INTEGRATION OF VOICE COMMUNICATION ON
A SYNCHRONOUS OPTICAL HYPERGRAPH

Yoram Ofek*

IBM
T. J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Abstract

The optical hypergraph is a novel network architecture in which each edge of the hypergraph is a multiple-access broadcast medium constructed as a passive optical star coupler. Access to each net (edge) is time-slotted, and the system maintains global slot synchronization. The integration of voice into the system is done by reserving time slots in a periodic manner. A packet which contains several voice parcels from different phone conversations is transferred in these slots. These parcels may have different destinations on the optical net. As a result of the global end-to-end synchronization, the delay from the source to the destination is a known constant, with accuracy of plus or minus half a time slot. In the analysis, it is shown that the system improves its operation as the communication bandwidth increases. In other words, the algorithms and protocols improve in performance as the communication bandwidth increases. Two criteria are used to exhibit this phenomenon (i) the utilization efficiency and (ii) the end-to-end delay.

1. Introduction

The integration of real-time communication with digital data communication is of growing interest (see [FiToSb], [Grubs 11, and [MontSS]) for several reasons: Networks' bandwidths are growing rapidly, so the excess capacity can be used for telephony or video, which saves the construction of separate networks. Digitized voice telephony has a better quality than the analog one. At present, and even more so in the future, there will be a demand for an off-line digital recording over a network of voice and video for future use. The basic principles for integration of voice into a large area network (thousands of nodes within an area of thousands of square kilometers) is presented in this paper. It is shown that the integration is more efficient and effective as the net bandwidth increases. In general, real-time communication constraints introduce additional complexity to communication management. However, real-time communication on a synchronous hypergraph network, proposed in [Ofek87c], introduces little additional complexity, and is an immediate result of the global event synchronization. Global system synchronization enables a well-defined global state transition at the end of each time slot, which simplifies the implementation of other distributed network control algorithms, such as access control, routing, and buffer management. The integration of distributed concurrency control algorithms into the system is done using time stamps, which enables to maintain total event ordering [McOfS7].

Section 2 describes some basic properties of the synchronous hypergraph; the emphasis on two-dimensional regular and partial hypergraphs. In Section 3, a protocol for integrating voice communication into a two-dimensional regular hypergraph is presented. The analysis of the communication efficiency and the end-to-end delay as a function of the communication bandwidth is discussed in Section 4. In Section 5 some conclusions and further considerations are discussed.

2. The Synchronous Hypergraph

2.1 Hypergraph Architecture

A hypergraph network is a set of nets; each net is a set of two or more nodes. The nets are exactly the edges of a hypergraph, and the nodes are its vertices [Ullm84], for a more formal definition see [Berg73]. Each node has ports to one or more nets, or buses. This type of system is also called a bus-based topology or bus-connected architecture [Witt81].

The topology of each net is a passive optical star. Communication transmissions from the net's nodes merge to one point in space, a centralized passive star coupler, where they are broadcasted back to all the nodes on the net. The transmission rate over the net is assumed to be one gigabit/second. The net has a cyclic symmetry, and the nodes are indistinguishable in their spatial location with respect to the center point.

*Parts of this work were done while the author was at the University of Illinois - Urbana.
2.2 Basic Topologies

Two basic hypergraph topologies are considered in this work:

1. Two-dimensional regular hypergraph (2D-R) - each net has \( n \) nodes, and each node has ports to two different nets. Figure 1a depicts a 4-by-4 regular hypergraph as a grid of orthogonal buses in the \( x - y \) plane. Figure 1b shows each of the six centralized optical stars that actually comprise a 3-by-3 regular hypergraph.

2. Two-dimensional partial hypergraph (2D-P) - each net has \( n \) nodes, such that (i) \( k \) nodes have ports to two different nets, (ii) \( a \) nodes have only one port, and (iii) \( k + a = n \). Figure 2 shows possible layouts of 2D-P hypergraphs. Each net is represented by a narrow rectangle. Note that since every net is a centralized optical star, the nodes (vertices) are indistinguishable with respect to the star's center. Therefore, the \( a \) nodes of each net can be arranged as shown in Figure 2b or Figure 2c. Partial hypergraphs represent an important class of hypergraph topologies. While retaining much of the uniformity of a regular hypergraph, many geographical distributions of nodes can be mapped onto a partial hypergraph.

2.3 Time Partitioning

The communication over each net is time-slotted, and each node's interface to the network has a slot counter for measuring the time. Slots are grouped into frames, and there are \( f \) slots per frame, as shown in Figure 3. The duration of one time frame is larger than the maximum delay between any two nodes on a net. For the remainder of this paper we shall assume that a frame consists of a single slot, implying that the first bit of a slot has arrived at all nodes on a net before the last bit has been transmitted.

Each slot is subdivided into \( r + 1 \) minislots, as shown in Figure 3. The first \( r \) of these are very short control minislots (CMSs), and the last is a data minislot (DMS). During the DMS, which occupies most of the time slot, one packet of data is sent from one node. The set of \( n \) nodes on a net is partitioned into \( r \) disjoint subsets, each with cardinality \( \frac{n}{R} \) or \( \frac{n}{R} + 1 \). The sum of the subset cardinalities is exactly \( n \), and each node belongs to exactly one of the subsets. One of the \( r \) CMSs is assigned to each subset. The nodes in a subset access the CMS deterministically, in a round-robin order. Because each node is guaranteed access to a CMS periodically, CMSs provide a way for nodes to exchange control, state and timing information in short, bounded time. Specifically, this bound is \( r + 1 \) \( T_s \), where \( T_s \) is the time slot duration.

2.4 Global Synchronization Condition

The following is a necessary condition for synchronization, or what is implied when we are saying that the system is synchronized. In the next section algorithms for net and global synchronization are presented. The time is slotted into predetermined intervals, each node has a slot counter which function as the node's own global clock.

Global Synchronization Condition - For any two nodes \( i \) and \( j \), and for any slot counter value \( k \), there exists a non-zero time interval such that \( C_i - C_j = k \) (\( 1 \leq i, j \leq n \)). Figure 4 presents a global view of the system's timing. During the time interval \( d \) (\( 0 < d < T_s \)), all the slot counters are incremented.
The global system state at time $k$ ($S^k$) is defined as the concatenation of the local state of all the system's nodes at time $k$ ($\{s^i\}$). The width of the global state transition from $S^k$ to $S^{k+1}$ is the time between the first local transition to state $k+1$ to the last local transition to state $k+1$. This time is measured by an external observer, and is defined to be $W_{\text{max}}$. A global state transition is said to be well-defined if the maximum transition width ($d = W_{\text{max}} = \max \{W_{\text{max}}\}$) is strictly less than the time slot duration. A well-defined state transition simplifies the operation of distributed network control algorithms (e.g., routing, buffer management), and the concurrency control of parallel processing transactions, that is, time stamps can be used for preserving the serialization of parallel algorithms.

### 2.5 Synchronization Algorithms

The following are the local and global synchronization algorithms. Details of the global synchronization algorithm for the optical hypergraph, with analysis that shows its correctness and high efficiency, may be found in [OFFa87].

#### Net Synchronization

Each time slot has a fixed time duration $T_s$, measured in bit periods. Nodes on a net use the center of the star, and their respective distance from it, as their basic timing reference. Therefore, it is required that the star coupler be constructed in a very small area in space. The one-way delay from the center of the star to node $i$ is $\Delta_i$, as shown in Figure 5. The $n$ nodes of each net may be regarded as lying on the circumference of an imaginary circle of radius $R$, such that $T_s \geq \max \{\Delta_i\}$, for $n \geq i \geq 1$. That is, $T_s$ is an upper bound on the delay from the center of the star to each node. In order to maintain net synchronization, each node $i$ determines the beginning of the next time slot by adding ($T_s - 2\Delta_i$) to the time it receives the first bit of the first CMS.

### 3. Voice Integration Protocol

#### 3.1 Telephony Characteristics

A typical sampling rate for voice communication is every 125 microseconds or 8,000 samples per second. Analog samples are converted into 8-bit bytes, which are grouped into voice-parcels. The size of a typical voice-parcel is between 80 and 400 bytes of speech, corresponding to 10-50 milliseconds of speech. The duration of a typical telephone conversation is about 200 seconds.

The following parameters will be used:
- $T_s$ - the duration of speech which is sampled into one parcel,
- $BW_{\text{max}}$ - the bandwidth of a single net,
- $T_{\text{MDS}}$ - the duration of one data minislot, and
- $T_{\text{trans}}$ - the transmitting duration of one voice-parcel.
3.2 Principles of Voice Integration

The integration of voice into the system is achieved by incorporating the following features to the optical architecture:

(1) Cycle - the time is divided into cycles of $c$ slots each. The duration of each cycle is exactly $T_p$, the time which one voice-parcel is generated. The number of time slots in every cycle is

$$c = \frac{T_p}{T_{PAR}}$$

(2) Reservation Register - each node has a slot reservation register of $c$ cells, with 2 bits per cell, defined as follows:
- 00 - the slot is free
- 01 - the slot is reserved by another node
- 10 - the slot is mine
- 11 - the slot is reserved for data communication

Free slots can be reserved for voice communication or can be used for data communication. Slots reserved for data communication cannot be reserved for voice. Note that the access control mechanism continues to operate as previously described, but it uses only the free slots and slots which have been reserved for data communication.

(3) Virtual Multiplexing - each data minislot (DMS) is subdivided such that each subsection will contain one voice-parcel. The number of voice-parcels in every DMS is

$$p = \frac{T_{DMS}}{T_{PAR}}$$

These $p$ parcels are sent from one origin but can have different destinations on that same net.

(4) Virtual Demultiplexing - each port of the net, which receives the broadcast packet with the $p$ voice-parcels, extracts those which are sent to it and discards the rest. It is done on-line by checking an 8-bit logical address header of each parcel.

(5) Flow Condition - in order to guarantee the real-time flow of a telephone conversation between two parties, every voice-parcel is transferred across one net in one cycle. Thus, voice-parcels cannot be accumulated and are guaranteed to reach their destinations in a fixed number of cycles.

3.3 The Protocol

The voice protocol has three phases: (i) dial-up for constructing a bidirectional virtual path between the two parties; (ii) conversation, in which voice-parcels are transferred every cycle between the two parties; and (iii) termination of the conversation. The protocol is for 2D-R, but can be easily extended to other synchronous hypergraphs.

The telephones are interfacing the host bus, which resides outside of the network interface. The telephone interface may accommodate several telephones. The voice protocol is performed by the network interface, which receives and transfers speech voice-parcels from/to the telephone interface.

The protocol uses only primary routes, which utilize the least communication capacity of the net. On a 2D-R, a phone conversation involves three nodes, as shown in Figure 6. The phone conversation is initiated via the SRC-node, the voice-parcels are sent on the SRC-net to the INT-node, and from there via the DST-net to the DST-node. The destination party is connected to the DST-node.

Figure 6. Voice on a 2D-R Hypergraph

3.3.1 The dial-up phase

The first step, when a phone on a SRC-node calls a phone on a DST-node, is to determine the primary route, i.e., to select the INT-node. The objective is to reserve a virtual path from the SRC-node to the DST-node and back. The transfer over each part of this path (net) is multiplexed by several conversations. Reservation means that the port interface to the net allocates a free space for one speech parcel in a DMS, which this port has reserved for its voice communication. If the port does not have a free space in a reserved DMS, it tries to reserve a free slot, and then it will allocate a free space for the parcel. If no free space is available, the dial-up procedure fails, and a "Line Busy" message is sent to the phone at the SRC-node.

Step 1 - the SRC-node tries to allocate a free space for a voice-parcel on the SRC-net. If the space allocation is successful, a dial-up message is sent to the INT-node; otherwise a "Line Busy" message is sent to the requestor. Step 2 - the INT-node tries to allocate a free space for voice-parcels on the SRC-net and on the DST-net. If the space allocation is successful, a dial-up message is sent to the DST-node, otherwise a "Line Busy" message is sent
to the requester via the SRC-node, and the SRC-node deallocates the free space already assigned.

Step 3 - the DST-node tries to allocate a free space for a parcel on the DST-net. If the allocation is successful, a "Line Ready" message is sent back to the SRC-node via the INT-node, otherwise a "Line Busy" message is sent back to the SRC-node via the INT-node, and both the INT-node and SRC-node deallocate the free spaces already assigned.

3.3.2 Conversation phase

For voice integration, the system has two levels of synchronization: (i) an event or a slot, and (ii) a cycle which starts whenever the value, modulus \( c \), of the slot counter is one; i.e., the cycles of all the nodes are in phase. As a result, an end-to-end voice synchronization is maintained, and the speech parcels which arrive at their destination can be converted back into voice at a constant rate (with maximum error of only \( \pm 0.5T_s \)).

Each source and destination telephone generates one parcel per cycle. Since the dial-up procedure guarantees that there is a space for each parcel during each cycle, a voice-parcel is transferred from SRC-node to DST-node in two cycles. The INT-node will not transfer a voice-parcel to the DST-node in the same cycle it has been received; therefore, the parcels ordering at the DST-node are preserved.

The transmission of voice, during the conversation phase, is via the virtual multiplexing/demultiplexing mechanism, performed in two steps:

Multiplexing - a node broadcasts a packet of data during the DMS with several speech parcels, each with different destination on the net. The parcels are packed into the packet with no specific order.

Demultiplexing - each node on the net receives a data packet and extracts the parcels with its destination and discards the rest.

The INT-node extracts parcels from the traffic on the SRC-net with a destination to the DST-node. Then transfers them to its orthogonal port, which sends them to the DST-node via the DST-net.

3.3.3 Termination phase

The termination procedure frees the space allocated for the conversation during the dial-up phase. The SRC-node sends a "Terminate" message to the INT-node and then to the DST-node.

4. Bandwidth effect on Performance

In the following discussion it will be shown that the 2D regular optical hypergraph improves its performance as the medium bandwidth increases. This property is of special importance in the design of very high bandwidth networks. There are cases in which an increase of the bandwidth will decrease the efficiency. Two criteria are used to demonstrate this property, (i) the reservation inefficiency and (ii) the end-to-end delay.

Some numerical examples are presented which use the following parameters:

- The speech is digitized by a sample of 8 bits every 125 microseconds
- \( T_s = 25,000 \) microseconds (the duration of speech which is sampled into one voice-parcel of 1,600 bits)
- \( BW_{net} = 10^6 \) bits per second (the bandwidth of a net)
- \( T_{DMS} = 32 \) microseconds (the duration of one data minislot)
- The duration of one parcel \( T_{PAR} = \frac{T_p}{125} \frac{8}{BW_{net}} = \frac{25,000}{125} \frac{8}{10^3} = 1.6 \) microseconds
- The number of parcels in one data packet (during one DMS) \( p = \frac{T_{DMS}}{T_{PAR}} = 20 \)
- The number of slots per cycle \( c = \frac{T_p}{T_s} \approx 700 \)
- \( n = 100 \) nodes per net

4.1 Reservation Inefficiency

Reservation inefficiency is defined as the communication capacity which is wasted because of the voice reservation mechanism. Assume that the average utilization of a reserved slot is 0.5, then every port on a net might waste an average half a DMS. Hence, the reservation inefficiency \( \xi \) is

\[ \xi_{RESERV} = 0.5 \frac{n}{c} \]

For the above example it is

\[ \xi_{RESERV} = 0.5 \frac{100}{700} = 0.07 \]

Let \( l \) be the size of a slot in bits, and

\[ T_s = \frac{l}{BW_{net}} \]

then the number of slots in one cycle \( c \) increases linearly as the \( BW_{net} \) increases

1B.3.5.
4.3.2 Scheduling by Path-length

Another possibility is to use the information on the remaining path-length of the voice-parcels for its scheduling. The voice traffic at each port is divided into two types:

(i) type 1 - parcels which have one net to cross, and
(ii) type 2 - parcels which have two nets to cross.

The parcels of type 2 are grouped into data packets which are scheduled to be transmitted at the beginning of the cycle. The parcels of type 1 are grouped into data packets, which are scheduled to be transmitted after the all packet with type 2 parcels, from all the system's nodes, have been transmitted and reached their destinations. In other words, first all the nodes transmit all their packets with type 2 parcels, and then transmit the packets with type 1 parcels.

Note that an incoming type 2 parcel will become type 1 parcel on the orthogonal net.

Under this scheme the delay for crossing two nets is less than a cycle. If the voice communication is only a small fraction of the total communication, then the delay will be a small fraction of the cycle time. In order to minimize the delay under this scheme, it is necessary to balance the voice load among all the network's nets. The reservation inefficiency in this scheme is twice as large as the original one. Again, this efficiency will improve as the bandwidth increases.

5. Conclusions

The infiltration of fiber optics into communication networking has manifested itself in two ways. In the first, optical fibers have been used as high bandwidth wire replacements. The functionality of the network has remained basically the same. In the second approach, the unique characteristics of the optical fiber media have been exploited in various facets of network architecture. These include network topology, access control, and the effective use of the high bandwidth in a distributed system design.

The latter approach has been used in this design, which exploits the virtues of the passive optical star. Potential applications of the optical hypergraph network are varied and include large distributed databases, metropolitan area networks, factory automation, and real-time networks. This work has presented a straightforward and efficient way to integrate telephony communications into a synchronous hypergraph.
References


1B.3.7.