the success or failure of their transmissions. In the CAMB Tree, the station waits for its packet to travel to the proper ancestor and back. In the Tinker Tree, the feedback information arrives after two trips between the source and destination stations - one trip for the data packet and another trip for the acknowledgement. Also, in the Tinker Tree there may be some queueing delay at the destination station before the acknowledgement is sent, and this delay must be included in the time-out period.

On the other hand, the discreteness of routing in the Tinker Tree allows greater concurrency than in the CAMB Tree, which supports concurrent transmissions only in non-overlapping subtrees. Essentially, the CAMB Tree uses broadcasting to receive faster feedback, and thus faster transmission retry rates, at the expense of concurrency.

7 Conclusions

In this paper, we described network architectures based on collision avoidance: the Broadcast Star, the CASB Tree and the CAMB Tree. They allow the implementation of a random access protocol without the penalty of collision among packets. The Tree network combines the benefits of random access (low delay when traffic is light; simple, distributed, and therefore robust, protocols) with excellent network utilization and suitability for the domain of high-speed, optical networking. The CAMB Tree has an additional advantage of allowing concurrency of transmissions. This paper presented the station protocols, switch protocols, switch designs, and implementations of the Collision Avoidance Tree. This paper also presented the performance characteristics of the Collision Avoidance Tree through analysis, simulations and measurements.

References

[1] F. Closs and R. P. Lee, "A Multi-Star Broadcast Network for Local-Area Communications," in *Proceedings of the IFIP Conference on Local Networks for Computer Communications*, pp. 61–80, 1981.

[2] A. Albanese, "Star Network with Collision-Avoidance Circuits," *The Bell System Technical Journal*, vol. 62, pp. 631–638, Mar. 1983.

[3] E. S. Lee and P. I. P. Boulton, "The Principles and Performance of Hubnet: A 50 Mbit/s Glass Fiber Local Area Network," *IEEE Journal on Selected Areas in Communications*, vol. 1, pp. 711–720, Nov. 1983.

[4] T. Suda, Y. Yemini, and M. Schwartz, "Tree Network with Collision Avoidance Switches," in *Proceedings of the IEEE Infocom*, pp. 105–113, Apr. 1984.

[5] J. Kurose, M. Schwartz, and Y. Yemini, "Multiple-Access Protocols and Time-Constrained Communication," *ACM Computing Surveys*, vol. 16, Mar. 1984.

[6] M. R. Finley, "Optical Fibers in Local Area Networks," *IEEE Communications Magazine*, vol. 22, pp. 22–35, Aug. 1984.

[7] S. Personick, "Protocols for Fiber-Optic Local Area Networks," *Journal of Lightwave Technology*, vol. LT-3, pp. 426–431, June 1985.

[8] I. Habbab, M. Kavehrad, and C. Sundberg, "Protocols for Very High Speed Fiber Passive Star LANs," in *Proceedings of the IEEE International Conference on Communications (ICC)*, pp. 1593–1600, June 1987.

[9] A. Acampora and M. Karol, "An Overview of Lightwave Packet Networks," *IEEE Networks*, pp. 29–41, Jan. 1989.

[10] B. Abeysundara and A. Kamal, "High-Speed Local Area Networks and Their Performance: A Survey," *ACM Computing Surveys*, vol. 23, pp. 221–264, June 1991.

[11] T. Suda, S. Morris, and T. Nguyen, "Tree LANs with Collision Avoidance: Protocol, Switch Architecture and Simulated Performance," in *Proceedings of the ACM SIGCOMM Symposium*, pp. 155–164, 1988.

[12] H. K. Huang and T. Suda, "Design, Implementation and Measurement of a Collision Avoidance Multiple Broadcast Tree Network," Tech. Rep. 92-37, University of California, Irvine, Apr. 1992. Submitted to the IEEE 17^{th} Conference on Local Computer Networks.

[13] A. E. Kamal, "Star Local Area Networks: A Performance Study," *IEEE Trans. on Computers*, vol. c-36, pp. 483–499, Apr. 1987.

[14] H. S. Hassanein and A. E. Kamal, "Performance Evaluation of Prioritized Collision-Avoidance Star Local Area Networks," in *Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, pp. 236–239, June 1989.

[15] A. E. Kamal, "A Semaphore-Based Access Protocol for Star Networks," in *Proceedings of the IEEE Infocom*, pp. 11–18, Apr. 1987.

[16] S. Morris, T. Suda, and T. Nguyen, "Tree LANs with Collision Avoidance: Photonic Switch Design and Simulated Performance," in *Computer Networks and ISDN Systems 17*, North-Holland, 1989.

[17] K. Goto and T. Suda, "Performance Analysis of a Broadcast Star Local Area Network with Collision Avoidance," in *Proceedings of the IEEE Tenth International Conference on Distributed Computing Systems*, pp. 522–529, May 1990.

[18] H. K. Huang, T. Suda, and Y. Noguchi, "LAN with Collision Avoidance: Switch Implementation and Simulation Study," in *Proceedings of the IEEE* 15^{th} *Conference on Local Computer Networks*, pp. 84–92, Oct. 1990.

[19] E. S. Lee, P. I. P. Boulton, and B. W. Thomson, "Hubnet Performance Measurement," *IEEE Journal on Selected Areas in Communications*, vol. 6, pp. 1025–1032, July 1988.

[20] S. Marano and A. Volpentesta, "Performance Evaluation of Alberonet by Simulation and Theoretical Analyses," in *Proceedings of the IEEE Infocom*, pp. 1137–1147, Apr. 1987.

[21] D. Saha and P. Dhar, "Fault-Tolerant Architectures for Tree LANs," *Computer Communications*, vol. 14, pp. 244–248, May 1991.

[22] W. Burr, "A Fault-Tolerant Hierarquical Broadcast Network," in *Proceeding of the Computer Networking Symposium*, pp. 11–17, Apr. 1984.

[23] T. Suda and S. Morris, "Tree LANs with Collision Avoidance: Station and Switch Protocol," in *Computer Networks and ISDN Systems 17*, pp. 101–110, North-Holland, 1989.

[24] Y. Yemini, "Tinkernet: or, Is There Life Between LANs and PBXs?," in *Proceedings of the IEEE International Conference on Communications (ICC)*, pp. 1501–1505, June 1983.

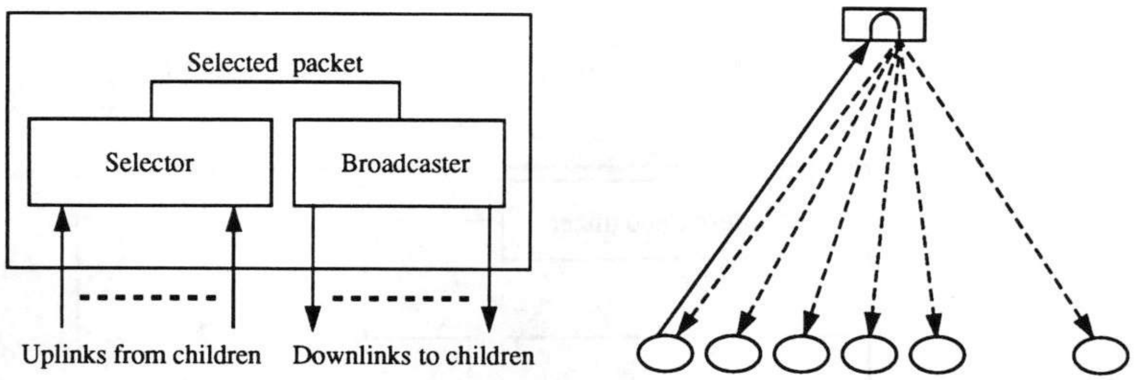

Figure 1: Broadcast Star and its Switch

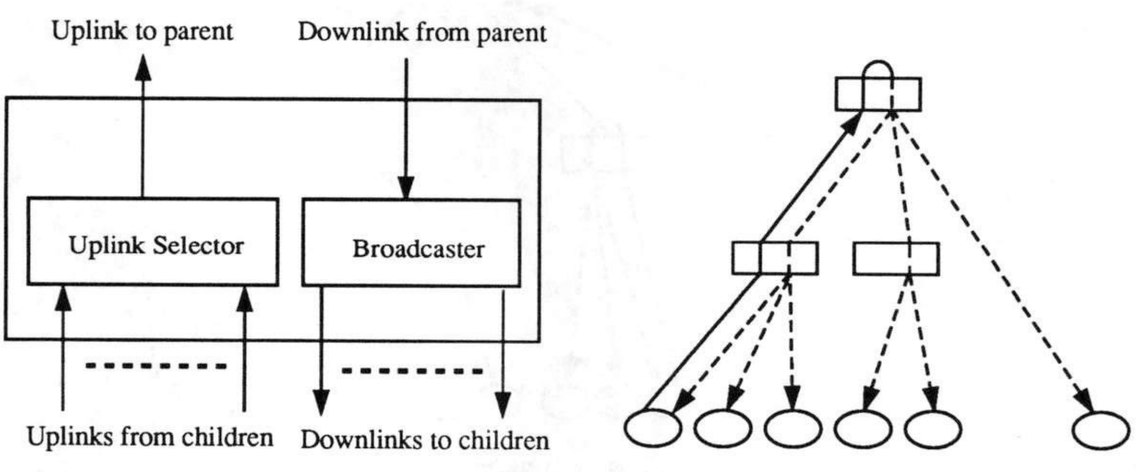

Figure 2: CASB Tree and its Switch

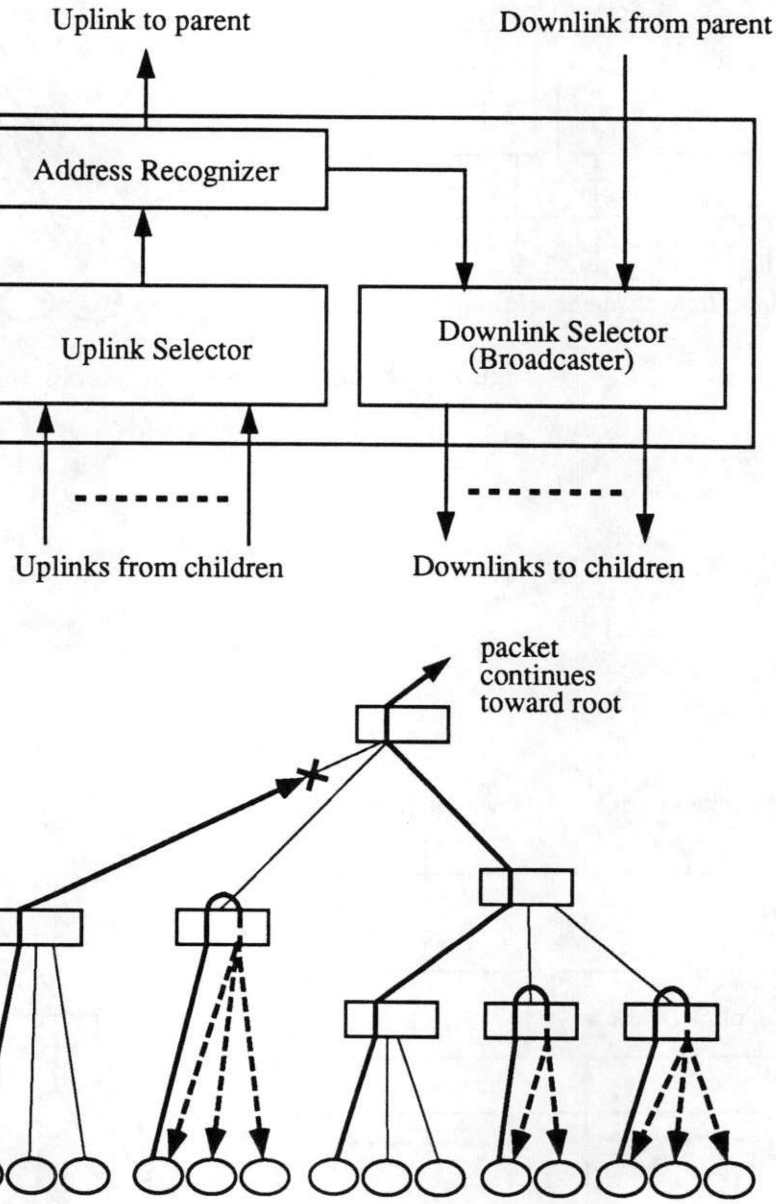

Figure 3: CAMB Tree and its Switch

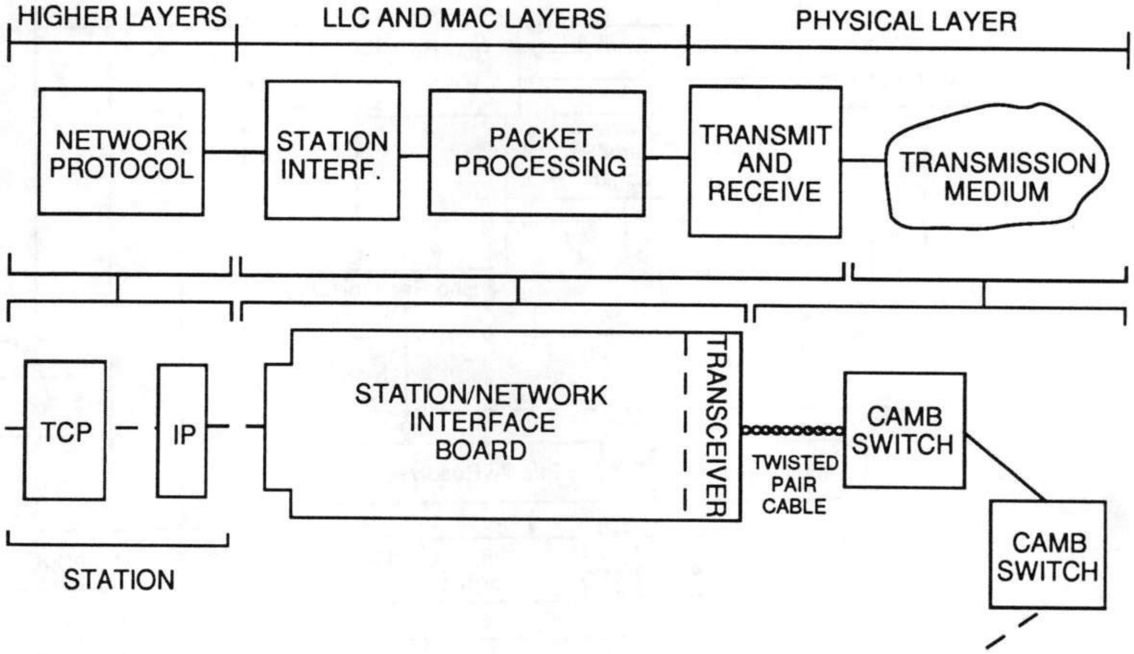

Figure 4: Layering Architecture of a CAMB Tree Implementation

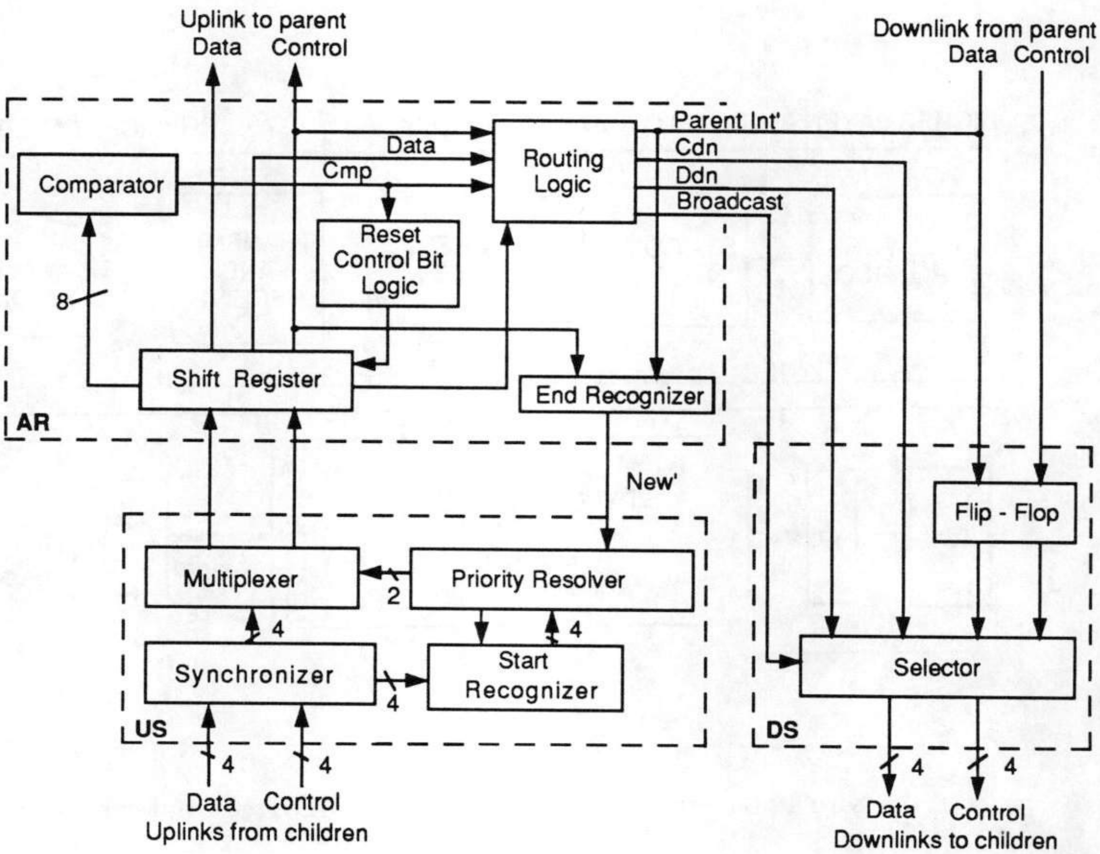

Figure 5: CAMB Switch Design

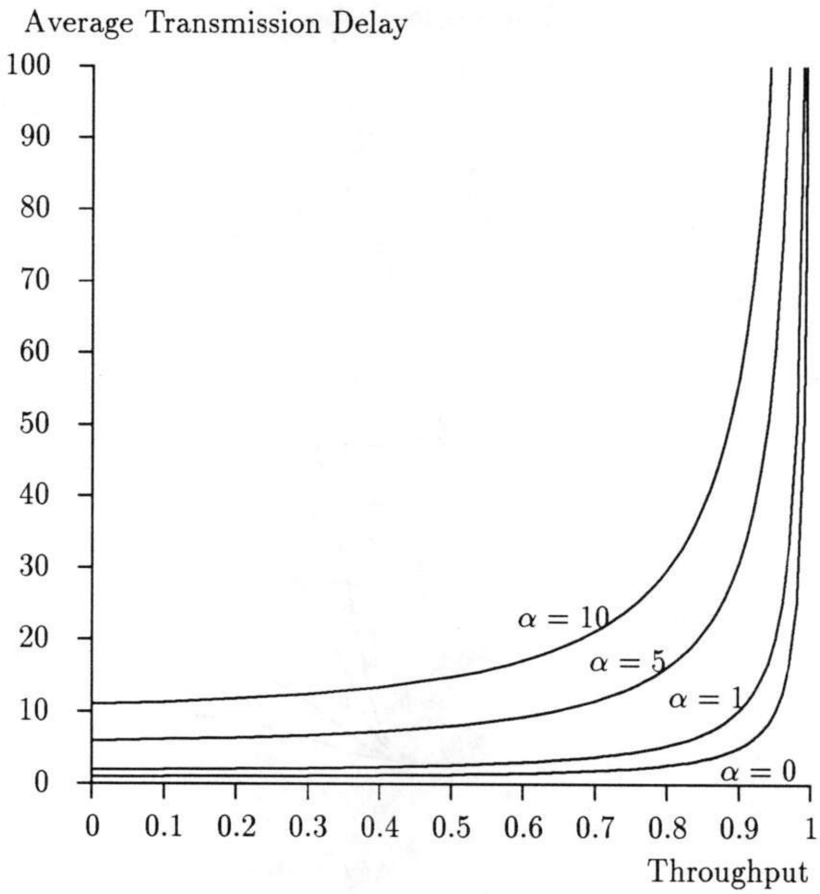

Figure 6: Transmission Delays in a Broadcast Star with Infinite Population

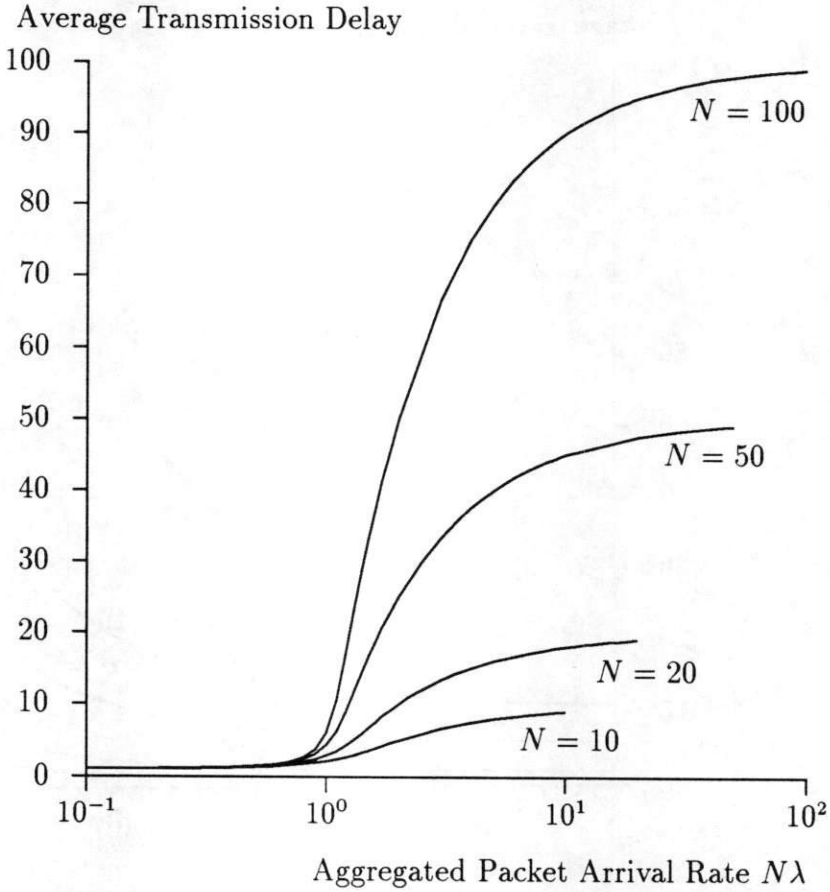

Figure 7: Transmission Delays in a Broadcast Star with Finite Population ($\alpha = 0.05$)

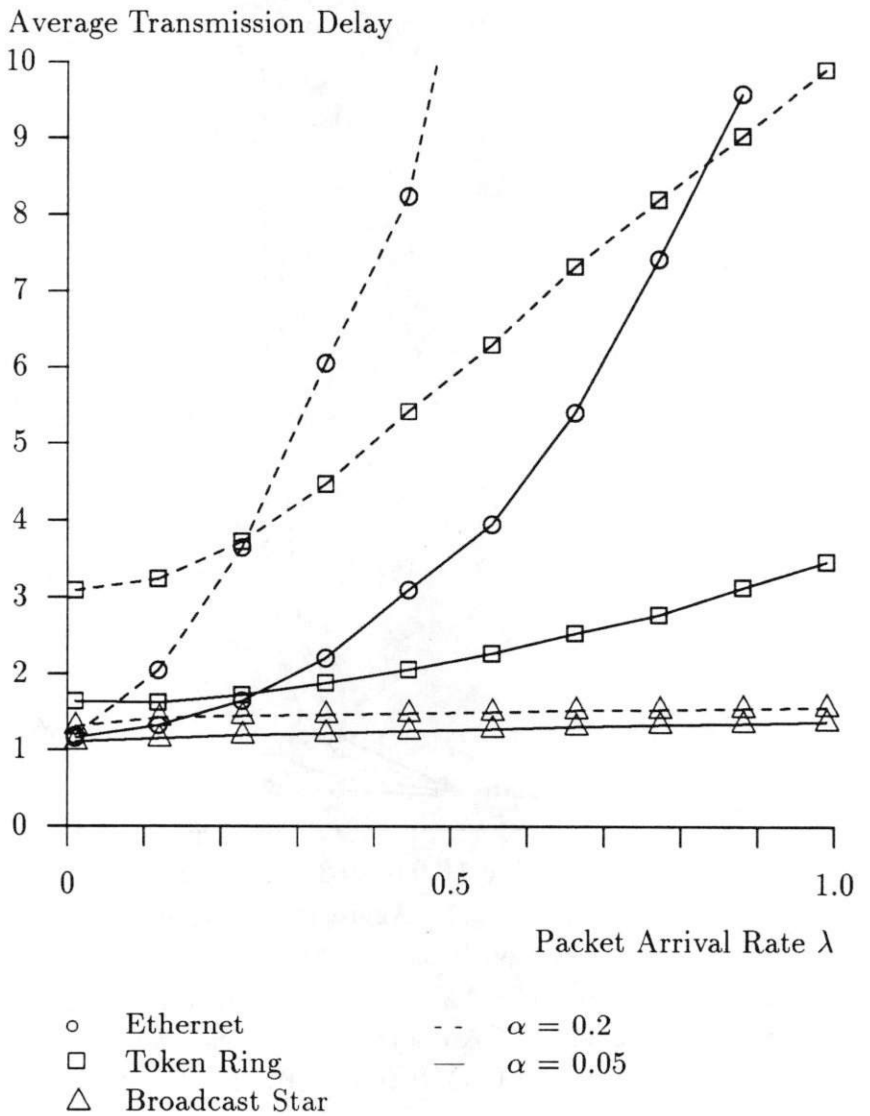

Figure 8: Comparison of Broadcast Star, Token Ring and Ethernet

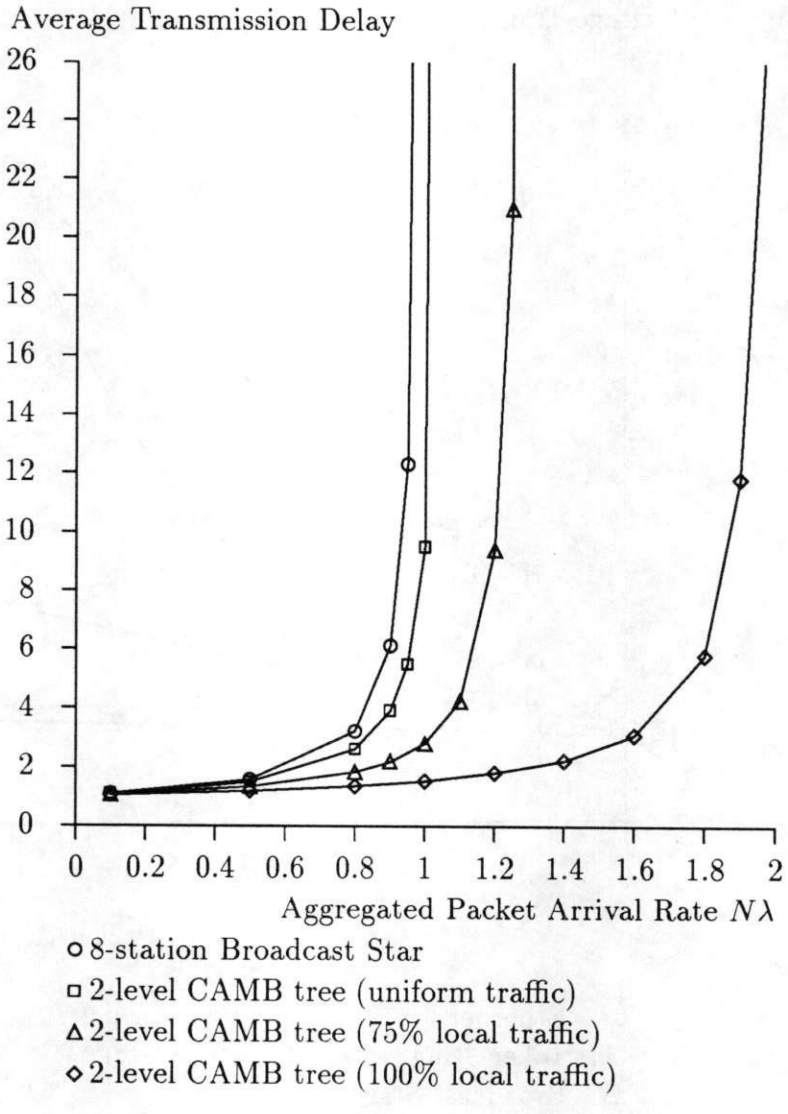

Figure 9: Transmission Delays in a CAMB Tree ($\alpha = 0.05$)

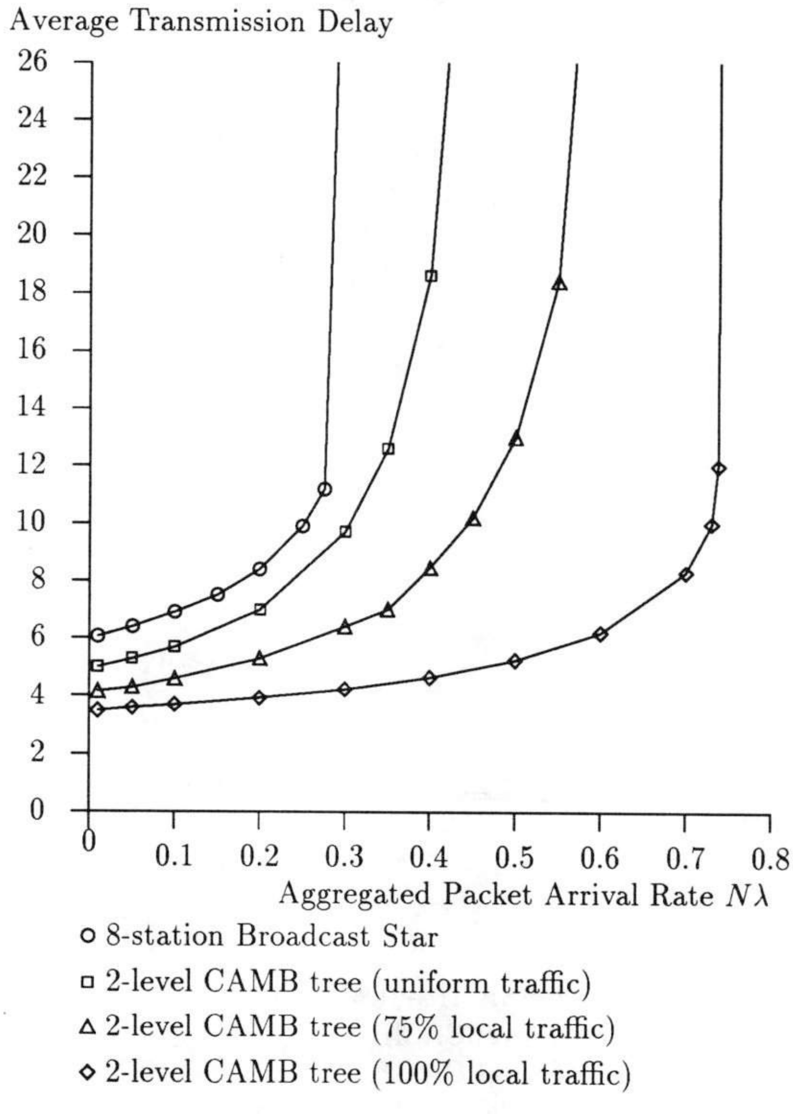

Figure 10: Transmission Delays in a CAMB Tree ($\alpha = 5$)

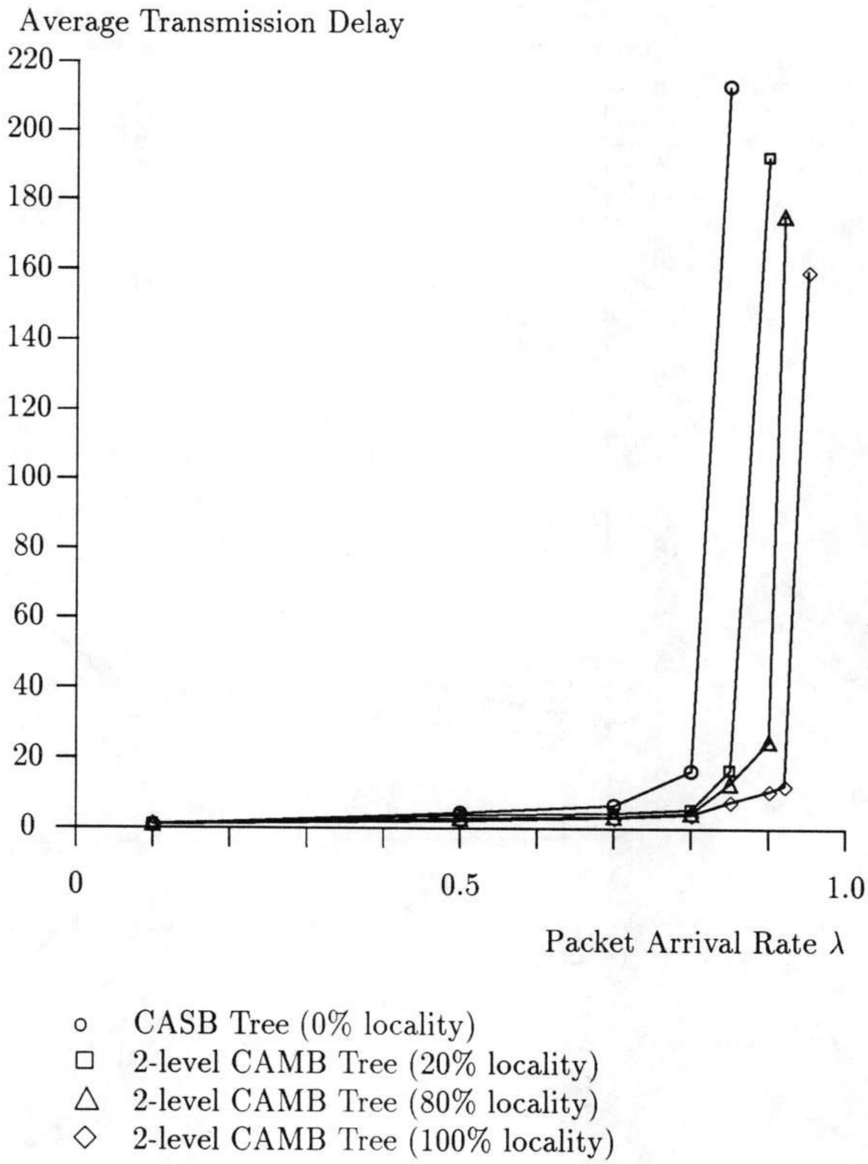

Figure 11: Transmission Delay in a 2-Level CAMB Tree ($\alpha = 0.006$)