TeleSoft International, Inc.
4029 S. Capital of Texas Hwy, Ste 220
Austin, TX 78704-7920

Tel: 512.373.4224
Fax: 512.373.4181
sales@telesoft-intl.com

TsLink3 High Availability

TeleSoft enables each of their customers to select the particular redundancy strategy that best suits their system architecture and implementation. TsLink3 code includes generic High Availability (HA) hooks to support the customer's choice of 1:1, N:1, and N:M.

Providing hooks for several types of HA implementations enables each customer to select and optimize the means of moving data from their primary board to another board.

Basic Theory of Redundancy:

TsLink3 protocol stacks are state machines composed of "state information" and "combinatorial logic" (See Figure 1: Abstract State Machine). The State Information is kept in structures and TeleSoft gives the user control over the structures to support High Availability.

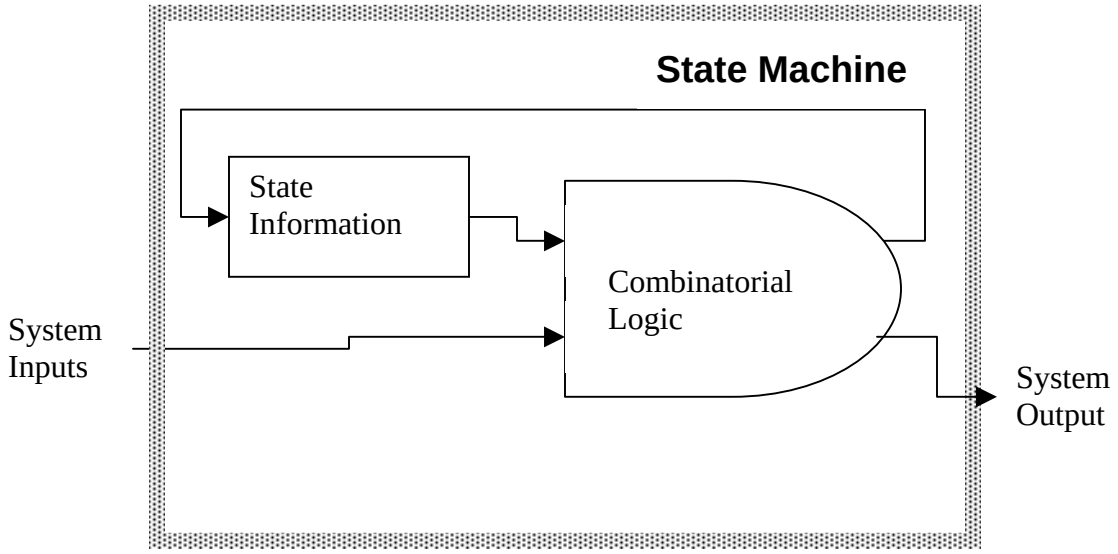

Figure 1: Abstract State Machine

The inputs to the TsLink3 software stack are messages from the lower layers and messages from the user application (layer 4 and up).

Types of High Availability:

There are many types of redundancy, including 1:1, N:1, and N:M.

1:1 Redundancy requires one backup board for every active (primary) board. The active board sends state information to the backup, the backup takes the state information, populates its own structures, and waits. When failover occurs, the backup starts running with minimal startup overhead.

- Advantages: The system is ready to restart immediately.
- Disadvantages: Requires one backup for each active board, so cost, rack space, and power consumption (and heat generated and battery backup) are all very high.

N:1 Redundancy requires one backup board for every "N" active boards. The active boards all send state information to the backup, the backup takes the state information, populates its own structures, and waits. When failover occurs, the backup must select the proper subset of structures to use, and may have to copy that data to a "working" set of structures. An alternate approach is to use some other board to hold all of the state information for the active boards, and populate the backup unit on failover.

- Advantages: lower cost, lower power consumption, lower rack space.
- Disadvantages: one backup unit may not be enough, slower recovery due to having to select from multiple backup sets, backup boards may need a large amount of memory.

N:M Redundancy requires "M" backup boards for every "N" active boards. The active boards all send state information to all (or some subset) of the backups, the backups take the state information, populate their own structures, and wait. When failover occurs, the backup must select the proper subset of structures to use, and may have to copy that data to a "working" set of structures. An alternate approach is to use some other board to hold all of the state information for the active boards, and populate the backup unit on failover.

- Advantages: lower cost, lower power consumption, lower rack space.
- Disadvantages: slower recovery due to having to select from multiple backup sets, backup boards need large amounts of memory. This approach may also require more system bus bandwidth, as each active board may have to send state information to several backup boards.

Memory Backup requires that the memory of each processor running the TsLink3 code be backed up in hardware. This is generally done by having an external memory board that is part of a backup set. Writes to one memory location are stored redundantly on two or more separate memory boards. If one board fails, the other takes over transparently. Note that the processor board must also be backed up.

- Advantages: no special programming required for HA.
- Disadvantages: cost, hardware complexity, rack space, power, heat.

When to backup state information

There is no one good answer to determine when state information should be backed up.

Systems that back up state information on every message will not lose any calls in progress and may not drop dialed digits. However, this frequency of backup leads to high system bus usage and a higher processor load.

Backing up state information for only two states ("active" and "not active") is an industry-accepted minimum and generally fulfills legal and billing requirements ("no active calls shall be dropped"). However, backing up state information for only two states will result in calls-in-progress being dropped which may become an issue with some of the supplementary services, such as call hold.

It is possible to find a middle ground, where "significant" events cause the state information to be backed up (Checkpointing). However, each customer has a different definition of what qualifies as "significant events."

TeleSoft believes that the customer is always right, so we enable you to decide when to backup the state information for each call.

When to restore state information

System architecture determines when to restore state information. In this document, "backup" is the term used to describe copying information from the active board into memory. "Restore" refers to populating the backup board with the state information.

A 1:1 redundant system will backup data from the active board and immediately restore it to the backup board. At failover, the backup will be loaded with all state information and may start processing messages almost immediately.

An M:N redundant system may backup data from the active board to a master system controller board. At failover, the data would be "restored" to the working memory of the backup board. Adding the master system controller into the equation will result in a longer recovery time.

Alternately, in the M:N redundant system, the backup boards may each have multiple "images", one for each primary board that it handles. The backup boards would be "restored" as in the 1:1 example above, with each active board backing up state information and the backup boards immediately restoring the data to the correct image. At failover, system pointers are set to the image of the failed board. While this gives a quicker recovery time, it increases bus bandwidth and processor utilization, and requires more memory on the backup boards.

Clearly, there is no one best answer. Each customer has different needs so TeleSoft enables you to implement the strategy that will work best for your system.

Failure modes

This paper addresses only board failures. Failures of the copper wire or fiber optic line are beyond the scope of this document.

Failure modes include an entire board failure and a partial board failure.

A failure of the entire board may be caused by a power failure, a bad processor, or some other critical component failing (or by a technician pulling the wrong board.) In this case, some other entity must determine and report the failure of the board.

A board may also fail when a component fails that does not affect the CPU, memory, or communication channels to other boards. An example of this would be the failure of an HDLC controller or line interface unit. In this case, the board can report its failure, and the system controller can help it to perform a more graceful shutdown.

Elements of High Availability

Systems supporting High Availability have similar functional requirements, and need to perform many of the same primitive operations:

1. Cause the backup unit to create a call when a call is created on the active board
2. Read state information from the active board for a specified call
3. Write (backup) state information to the backup board for a specified call
4. Terminate the "call" on the backup unit when the call is terminated on the active board.
5. Periodically audit to insure backup is an accurate copy of active board
    a. This requires the ability to read the state information from the backup board, but this may be the same function as reading state information from the active board.
6. Perform the failover operation so that the backup board becomes active
7. Start appropriate call timers when the system failover occurs.
8. Monitor the active and backup boards to insure that they are operating properly
9. Notify operators that a failure has occurred.

TsLink3 software directly supports the first seven items, as detailed in the next section. Because system architectures differ, the user must implement a method to transfer the state information from active to (optional) intermediate storage area to backup board.

Monitoring the boards is the user's responsibility. Polling primitives may be created to ensure processor activity. Failure of the interface chips will cause protocol failures which will result in consistent N_DISC_IN on call requests made on specific physical interfaces.

Notification of failure is system dependent and is the user's responsibility.

TeleSoft's support of High Availability

TeleSoft provides hooks that allow the customer to support all of the above listed forms of High Availability.

TeleSoft supplies an implementation that runs on a single processor. In order to implement HA, the user must write a few routines in order to transfer the data from one board to another.

The user must also determine when the active board has failed, and cause the system to failover to the backup board.

TsLink3 provides five basic primitives to support HA. These are

N_PRI_CONTEXT_IN	asynchronous primitive notifying application, or control layer, when the call state changes.
N_PRI_CONTEXT_RQ	primitive requesting call context information
N_PRI_CONTEXT_RS	primitive containing call context information in response to request; contains same information as N_PRI_CONTEXT_IN
N_PRI_CONTEXT_SET_RQ	Sets up call context information
N_PRI_CONTEXT_ACT_RQ	Activates call from stored information: ▪ Allocates "connid" (handle) ▪ Allocates CRV ▪ Starts timers ▪ Assigns TEIs and starts L2/L1 establishment ▪ (Hardware and general stack initialization must be done by user prior to use of SET or ACT requests.)

Although these primitives do carry adequate information to backup and activate basic PRI calls, all TsLink3 primitives are designed to be customized for specific application needs.

From these five basic primitives, the first seven elements of High Availability may be supported as follows (also see Table 1: Elements of High Availability related to primitives, below):

1. Use N_PRI_CONTEXT_RQ/RS and N_PRI_CONTEXT_SET_RQ to receive, and copy context information.
2. Use N_PRI_CONTEXT_RQ/RS to get information.
3. Use N_PRI_CONTEXT_SET_RQ to set information.
4. Use N_PRI_CONTEXT_SET_RQ to set information.
5. Use N_PRI_CONTEXT_IN to store information or periodically poll using N_PRI_CONTEXT_RQ/RS. We support both to allow for specific needs.
6. Use N_PRI_CONTEXT_ACT_RQ to start up calls on backup board.
7. Created (using default timer values) at use of N_PRI_CONTEXT_ACT_RQ.

The following table relates the primitives to the elements of High Availability:

Table 1: Elements of High Availability related to primitives

1	Cause the backup unit to create a call when a call is created on the active board	Use N_PRI_CONTEXT_RQ/RS and N_PRI_CONTEXT_SET_RQ to receive, and copy context information
2	Read state information from the active board for a specified call	Use N_PRI_CONTEXT_RQ/RS to get information
3	Write (backup) state information to the backup board for a specified call	Use N_PRI_CONTEXT_SET_RQ to set information
4	Terminate the “call” on the backup unit when the call is terminated on the active board.	Use N_PRI_CONTEXT_SET_RQ to set information
5	Periodically audit to insure backup is an accurate copy of active board	Use N_PRI_CONTEXT_IN to store information or periodically poll using N_PRI_CONTEXT_RQ/RS. We support both to allow for specific needs.
6	Perform the failover operation so that the backup board becomes active	Use N_PRI_CONTEXT_ACT_RQ to start up calls on backup board.
7	Start appropriate call timers when the system failover occurs.	Created (using default timer values) at use of N_PRI_CONTEXT_ACT_RQ.
8	Monitor the active and backup boards to insure that they are operating properly	Depends on system architecture
9	Notify operators that a failure has occurred.	Dependent on customer’s marketing requirements