

## TCP's Reaction to Soft Errors

### *Discussion list*

There are a number of possible solutions to the problem. However, before evaluating the solutions themselves, there are a number of aspects that should be evaluated, which relate to the criteria we should apply when deciding what the best way to solve the problem is:

- **Who** should decide which is the best solution?
  - Protocol developers?
  - System Administrators?
  - Application Programmers?
  
- **When** do we want the fix to begin to be used?
  - Now?
  - In many years (if ever)?
  
- What do **robust** applications do?
  - **Interactive applications**  
Is there any **real** success in connection-establishment after several minutes?
  
  - **Non-interactive applications**  
Don't they have to implement retry mechanisms, anyway?

## Possible Solutions

### 1. Changing TCP's Reaction to Soft Errors

There are a number of possibilities to implement this solution.

#### 1.1 Should it be enabled by default?

Whether the fix should be enabled by default or not, depends on when we want the fix to begin to be used. Clearly, if we want the fix to begin to be used now, it should be enabled by default. If the fix were to be enabled by system administrators or application programmers, it might take years (if ever) before it begins to be used.

#### 1.2 Should a System Administrator be allowed to opt?

Enabling Systems Administrators to opt could be useful in those scenarios where the fix could have undesirable effects. In such scenarios, a knowledgeable System Administrator could explicitly disable the fix.

### 1.3 Should an Application Programmer be allowed to opt?

Whether or not application programmers should be enabled to opt probably has to do with whether we can expect an application programmer to be knowledgeable about TCP's specifics. Clearly, an application programmer, such as an SQL programmer, should not be expected to know about TCP fault recovery mechanisms.

It *could* be useful for a low-level API (such as the Sockets API) to provide this functionality. However, for the reasons stated above, in the event this functionality were provided, the fix should be enabled by default..

### 1.4 Possible drawbacks

The possible drawback of this solution is that in some scenarios some connection attempts, that otherwise would have succeeded, could be aborted because of the fix. However, two issues have to be considered here:

#### 1.4.1 How likely are these scenarios?

The only scenarios where there would be deterministic connection aborts due to the proposed fix would be those in which upstream network connectivity is triggered by network use, and thus ICMP Destination Unreachable messages would be sent to the sending host while the upstream network connectivity is being established. In those scenarios, the connection request triggering upstream network connectivity would be aborted. For this scenarios, the system administrator could override the default behaviour on a system-wide basis.

#### 1.4.2 How real is the drawback?

As discussed above, even if a connection attempt were aborted due to the proposed fix, we should consider whether this would **really** be harmful.

For interactive applications, a connection-attempt that succeeds after several minutes is unacceptable. Users would be much happier to get notified about the network problems, and would themselves trigger another connection attempt, some time later.

For non-interactive applications, application-layer retry mechanisms must be implemented, anyway. Robust applications must be prepared to handle scenarios where network connectivity is absent even for several hours. Consider a Mail Transfer Agent (MTA), for example.

## 2. Asynchronous Application Notification

Whether or not this is a good alternative depends on the criteria discussed at the beginning of this document. Clearly, if we want the fix to begin to be used now, this alternative should not be considered feasible. It could take years (if

ever) before the fix begins to be used. Also, if we think Application Programmers should not be bothered with protocol specifics, this fix would not only require them to be knowledgeable about TCP/IP's internals, but would also require additional application complexity.

### 3. A Higher-Level API

A Higher-Level API would be useful for isolating Application Programmers from protocol specifics. One could even argue that this document wouldn't have existed if application programmers had been using a Higher-Level API.

While this is an interesting long-term solution, it is inappropriate if we want to solve this problem now.