This article describes how to troubleshoot overload conditions.

Guide Contents
Main Article
Understanding the WAAS Architecture and Traffic Flow
Preliminary WAAS Troubleshooting
Troubleshooting Optimization
Troubleshooting Application Acceleration
Troubleshooting the CIFS AO
Troubleshooting the HTTP AO
Troubleshooting the EPM AO
Troubleshooting the MAPI AO
Troubleshooting the NFS AO
Troubleshooting the SSL AO
Troubleshooting the Video AO
Troubleshooting the Generic AO
Troubleshooting Overload Conditions
Troubleshooting WCCP
Troubleshooting AppNav
Troubleshooting Disk and Hardware Problems
Troubleshooting Serial Inline Clusters
Troubleshooting vWAAS
Troubleshooting WAAS Express
Troubleshooting NAM Integration

Contents

Overview

The Cisco WAAS network would have been designed to optimize a certain number of TCP connections, based on customer requirements. Depending on the model of the WAE, there could be additional connection limitations for the SSL and CIFS application accelerators. When either the overall connection limit or a specific application accelerator connection limit is exceeded, the device is overloaded. In this situation, more

traffic is entering the device than it can handle and so traffic may not be optimized as expected (overloaded traffic is passed through unoptimized).

How to Monitor TFO Flows and Overload Conditions

When a WAAS accelerator device is overloaded, you typically see the following Central Manager alarm: Entering overload state due to Max connections (*nnn*). The number *nnn* is the number of times the WAE has become overloaded since the last reboot.

The device also logs a syslog error message similar to the following:
Sysmon: %WAAS-SYSMON-3-445015: Fault detected: The TFO accelerator is overloaded (connection limit)

You can use various **show** commands at the CLI to determine the number of allowed and actual connections and gather more information.

Checking the TCP Connection Limit

The first useful command is **show tfo detail**, which can tell you how many optimized TFO connections that the device can handle, as follows:

```
wae-7341# show tfo detail

  Policy Engine Config Item            Value
  -------------------------            -----
  State                                Registered
  Default Action                       Use Policy
  Connection Limit                     12000           <-----Maximum number of TFO optimized conn
  Effective Limit                      11988
  Keepalive timeout                    3.0 seconds
```

The Connection Limit value tells you that this WAAS device can support 12000 TFO optimized connections.

The Effective Limit may be lower than the Connection Limit if the MAPI AO has reserved some connections. The reserved connections are subtracted from the Connection Limit to get the Effective Limit.

Checking the Optimized TCP Connections

To understand the TCP flows on the device, you can use the **show statistics connection** command (in version 4.1.1, use the **show statistics connection all** command). This command displays the currently handled TFO/DRE/LZ flows, pass-through flows, and flows that are being handled by a specific application accelerator. An example of this command follows:

```
wae# show statistics connection

Current Active Optimized Flows:                    5
   Current Active Optimized TCP Plus Flows:        5
   Current Active Optimized TCP Only Flows:        0
   Current Active Optimized TCP Preposition Flows: 0
Current Active Auto-Discovery Flows:               0
Current Reserved Flows:                            12          <---------- Added in 4.1.5
Current Active Pass-Through Flows:                 0
Historical Flows:                                  143


D:DRE,L:LZ,T:TCP Optimization,
```

```
A:AOIM,C:CIFS,E:EPM,G:GENERIC,H:HTTP,M:MAPI,N:NFS,S:SSL,V:VIDEO

ConnID  Source IP:Port        Dest IP:Port          PeerID             Accel

92917   10.86.232.131:41197   70.70.7.11:3268       00:1a:64:69:19:fc  TDL
92918   10.86.232.131:41198   70.70.7.11:3268       00:1a:64:69:19:fc  TDL
92921   10.86.232.131:41216   70.70.7.11:3268       00:1a:64:69:19:fc  TDL
94458   10.86.232.131:45354   70.70.7.11:1026       00:1a:64:69:19:fc  TDL
36883   10.86.232.136:1857    10.86.232.131:1026    00:1a:64:69:19:fc  TDL
```

From the first line in the output (Current Active Optimized Flows), you can see that the device currently has five active optimized flows. From the second counter (Current Active Optimized TCP Plus Flows), you can see that they are all being handled with TFO/DRE/LZ optimization (TFO Plus means that DRE and/or LZ optimization is being used in addition to TFO). The third counter (Current Active Optimized TCP Only Flows) shows flows that are optimized by TFO only.

Another useful counter is the Current Active Auto-discovery Flows, which displays flows that have not been fully set up to become optimized flows or pass-through flows. To become fully set up, the connection must see the SYN, SYN ACK, ACK handshake, which is useful to note when dealing with an overload condition. The Current Active Pass-Through Flows counter shows connections that the device has determined to be pass-through or where the device did not see the SYN, SYN ACK, ACK setup. These flows will not be counted as optimized flows. For pass-through flows, a device should be able to handle up to 10 times the number of optimized flows for which it is rated.

The Current Reserved Flows counter shows the number of connections reserved for the MAPI accelerator. For more details about reserved MAPI connections and their impact on device overload, see the section MAPI Application Accelerator Reserved Connections Impact on Overload.

The sum of the following three counters tells you how close the WAE device is to its connection limit:

- Current Active Optimized Flows
- Current Active Auto-Discovery Flows
- Current Reserved Flows (available only in 4.1.5 and later)

If this sum is equal to or greater than the connection limit, the device is in an overload condition.

Details about the five optimized flows are displayed in the table below the counters.

Another command that you can use to see the number of TFO flows currently on a device is the **show statistics tfo detail** command. Two of the most useful counters in the output are "No. of active connections" and under the Policy Engine Statistics the "Active connections", as follows:

```
wae# show statistics tfo detail

 Total number of connections                      : 22915
 No. of active connections                        : 3                <-----Current optimized c
 No. of pending (to be accepted) connections      : 0
 No. of bypass connections                        : 113
 No. of normal closed conns                       : 19124
 No. of reset connections                         : 3788
    Socket write failure                          : 2520
    Socket read failure                           : 0
    WAN socket close while waiting to write       : 1
    AO socket close while waiting to write        : 86
    WAN socket error close while waiting to read  : 0
    AO socket error close while waiting to read   : 80
    DRE decode failure                            : 0
```

```
   DRE encode failure                                : 0
   Connection init failure                           : 0
   WAN socket unexpected close while waiting to read : 1048
   Exceeded maximum number of supported connections  : 0
   Buffer allocation or manipulation failed          : 0
   Peer received reset from end host                 : 53
   DRE connection state out of sync                  : 0
   Memory allocation failed for buffer heads         : 0
   Unoptimized packet received on optimized side     : 0
Data buffer usages:
   Used size:          0 B,  B-size:        0 B,  B-num: 0
   Cloned size:    54584 B,  B-size:    73472 B,  B-num: 111
Buffer Control:
   Encode size:        0 B,  slow:          0,  stop:          0
   Decode size:        0 B,  slow:          0,  stop:          0
AckQ Control:
   Total:          0, Current:          0
Scheduler:
   Queue Size: IO:          0,  Semi-IO:          0, Non-IO:          0
   Total Jobs: IO:     219110,  Semi-IO:     186629, Non-IO:      49227

 Policy Engine Statistics
 ------------------------
 Session timeouts: 0,  Total timeouts: 0
 Last keepalive received 00.0 Secs ago
 Last registration occurred 8:03:54:38.7 Days:Hours:Mins:Secs ago
 Hits:                    52125, Update Released:                 17945
 Active Connections:          3, Completed Connections:           37257  <-----Active Connection
 Drops:                       0
 Rejected Connection Counts Due To: (Total: 12)
    Not Registered      :         12,  Keepalive Timeout   :          0
    No License          :          0,  Load Level          :          0
    Connection Limit    :          0,  Rate Limit          :          0   <-----Connection Limit
    Minimum TFO         :          0,  Resource Manager    :          0
    Global Config       :          0,  Server-Side         :          0
    DM Deny             :          0,  No DM Accept        :          0

 Auto-Discovery Statistics
 -------------------------
 Total Connections queued for accept:  22907
 Connections queuing failures:         0
 Socket pairs queued for accept:       0
 Socket pairs queuing failures:        0
 AO discovery successful:              0
 AO discovery failure:                 0
```

In some cases, the two counters will differ and the reason is that the ?No. of active connections? displays all the current flows that being optimized by TFO, TFO/DRE, TFO/DRE/LZ, and TFO/DRE/LZ and an application accelerator. The ?Active Connections? under the policy engine statistics includes all the flows in the state above plus the connections that are optimized only by TFO and an application accelerator. This situation means that a TCP flow has come in and matched an application accelerator classifier but the SYN, SYN ACK, ACK handshake has not been completed.

In many TFO overload cases, if the problem is still occurring, you can look at these commands and determine if the number of optimized flows is around the rated number of optimized TCP connections for the hardware. If it is, then you can view the flow details and see what is using up all the flows to determine if this traffic is legitimate and overloading the device or is a virus, security scanner, or something else occurring on the network.

The "Connection Limit" counter under the policy engine statistics reports the number of connections rejected

and passed through because the WAE has exceeded its rated number of optimized TCP connections. If this counter is high, it means the WAE is frequently getting more connections than it can handle.

If the number of optimized connections is not close to the rated number of optimized TCP connections and you are still getting an overload alarm, then you should look at the Current active auto-discovery flows from the **show statistics connection** command or the ?Active Connections? under Policy Engine Statistics from the **show statistics tfo detail** command. In some cases, the number of optimized connections can be very low but the Active Connections under the Policy Engine Statistics are roughly equal to the rated number of optimized flows for the hardware. This situation means that there are many flows that match a classifier but they are not fully established. When a TCP SYN matches a classifier, it will reserve an optimized connection. This connection will not appear in the optimized TCP connections count until the TCP handshake is finished and optimization starts. If the device determines that the flow should not be optimized, it will be removed from the count of active connections under the Policy Engine Statistics.

To further troubleshoot cases where TFO overload is occurring and the Policy Engine Statistics Active Connections seem to be using up all the optimized TCP connections on the device, use the **show statistics accelerator detail** command. In the output of this command, look at the Active Connections under the Policy Engine Statistics for each application accelerator to determine which application accelerator is receiving these connections that are not fully established. Next, look at what state these flows might be in by using the **show statistics filtering** command, which provides you with the number of filtering tuples on the device, as follows:

```
wae# show statistics filtering

Number of filtering tuples:                              18
Number of filtering tuple collisions:                    0
Packets dropped due to filtering tuple collisions:       0
Number of transparent packets locally delivered:         965106
Number of transparent packets dropped:                   0
Packets dropped due to ttl expiry:                       0
Packets dropped due to bad route:                        10
Syn packets dropped with our own id in the options:      0
Syn-Ack packets dropped with our own id in the options:  0
Internal client syn packets dropped:                     0
Syn packets received and dropped on estab. conn:         0
Syn-Ack packets received and dropped on estab. conn:     0
Syn packets dropped due to peer connection alive:        525
Syn-Ack packets dropped due to peer connection alive:    0
Packets recvd on in progress conn. and not handled:      0
Packets dropped due to peer connection alive:            1614
Packets dropped due to invalid TCP flags:                0
Packets dropped by FB packet input notifier:             0
Packets dropped by FB packet output notifier:            0
Number of errors by FB tuple create notifier:            0
Number of errors by FB tuple delete notifier:            0
Dropped WCCP GRE packets due to invalid WCCP service:    0
Dropped WCCP L2 packets due to invalid WCCP service:     0
Number of deleted tuple refresh events:                  0
Number of times valid tuples found on refresh list:      0
```

The number of filtering tuples is the number of flows on the device that are optimized, in pass-through, in FIN WAIT state, in setup state, and so on. Each established flow appears as two tuples, one for each side of the flow, so the number that you see in this output may be much larger than the number of flows that you are seeing in the other commands.

To get more information on the flows in the filtering list, you can use the **show filtering list** command as follows:

```
wae# show filtering list

E: Established, S: Syn, A: Ack, F: Fin, R: Reset
s: sent, r: received, O: Options, P: Passthrough
B: Bypass, L: Last Ack, W: Time Wait, D: Done
T: Timedout, C: Closed

   Local-IP:Port        Remote-IP:Port          Tuple(Mate)        State
  10.86.232.82:23      10.86.232.134:41784 0xbc1ae980(0x0       )   E
 10.86.232.131:58775      70.70.7.11:3268  0x570b2900(0x570b2b80)   EW
    70.70.7.11:3268    10.86.232.131:58775 0x570b2b80(0x570b2900)   EDL
    70.70.7.11:3268    10.86.232.131:57920 0x570b2d80(0x570b2800)   E
 10.86.232.131:57920      70.70.7.11:3268  0x570b2800(0x570b2d80)   E
  10.86.232.82:23      161.44.67.102:4752  0xbc1aee00(0x0       )   E
 10.86.232.131:58787      70.70.7.11:1026  0x570b2080(0x570b2e80)   EW
    70.70.7.11:1026    10.86.232.131:58787 0x570b2e80(0x570b2080)   EDL
 10.86.232.131:48698      70.70.7.11:1026  0x570b2f00(0x570b2880)   PE
 10.86.232.131:58774      70.70.7.11:389   0x570b2300(0x570b2180)   EW
    70.70.7.11:389     10.86.232.131:58774 0x570b2180(0x570b2300)   EDL
 10.86.232.131:58728      70.70.7.11:1026  0x570b2380(0x570b2a00)   E
 10.86.232.131:58784      70.70.7.11:1026  0x570b2e00(0x570b2980)   EW
    70.70.7.11:1026    10.86.232.131:58784 0x570b2980(0x570b2e00)   EDL
    70.70.7.11:1026    10.86.232.131:48698 0x570b2880(0x570b2f00)   PE
 10.86.232.131:58790      70.70.7.11:3268  0x570b2100(0x570b2c80)   EW
    70.70.7.11:3268    10.86.232.131:58790 0x570b2c80(0x570b2100)   EDL
```

If the **show statistics accelerator all** command shows you which application accelerator is using up all the optimized TFO connections, you can filter on that port or traffic. For example, if you want to filter on port 80 traffic, use the **show filtering list | I :80** command.

Look at the legend in the State column. In the case where the flows are in the SYN state, you may see a lot of flows with a state of S. If the WAE has sent back the SYN ACK with options set you may see the state SAsO. This indication may help you determine the state of the flow and from there, you can determine if there is a routing problem, virus, or a problem with the WAE not releasing connections. You may need traces to determine exactly what is happening to the flows but the commands above should give you an idea of what to look for.

MAPI Application Accelerator Reserved Connections Impact on Overload

Often, a TFO overload can be caused by the MAPI application accelerator reserved connections, so it is helpful to understand the process of how the MAPI application accelerator reserves connections.

The MAPI application accelerator reserves TFO connections to ensure that it will have enough connections available to it to accelerate all current and future connections that the clients will make to the Exchange servers. It is normal for a MAPI client to make multiple connections. If a client makes the initial connection through the MAPI application accelerator, but the subsequent connections fail in the MAPI application accelerator, there is a risk that the client's connection might fail.

In order to avoid these potential connection failures, the MAPI application accelerator reserves connection resources as follows:

- Before any client connections begin, it reserves 10 connections for itself, as a buffer for anticipated new connections.
- For each client connection to the server, it reserves three TFO connections for that client-server pair and one of the three is used as an active connection for this first connection. If the same client makes

a second or third connection to the same server, those are handled out of the reserved connection pool. If a client only ever makes a single connection to the server, those two reserved connections will be unused and remain in the reserved pool. If the client makes a connection to a different server, three new connections are again reserved for that client-server pair.

All of these reserved connections are designed to improve performance and to reduce the possibility of a client connection failing because of the inability to make additional connections through the MAPI application accelerator.

Overload occurs when Current Active Optimized Flows + Current Active Auto-Discovery Flows + Current Reserved Flows is greater than the device's fixed Connection Limit. In general, new connections would then be passed through. But some new MAPI connections may still be optimized. When the device is at the overload point, if a client makes an additional request to a MAPI server it already has connected to, then reserved connections are used. But if there are not enough reserved connections (for example, if a client makes a fourth connection to the same MAPI server and the WAE is already in overload) then an escaped connection condition might occur, and this could lead to erroneous behavior such as a client receiving many duplicate copies of the same single mail message.

If the system did not forward the connection to the MAPI application accelerator, you should see "PT Rjct Resources" or "PT in progress", depending on whether there is activity on the connection. If the connection was forwarded to the MAPI application accelerator and then the reservation failed, the connection will be marked with a "G" for the Accelerator, instead of an "M" (in the **show statistics connection optimized mapi** command output). For an example of this command, see the article Troubleshooting the MAPI AO.

If you are experiencing frequent overload conditions, it is important to understand how the Outlook clients are making connections (how many connections to how many Exchange servers). With Outlook running on a client, hold the **Ctrl** key while you right-click on the Outlook icon in the system tray on the task bar. Choose **Connection Status** to display the list of servers to which the Outlook client has connected. From that you can see how many connections the client is making and to how many different Exchange servers. If the client is making connections to several different servers, it would be helpful to investigate ways to consolidate mail so a user only opens MAPI connections to a single Exchange server, and makes use of multiple connections to that server.

It is also useful to investigate whether there are any other applications that might be making MAPI connections.

Solutions for Overload Conditions

Examine optimized connections to see if they are legitimate. In many cases, a Denial of Service (DoS) attack encountered in the network may be causing the WAE to attempt to optimize connections. If so, employ a DoS protection mechanism in the network to proactively close the connections.

In cases where the connections are legitimate, the WAE deployed in the location is undersized and may need to be upgraded, or an additional WAE can be deployed to increase scalability within that site.