CHAPTER 9.

INTEGRATION OF VOICE ON A SYNCHRONOUS HYPERGRAPH

9.1. Introduction

The integration of real–time communication with digital data communication is of growing interest (see [FiTo86], [Grub81], and [Mont83]), 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 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, so real–time data transfers over a large network are an actual necessity. In this chapter it will be shown that the integration of voice on an optical hypergraph architecture is more efficient and more effective as the net bandwidth increases.

In general, real–time communication constraints (the necessity to complete the communication in a bounded time) introduce additional complexity to communication management. However, real–time communication on a synchronous hypergraph network introduces little additional complexity and is an immediate result of global event synchronization.

The voice transmission is packetized and sent over the net by constructing virtual multiplexers/demultiplexers. It is done without changing any of the previous design elements. Each telephone conversation has a very low bandwidth. Therefore, it is important to be able to manage and transfer efficiently small blocks of data.

The following discussion presents the basic protocols for integrating voice into the network. The scope of the discussion is limited to the principles of operation and to the analysis of the integration. This discussion is limited to two–party conversations.

9.2. Telephony Communication Characteristics

A typical sampling rate for voice communication is every 125 μseconds 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 160 and 400 bytes of speech, corresponding to 20–50 milliseconds of speech. The duration of a typical telephone conversation is about 200 seconds (3 minutes).

The following parameters will be used:

- T_p – the duration of speech which is sampled into one parcel,
- BW_{net} – the bandwidth of a single net,
- T_{DMS} – the duration of one data minislot,
- T_{PAR} – the transmitting duration of one voice–parcel.

9.3. Principles of Voice Integration

The integration of voice into the system is achieved by adding 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 $c = \dfrac{T_p}{T_{PAR}}$.

(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.

9.4. The Voice 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 9.1. 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.

9.4.1. 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 of following steps 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

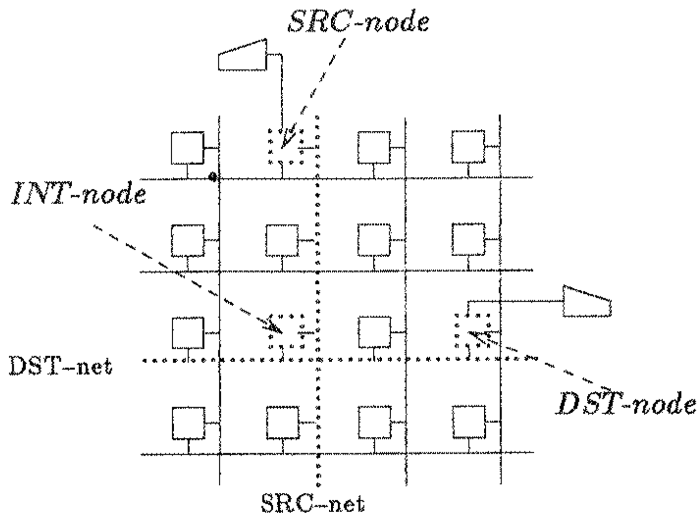

Figure 9.1: Integrating Voice on a 2D Regular Hypergraph

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 requester.

Step 2 - the INT-node tries to allocate a free space for two 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.

9.4.2. 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, modulo 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 in a constant rate (with maximum error of only $\pm 0.5 T_s$).

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.

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 and transfers them to its orthogonal port which sends them to the DST–node via the DST–net.

9.4.3. 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.

9.5. Analysis

Analysis evaluates the consequences of the voice integration on a 2D–R synchronous hypergraph. The analysis uses the parameters previously defined.

Example:–

For the discussion some numerical results are presented which use the following parameters:

- The speech is digitized by a sample of 8 bits every 125 μseconds (the bandwidth is 8,000 bytes per second)
- $T_p = 25{,}000$ μseconds (the duration of speech which is sampled into one voice–parcel of 1,600 bits)
- $BW_{net} = 10^9$ bits per second (the bandwidth of a net)
- $T_{DMS} = 32$ μseconds (the duration of one data minislot)
- $T_{PAR} = \frac{T_p}{125} \frac{8}{BW_{net}} = \frac{25{,}000}{125} \frac{8}{10^3} = 1.6$ μseconds (the duration of one parcel)
- $p = \frac{T_{DMS}}{T_{PAR}} = 20$ parcels in one data packet (during one DMS)

- $c = \dfrac{T_p}{T_s} \approx 700$ slots per cycle

- $n = 100$ nodes per net

9.5.1. End–to–end delay

The end–to–end delay is measured in cycles (T_p) and is defined as the time interval between the analog/digital conversion (ADC) and the digital/analog conversion (DAC). It takes one cycle for ADC of one parcel; at the end of the cycle the parcel is at the SRC–node. During the next cycle the parcel arrives at the INT–node and at the DST–node during the third cycle. The DAC, for regenerating the voice–parcel, can then start. Thus, it takes three cycles for a voice–parcel to get from the source phone to the destination phone; e.g., for $T_p = 25$ milliseconds the end–to–end delay is 75 milliseconds.

The time for internal data transfers within the interface is ignored since it is assumed to be much smaller than the cycle time. Also, the frame delay is ignored since it is only a few slots much less than the 700 slots of one cycle.

9.5.2. 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, so every port on a net might waste an average 0.5 DMS. Hence, the reservation inefficiency (ξ) is

$$\xi_{RESERV} = 0.5\frac{n}{c}100(\%).$$

For the above example it is $\xi_{RESERV} = 0.5\frac{100}{700}100 = 7\%$.

Note: (i) The wasted space is actually free and can be used by the node for data transfers or for dial–up and termination messages. (ii) The reservation inefficiency decreases as the number of nodes decreases. (iii) The number of slots in one cycle (c) increases linearly as the BW_{net} increases; therefore, in a higher communication bandwidth the reservation inefficiency decreases (assuming that all other parameters are not changed). This last result is especially important since it indicates a direct advantage for using channels with a higher bandwidth.

9.6. Routing by Local Balancing

The routing algorithm is performed by the SRC–node before beginning the dial–up protocol. In general, telephony communication is easier to balance, since each conversation duration is about four orders of magnitude longer than the time slot. Therefore, load balancing can be viewed as a static procedure rather than a dynamic one.

The proposed routing algorithm has the following principles:

(i) each node has a reservation register at each port,

(ii) each node does not reserve more than it actually needs,

(iii) only primary routes are considered,

(iv) reservation criterion – the selection between the two primary routes is made such that the total number of reserved slots on the two orthogonal nets is the same, i.e., the two reservation registers have reserved the same number of slots.

The major advantage of this procedure is that it can be done in a completely distributed manner and without exchanging any additional information. If the nets systems are more or less uniform, this algorithm may have an efficient performance.

9.7. Discussion

The integration of voice telephony into the synchronous optical hypergraph has been shown to be straightforward. From the hardware point of view there are two additional requirements: (i) reservation register of c cells at each port, and (ii) comparator for the virtual demultiplexing.

The low-dimension, high-bandwidth hypergraph has two major advantages over higher dimension hypergraph (or point-to-point network):

(1) The end-to-end delay of the voice-parcel is proportional to the length of the path, which is measured by the number of nets in the path. Therefore, a lower dimension with fewer nets in its path results in a shorter end-to-end delay.

(2) The inefficiency for the integration of voice is smaller as the bandwidth gets higher, as shown in the following equations:

$$\xi_{RESERV} = 0.5\frac{n}{c}100(\%).$$

Since, (i) $c = \frac{T_p}{T_s}$, (ii) r is the size of a slot in bits, and (iii) $T_s = \frac{r}{BW_{net}}$, the ξ_{RESERV} is

$$\xi_{RESERV} = 0.5\frac{n}{T_p}\frac{r}{BW_{net}}100(\%).$$