Dynamic Connection Check

Use Case:

Most of the time an Application owner knows the capacity of Application and how many connections and requests it can handle at one point in time. This use case demonstrates the ability to check for concurrent connections in run time and enforce policies where you can respond back from ADC while itself and not let the connection pass on to backend.

F5 iRules:

```
when RULE_INIT {

   # Set a global max for number of concurrent TCP connections
   set ::max_connections 2

   # Set an HTML response to sent to clients who make a request while the VIP is over the max connection count
   set ::html_content "<html>over limit</html>"

   # Print debug messages to /var/log/ltm?  1=yes, 0=no
   set ::debug 1

   # Initialize a counter for active connections (don't modify this)
   set ::active_connections 0
}
when HTTP_REQUEST {

   # If we're over the limit for this connection, send a response
   if {$::active_connections > $::max_connections}{

      # Send a response
      HTTP::respond 200 content $::html_content
```

```
      # Close the connection
      TCP::close
       # Log a message to /var/log/ltm if debug is enabled
      if {$::debug}{log local0. "Over limit (current/max: $::active_connections/$
::max_connections). Sent response to [IP::client_addr]"}


   # We're not over the limit, so check if this is the first HTTP request on the
TCP connection.
   } elseif {[HTTP::request_num] == 1}{
      set validrequest 1
      # Increment the TCP connection count.
      incr ::active_connections 1
   }
}

when CLIENT_CLOSED {
   # A connection was closed, so decrement the global counter
   if {$validrequest == 1}{
       incr ::active_connections -1
   }
}
```

NetScaler Solution:

```
add responder action max_connect respondwith '"HTTP/1.1 502 Over
Limit\r\nContent-Length:25\r\nConnection: close\r\n\r\nHttp/1.1 Over Limit"'
add responder policy max_pol "SYS.VSERVER("vip1").CONNECTIONS.GE(2)" max_connect
bind responder global max_pol 1 -type req_deFAULT
```

Here NetScaler is able to perform this task with a simple policy expression and one can keep changing the input values based on the use case.