

Introduction

The following is a collection of ideas that implement, or extend existing limited TCP implementations. The idea is to provide simple low complexity version of TCP that can be implemented with a minimum to medium set of resources.

SYN-Queueing

When a server has a limited number of connections allowed at one time, you want to queue up the SYN requests on that socket. In iReady's current technology you may have multiple socket connections allowed for each service, but you still want to queue the SYNs that belong to each service (port).

A beneficial implementation for the iReady technology is to implement SYN queueing for only one service or port. If multiple sockets are listening on different services only one can take advantage of SYN queueing. For an embedded device this should be sufficient.

Technology has the following attributes :

- Complements iReady's current technology.

- Extends a limited number of "socket resources" when device is used as a server.

Stateless TCP

Filesystems by port, no knowledge of sockets. Useful in a server environment, less in a client environment unless you can embed the client state in the program code. Can be considered a 1 way Socket connection. Can be coupled with "real" socket to provide post mechanics. Otherwise data can be blindly accepted and processed.

Works like so:

Incoming Packet examples with response :

Syn -> Respond with ACK(seq+1) + SYN sequence # FFFFFFFF

Ack -> Respond with block at location (port-baseport) * block size + (Seq mod blocksize) + Always ACK all their data.

FIN -> Respond with FIN/ACK

Technology has following attributes :

- Fixed Block sizes for all TCP segments.

- File-system based on port numbers.

- No Concept of sockets.

- Client generated packet causes response.

- Completely stateless.

Limitations :

- no dynamic serving unless block based.

Limited State TCP (server only)

Limited knowledge of sockets, index into socket table is sequence number. Here we go one step further than Stateless TCP and we keep the concept of a service belongs to one port. Because of this we need to keep some state information for each socket.

The key to Limited State TCP is during the reception of a SYN the server send a sequence number that's MSByte contains an index value and the lower bits flaged to zero. A simple 2 byte lookup table can keep track of the conection and the filesystem index.

Technology attributes:

Limitations: No dynamic serving unless block based.

Mini-State TCP

Additions to Limited state TCP. Sequence number prediction, dynamic streams, srp.

Limitations

Client Side Stateless TCP

SRP type TCP

Limited transmit buffers, multiple sockets share transmit buffers and must SRP each buffer.

On receive buffers are passed up asap.

State information contains limited sequence info.

Limited socket state information stored.

Other notes:

Server is a filter that passes up control to software.