CHAPTER 8.

FAULT TOLERANCE OF THE OPTICAL HYPERGRAPH

The basic principles which enable this optical architecture to tolerate physical failures are outlined in this chapter. The scope of the fault tolerant (FT) enhancements includes functions performed by the network interface and not FT functions performed by the hosts. Furthermore, the realization of some FT functions by the interface reduces the complexity of the distributed software.

The basic architectural and operational principles of the system are very suitable for fault tolerant enhancements. The two most important properties are

(1) **Passive optical star** – this centralized topology is extremely suitable for fault tolerant architecture because

(i) The net can tolerate multiple link and/or coupler failures (i.e., graceful degradation of performance).

(ii) The broadcast transmission over the net is redundant in nature. As a result, all messages are received by all the net's ports. In general, much of the state and time information, received by each node during the CMSs, is redundant.

(iii) The net has a single time reference, which enables distributed net synchronization and well–defined state transitions at the end of every time slot.

(2) **Periodic exchange of state information** – this mechanism is equivalent to exchanging "*I am alive*" messages in a synchronous and deterministic manner, and any deviation from this behavior signals a failure. Note that the actual

information being exchanged determines what failures can be tolerated.

Fault tolerance (FT) is the surviving attribute of a system. The following discussion emphasizes how the system can tolerate physical failures rather than man–made faults; for FT taxonomy see [Aviz78]. The system consists of multiple computing and communicating nodes the objective being that a failure of some nodes will not disable the whole system. Thus, the basic FT methodology, a graceful degradation of performance, is completely distributed; i.e., no "*king*" is used in the fault tolerant procedures. As a result, the FT mechanism exhibits the equivalence between a regular and partial hypergraph; i.e., the regular hypergraph can be gracefully degraded in a distributed manner into almost any arbitrary partial network.

8.1. The Fault Tolerance Objectives

There are two major reasons for making this system capable of tolerating failures:

(1) Global Event Synchronization, in which the fundamental attribute of this system makes the system more vulnerable. A single synchronization failure may cause an entire system to fail. It will be shown that this unique optical architecture is, indeed, capable of overcoming multiple synchronization failures in a complete distributed manner.

(2) Lossless, the decreased probability of a **packet loss** whereby communication reliability is increased. A packet, transferred to the network interface, would not be lost without leaving some traces. A successful data transfer occurs when (i) the packet has reached its destination and left the network interface, or (ii) the packet was detected as faulty by the **sending side** (self check by each port) and

then retransmitted by the interface, or (iii) the packet can not reach its destination because of some port failures, and a message is then returned to the packet's origin. A loss of packet occurs when the packet is transferred to the network interface and then **"disappears"** without any **direct** trace. Only a higher-level communication protocol, at the destination host, can indirectly detect a missing packet and then send a special retransmission request.

8.2. Methodology

In general, a fault-tolerant procedure may have three steps: (i) fault detection – achieved by adding redundant parameters (e.g., extra bits in the broadcast control messages), (ii) fault diagnosis – determining the source of the failure, and (iii) fault recovery – continuation of the system's regular operation. In the following analysis the emphasis is on the individual net. Once a net is FT and the connection between the two ports of the interface is made FT, the whole system can then be made FT.

The method for fault-tolerant of the global event synchronization and time-stamping is by using fault masking; i.e., majority voting in real-time. This method uses the inherent redundant nature of broadcasting over the optical star.

The fault-tolerant methodology for preventing packet loss is based on the ability to detect faulty ports within a very small delay. Once a faulty port has been detected, the net gracefully degrades its operation by isolating and bipassing the faulty port from the net.

The FT analysis proceeds **bottom-up**, as shown in Figure 8.1. The lower level and the basis of the FT analysis is the transmission medium (optical star). The medium is analyzed with respect to physical disconnection and for external interference (noise).

The basic FT principle of the system is **"isolate and skip."** A port which diagnoses itself as faulty will **isolate** itself from the net. Due to the deterministic, periodic exchange mechanism, all other ports can detect idle or faulty ports and **skip** them. The last step models some possible failures in the synchronization and packet transfer operations and then presents solutions for overcoming them. These procedures are performed in real time, in a distributed manner.

8.3. Optical Medium Failure

The basic fault–tolerant features of the net are derived from its centralized, passive, optical star topology, which can tolerate multiple link failure.

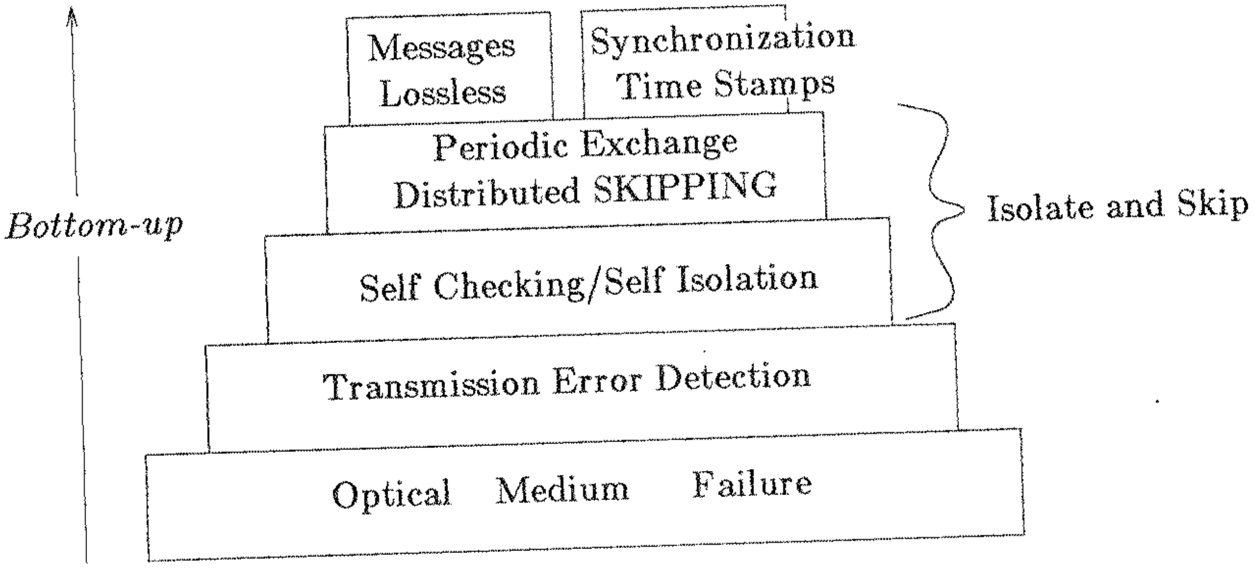

Figure 8.1: Fault Tolerance Methodology

Fault model of the medium – the cause of failure of the optical medium can be as a result of a mechanical damage, which results in a disconnection of the medium or star coupler. The optical star can be protected by a metal box in order to prevent physical damage. The optical net cannot be completely disconnected by a single coupler failure. If the star is constructed, as shown in Figure 2.5, by a $\frac{n}{2}\log_2 n$ couplers, then at least $\frac{n}{2}$ couplers should fail in order that the whole passive star will fail, or that the net will become completely disconnected. Any partial failure where the net is still partially connected, can be overcome by the **isolate and skip** mechanism. Similarly, multiple fiber links failures are overcome by the **isolate and skip** procedure (unlike in rings or linear buses where a single broken link can bring down the whole net).

Furthermore, the optical medium is immune to almost all electro–magnetic, magnetic, and electric interferences (EMI, RFI, etc.), which sharply reduce the probability of intermittent failures, i.e., bit failures in the optical channel. The electronic circuits, not the optical medium, is the main source of failure.

8.4. Transmission and Error Detection

The next step in this FT design is the detection of errors in the serial bit stream (methods for error corrections are not discussed in this chapter). The detection is done by the digital part of the receiver interface. Thus, detected failures can occur in the optical medium, in the analog circuitry of the transmitter, or in the receiver.

The serial bit stream is encoded by the conservative code, which is self–clocking; under normal conditions, the serial–to–parallel conversion is free of the metastable problem, which reduces the reliability of the digital receiver.

Some fault models in the serial bit stream and their corresponding FT solutions are

(1) A faulty detection of the codeword's delimiting transition – this might happen as result of a missing transitions (rising and falling edges); as a result, the sampling of the current codeword is extended into the next codeword, so that all the following codeword samplings are not done on the exact codeword boundaries.
The solution is to map all the unused codewords to an error signal; e.g., the 8B/12B conservative code has 462 possible codewords; since only 256 are needed, the remaining 206 codewords can be used for error detection. Thus, after a few erroneous samplings of codewords, a codeword which is mapped to the error signal would occur. If the number of codewords which are used for legal data word is p, and the total different possible codewords is s, then with probability $\left(\frac{p}{s}\right)^n$ after sampling n successive codewords, the error signal would be asserted.

(2) A faulty transition position – the number of transitions is still preserved, but one of the codeword edges has been shifted.
The previous solution will work only with a probability of $\frac{p}{s}$. Thus, for this failure of the conservative code, the CRC error detection code can be used. The CRC can be computed for each packet before the conservative transformation, and its reminder is added to the tail of the packet. At the receiving side, after the conservative decoding, the CRC is computed and checked.

(3) Collisions – the primary reason for a collision is a failure in the distributed synchronization procedure at one of the net's nodes. Since the transmission from all

the nodes merges to one point in space, all the nodes are capable of detecting collisions of any length.

Collision is detected by using the conservative/balanced code, as described in Section 3.3. Each node can determine if its own transmission was involved in the collision if it knows the collision time. This information can then be used in the self–isolation procedure and in the synchronization, fault tolerant procedure.

8.5. The Interface and Self Isolation

In the network interface, described in Chapter 4, each port is full duplex. For FT purposes it is assumed that the operation of the receiver and transmitter sides are completely independent. The interface is designed hierarchically, and only very small parts of its digital components change their state in a high bandwidth. The full duplex, hierarchical interface and the synchronous decoding improve the reliability of the interface.

From the net point of view, the objective of the FT procedure of the port is to stop the transmission in case of a failure; i.e., self isolation, in which the port does not use either its CMS or any previously reserved DMS. This strategy has major characteristics: (i) it enables the net to maintain or regain synchronization among the nonfaulty nodes, and (ii) it prevents the propagation of error in the system. The isolated port can be passive (not necessarily dead), and continue to monitor the activity over the net.

The following are several fault models which result in self isolation. The means (hardware or software) for detecting these failures is explicitly described. In the context of this discussion there is no attempt to cover all possible failure modes. It is assumed that the conservative/balanced code is used, which makes it possible to determine if a transmission failure is because of a collision or other causes.

(1) Transmission failure – the receiver side of the port receives its own transmission after it was propagated via the optical star. If an error (not a collision) is detected, then the port can retransmit the packet. Since a packet propagates through the net within f time slots, then after $f+1$ slots the port can determine if the transmission was successful and only then discard the transmitted buffer. If the node failed for two successive transmissions, then the node concludes that something is faulty and is isolated from the net.

(2) Transmitter clock failure – this clock is used for shifting the serial data out, and for determining when each slot begins. A small deviation of this clock frequency can be catastrophic for the net operation, since it can cause collisions.
A redundant clock, functioning as a watch–dog timer, detects clock failures. Any disagreement between these two clocks causes the self–isolation of the port.

(3) Collision – a port which detects a collision diagnose to check if its own transmission was involved. It is done by measuring the time difference between the collision detection and the port's last transmission. If the port's transmission is involved in the collision, the port immediately isolates itself.

A self–isolated port performs self–tests in order to further diagnose the source of the failure. If nothing has been found, the port will become active again by simply starting to send control messages during its predetermined CMS.

8.6. Skipping over Self–Isolated Ports

The net continues its operation in the presence of self–isolated ports by skipping or bipassing them during the CMSs and DMSs. This graceful degradation mechanism enables the active ports to continue normal operation. The use of the CMSs is deterministic, and the order of use is predetermined and commonly known. Hence, if a port is idle during its CMS, its successor will continue in its turn. If a port is idle in using its CMS twice successively, then (i) all the DMSs which are reserved by the port are deleted, and (ii) no data packet is sent to this port.

Since there is a one–time reference on each net, the idle information about the self–isolated ports is incorporated into the state transitions of all the nodes of a single net at the same time (a well–defined state transition), which greatly simplifies the handling of idle nodes. Whenever a port rejoins the activity on the net, by using its control minislot, the other ports stop their skipping.

8.6.1. Hardware requirements for skipping

The hardware requirement for skipping is measured by the number of registers, which contain the addresses of the nodes transmitted before this node. Each node has its own set of registers.

In order to handle any number of failures in the use of the control minislot, each node should have a complete list of the nodes in its subset. In the uniform case of n nodes on a net, which use r CMSs, then each node should have a list of

$$l = \left\lceil \frac{n}{r} \right\rceil \; registers.$$

The use of the DMS is dynamic and depends on the specific scheduling algorithm. If a uniform round–robin algorithm is used, then several cases can be identified:

Case 1 – no skipping capability, each node which uses the DMS should store the addresses of $f+1$ nodes which transmit in the sequence before it.

Case 2 – in order to overcome a single failure by skipping, the node should store the address of one more node in the sequence which transmits before it, or total of $f+2$ addresses.

Case 3 – in order to overcome g failures by skipping, the node should store the address of g more nodes in the sequence which transmits before it, or total of $f+1+g$ addresses.

Case 4 – in order to overcome any number of failures by skipping, the node should be able to store the address of all the $n-1$ nodes of the net.

8.7. Lossless System

One of the major objectives of making this system fault tolerant is to minimize the probability of a packet loss. It is achieved by minimizing two events:

(i) sending a packet to a faulty port, achieved by using the complementary acknowledgement mechanism, and

(ii) sending a packet to a port with no empty buffers, achieved by using overflow hazard flags in the control messages.

8.7.1. Complementary acknowledgement

The use of the control minislots by the net's ports is deterministic and periodic, and can be viewed as "*I am alive*" messages; i.e., a port may check itself continuously, as described before, and as long as everything is normal, this port will continue to use its CMS. Once a failure is detected, the port stops using its CMS, and thereby indirectly notifies all the other ports of its failure.

This technique can be viewed as a **complementary acknowledgement.** There are two advantages of this technique; first, communication validity is close to what can be achieved with the usual acknowledgement protocol, and second, the significant acknowledgement overhead is saved.

8.7.2. Overflow prevention

Each node has a finite set of m buffers; one of the objectives of the buffer controller is to prevent overflow and to ensure maximum utilization of the communication capacity. Section 7.5 shows how a switch node of the 2D–P hypergraph prevents overflow and ensures maximum flow. All the nodes of 2D–R can be viewed as switch nodes, and the conditions and algorithms for achieving these objectives are the same. Thus, if the total number of buffers is m, the queue of each port can not have more than half of them. Each node can transmit a control message every l' slots, and f is the number of slots in each frame, then the necessary condition for preventing overflow and ensuring maximum flow

$$\left\lfloor \frac{m}{2} \right\rfloor > 2(l' + 2f + 1).$$

8.7.3. Verifying successful transmission

Two more conditions are needed to be added in order to verify with a high probability that the transmission of a packet is successful:

(i) A port performs a self check on every control message it sends during the CMS; i.e., the port waits $f+1$ time slots and compares the receiving control message with the original one. If the comparison fails, then the port will try to retransmit once more. If it fails again, then the port will isolate itself and the packet will be sent to the host with an unsuccessful transfer message.

(ii) A node does not discard a message for $\left\lceil \frac{n}{r} \right\rceil + 2f+1$ time slots. Within this period of time, the destination node should transmit via its CMS. If the node fails to do so, then the source node should assume that the node is faulty and that the message failed to reach its destination. In the case of failure, the port interface will notify its host.

The delay period for these two conditions is taken in parallel, and since the time for the second condition is always longer, it is necessary to wait only that period of time.

8.8. Synchronization Fault Tolerance

The global event synchronization is a fundamental operation principle of this system. A failure of the synchronization mechanism on one of the net's ports can cause the failure of the whole net. It important to ensure that a net failure will not cause failure of other nets.

8.8.1. Synchronization, failure detection, and protection

The optical net is a shared medium; thus, a synchronization failure is very likely to be manifested by a collision. A conservative/balanced code detects a collision. In order that a node will determine if its own transmission was involved, the node measures the time between its transmission and the collision detection. It is done at the receiving side of the port, by an independent clock. If the collision was detected, within a delay of f slots to $f+1$ slots, since the port's last transmission, the port will then assume that its transmission was involved in the collision. As a protection measure, the ports which were involved in a collision will isolate themselves.

8.8.2. Slot counter failure

The objective of the global synchronization procedure is to uniquely time stamp uniquely all computation and communication events. Thus, a slot counter failure will cause a false time stamp, which can result in a faulty computation.

This type of failure can be masked by comparing the value of the slot counter with the time stamps of the control and data messages, which are continuously received by the node. In the case of disagreement, the node performs majority voting and updates the value slot counter. If this failure repeats, the slot counter should be declared faulty, and the two ports of the node will isolate themselves.

8.8.3. Global synchronization error detection

Error detection in the global synchronization algorithm is achieved by measuring the timing difference between the two ports; if it exceeds $\frac{1}{2}t_s$ an error would be indicated. Hence, if a port causes an arbitrarily long delay (as a result of malfunction), it can be detected by all the other ports.

Note, that this type of error is not likely to happen, since an error should be detected on one of the system's nets (usually a collision).

8.9. Discussion

In the previous fault–tolerance discussion, the main redundancy technique used was hardware. Other FT techniques such as (i) communication redundancy – extra redundant messages, or (ii) time redundancy – repeated transactions, is used to a very limited extent. Thus, fault tolerance enhancements minimally increase the system's operational complexity.

The complementary acknowledgement method ("*I am Alive*"), and the ability to execute a distributed transaction in an *open mode* fashion (see Section 6.4) are important for FT and for reducing the complexity of the distributed computation in the system.

The discussion on fault tolerance in this chapter added another dimension to the significant capabilities of the proposed optical hypergraph architecture. It is very important to note that both the synchronization and fault tolerance mechanisms are derived from the centralized, passive, optical star topology, and the periodic exchange of state information. Thus, the topology and the periodic exchange are important building blocks of large distributed systems.