This article describes how to troubleshoot Serial Inline Cluster issues.

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

NOTE: Serial inline clustering between non-optimizing peers and interception ACLs were introduced in WAAS version 4.2.1. This section is not applicable to earlier WAAS versions.

Checking for Connectivity Between the Serial Peers

To see which devices are connected to the inline interfaces, use the **show cdp neighbors** command, as follows:

```
WAE#show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater
Device ID        Local Intrfce    Holdtme    Capability  Platform  Port ID
BBSw-R32-R62     Inline 1/1/lan   154             S I    WS-C3750G-Gig 3/0/17
BBSw-R32-R62     Inline 1/0/lan   154             S I    WS-C3750G-Gig 2/0/18
BBSw-R32-R62     Gig 1/0          126             S I    WS-C3750G-Gig 2/0/22
PLT-32-08-7301   Inline 1/1/wan   148              R     7301      Gig 0/2
PLT-32-08-7301   Inline 1/0/wan   147              R     7301      Gig 0/1
WAE-32-08-7341   Inline 1/1/wan   145             T H    OE7341    Inline 1/1/w
WAE-32-08-7341   Inline 1/0/wan   145             T H    OE7341    Inline 1/0/w
```

If the serial peers are separated by one or more switches, the peer will not show up in the output above.

Verifying that the Serial Peers are Configured Correctly

To verify that the serial peers are configured correctly, use the **show peer optimization** command, as follows:

```
WAE#show peer optimization
Configured Non-optimizing Peers:
        Peer Device Id: 00:1a:64:c2:40:8c
```

Run this command on both the peers and make sure that each device shows up on the other correctly.

Use the **show device-id** command to check the device ID, as follows:

```
WAE#show device-id
System Device ID is: 00:21:5e:57:e9:d4
```

Verifying that a Serial Inline Cluster is Operational

Given the following topology example:

BR-WAE ----------WAN---------- DC-WAE2 -- DC-WAE1

or

BR-WAE1 -- BR-WAE2 ----------WAN---------- DC-WAE2 -- DC-WAE1

Normally, optimization should take place between the outermost WAEs, that is, BR-WAE and DC-WAE1, or BR-WAE1 and DC-WAE1. To ensure this, verify the device IDs on the connections by using the **show statistics connection** command. The PeerID on BR-WAE should indicate that it is optimizing with DC-WAE1 and the PeerID on DC-WAE1 should indicate that it is optimizing with BR-WAE.

```
BR-WAE#show statistics connection

Current Active Optimized Flows:                   7552
  Current Active Optimized TCP Plus Flows:        7563
  Current Active Optimized TCP Only Flows:        0
```

```
  Current Active Optimized TCP Preposition Flows:   0
Current Active Auto-Discovery Flows:                12891
Current Reserved Flows:                             100
Current Active Pass-Through Flows:                  3053
Historical Flows:                                   429


D:DRE,L:LZ,T:TCP Optimization RR:Total Reduction Ratio
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 RR
786432   190.190.3.175:19268      155.155.7.208:80 00:21:5e:52:25:5c THDL  00.0%
786435   190.190.5.115:19283      155.155.0.144:80 00:21:5e:52:25:5c THDL  86.0%
786438     199.199.3.0:58436      155.155.9.15:443 00:21:5e:52:25:5c TSDL  00.0%
786440   190.190.2.231:19312      155.155.0.112:80 00:21:5e:52:25:5c THDL  86.0%
```

The PeerID in the above output should match that of DC-WAE1.

All connections on DC-WAE2 should be in the "PT Intermediate" state.

If DC-WAE1 fails or goes into overload, new connections should be optimized between BR-WAE1 and DC-WAE2. You can verify this by using the **show statistics connection optimized** command on DC-WAE2. Optimized connections should be seen on DC-WAE2, with the peer ID of BR-WAE1 as the peer device.

If BR-WAE1 fails or goes into overload, there should *not* be optimization between DC-WAE2 and DC-WAE1. All connections should be in the "PT Non-optimizing Peer" state on DC-WAE1 and "PT No Peer" on DC-WAE2. An example of the expected **show statistics connection** command output follows:

```
DC-WAE1# sh stat conn

Current Active Optimized Flows:                     0
   Current Active Optimized TCP Plus Flows:         0
   Current Active Optimized TCP Only Flows:         0
   Current Active Optimized TCP Preposition Flows:  0
Current Active Auto-Discovery Flows:                0
Current Reserved Flows:                             100
Current Active Pass-Through Flows:                  1
Historical Flows:                                   1


Local IP:Port         Remote IP:Port        Peer ID           ConnType
2.74.2.162:37116      2.74.2.18:80          00:21:5e:27:ae:14 PT Non-optimizing Peer
2.74.2.18:80          2.74.2.162:37116      00:21:5e:27:ae:14 PT Non-optimizing Peer



DC-WAE2# sh stat conn

Current Active Optimized Flows:                     0
   Current Active Optimized TCP Plus Flows:         0
   Current Active Optimized TCP Only Flows:         0
   Current Active Optimized TCP Preposition Flows:  0
Current Active Auto-Discovery Flows:                0
Current Reserved Flows:                             100
Current Active Pass-Through Flows:                  1
Historical Flows:                                   1


Local IP:Port         Remote IP:Port        Peer ID           ConnType
```

```
2.74.2.162:37116      2.74.2.18:80          N/A              PT No Peer
2.74.2.18:80          2.74.2.162:37116      N/A              PT No Peer
```

You can also use the Central Manager Connection Statistics report (***Device* > Monitor > Optimization > Connections Statistics**) to display device connection statistics in a table, as shown in Figure 1. The Peer IDs are indicated by device name.

Figure 1. Central Manager Device Connection Statistics Report

Source IP:Port	Dest IP:Port	Peer Id	Applied Policy / Bypass Reason	Connection Start Time	Open Duration (hh:mm:ss)	Org Bytes	Opt Bytes	% Comp	Classifier
10.34.30.180:3558	128.107.191.124:1703	SJCF-00A-WAAS02		24-Feb-10 20:53	1:13:24	35.3691 KB	15.2803 KB	57%	**Map De
10.34.30.180:3560	128.107.191.124:1703	SJCF-00A-WAAS02		24-Feb-10 20:53	1:13:24	3.7715 KB	3.6211 KB	4%	**Map De
10.34.30.180:3561	128.107.191.124:1703	SJCF-00A-WAAS02		24-Feb-10 20:53	1:13:24	1.0129 MB	179.4541 KB	83%	**Map De

Detecting Serial Peer Configuration Mismatch

Serial peers must be configured so that each is designated as a non-optimizing peer with the other. If device A is configured as a peer of B, but B is not configured as a peer of A, that is a mismatch. To discover a mismatch, you can use the Central Manager **My WAN > Configure > Peer Settings** page, which reports on the status of all serial peers, as shown in Figure 2. All correctly configured serial peers have a green check mark in the Mutual Pair column. Any devices without a green check mark are incorrectly configured with a serial peer that is not also configured with the device as its serial peer.

Figure 2. Central Manager Peer Settings

To detect a serial peer configuration mismatch, you can also look for syslog messages such as the following:

```
%WAAS-SYS-4-900000: AD: Serial Mode configuration mismatch with peer_id=00:21:5e:27:a8:80
```

This error indicates that the serial peer configuration is not symmetrical on both of the peer devices.

Troubleshooting MAPI Acceleration

General MAPI AO troubleshooting is covered in the section "MAPI Accelerator" in the Troubleshooting Application Acceleration article.

The following issues can occur with MAPI acceleration on serial inline clusters:

- Outlook connection to the Exchange server is disconnected and restored
- Outlook connection to the Exchange server is disconnected and stays that way
- Outlook has problems establishing connections with the Exchange server
- Outlook connection to the Exchange server is not optimized by WAAS (either it is in pass-through or no MAPI AO optimization is done)
- MAPI escaped connections because of the EPM policy timeout in the DC WAE

Check EPM and MAPI Dynamic Policies

Use the **show policy-engine application dynamic** command to check the EPM and MAPI dynamic policies, as follows:

```
WAE34#show policy-engine application dynamic
Dynamic Match Freelist Information:
  Allocated: 32768  In Use: 3  Max In Use: 4  Allocations: 14

Dynamic Match Type/Count Information:
  None                    0
  Clean-Up                0
  Host->Host              0
  Host->Local             0
  Local->Host             0
  Local->Any              0
  Any->Host               3
  Any->Local              0
  Any->Any                0

Individual Dynamic Match Information:
  Number:     1   Type: Any->Host (6)  User Id: EPM (3)      <------ EPM Policy
    Src: ANY:ANY  Dst: 10.56.45.68:1067
    Map Name: uuid1544f5e0-613c-11d1-93df-00c04fd7bd09
    Flags: TIME_LMT REPLACE FLOW_CNT
    Seconds: 1200  Remaining: 8  DM Index: 32765
    Hits: 1  Flows: 0  Cookie: 0x00000000
    DM Ref Index: -None-  DM Ref Cnt: 0

  Number:     2   Type: Any->Host (6)  User Id: EPM (3)      <------ EPM Policy
    Src: ANY:ANY  Dst: 10.56.45.68:1025
    Map Name: uuidf5cc5a18-4264-101a-8c59-08002b2f8426
    Flags: TIME_LMT REPLACE FLOW_CNT
    Seconds: 1200  Remaining: 10  DM Index: 32766
    Hits: 1  Flows: 0  Cookie: 0x00000000
    DM Ref Index: -None-  DM Ref Cnt: 0

   Number:     3   Type: Any->Host (6)  User Id: EPM (3)
    Src: ANY:ANY  Dst: 10.56.45.68:1163
    Map Name: uuida4f1db00-ca47-1067-b31f-00dd010662da
    Flags: TIME_LMT REPLACE FLOW_CNT
    Seconds: 1200  Remaining: 509  DM Index: 32767
    Hits: 5  Flows: 0  Cookie: 0x00000000
    DM Ref Index: -None-  DM Ref Cnt: 0

WAE33#show policy-engine application dynamic
Dynamic Match Freelist Information:
 Allocated: 32768  In Use: 2  Max In Use: 5  Allocations: 12

Dynamic Match Type/Count Information:
  None                    0
  Clean-Up                0
```

```
  Host->Host              1
  Host->Local             0
  Local->Host             0
  Local->Any              0
  Any->Host               1
  Any->Local              0
  Any->Any                0

Individual Dynamic Match Information:
 Number:     1   Type: Host->Host (2)  User Id: MAPI (5)       <------ MAPI Policy
   Src: 10.56.45.246:ANY  Dst: 10.56.45.68:1163
   Map Name: uuida4f1db00-ca47-1067-b31f-00dd010662da
   Flags: REPLACE FLOW_CNT RSRVD_POOL REF_SRC_ANY_DM
   Seconds: 0  Remaining: - NA -  DM Index: 32764
   Hits: 12  Flows: 5  Cookie: 0x00000000
   DM Ref Index: 32767  DM Ref Cnt: 0

 Number:     2   Type: Any->Host (6)  User Id: EPM (3)
   Src: ANY:ANY  Dst: 10.56.45.68:1163
   Map Name: uuida4f1db00-ca47-1067-b31f-00dd010662da
   Flags: TIME_LMT REPLACE FLOW_CNT
   Seconds: 1200  Remaining: - NA -  DM Index: 32767
   Hits: 2  Flows: 0  Cookie: 0x00000000
   DM Ref Index: -None-  DM Ref Cnt: 1
```

Check Filtering and Auto-discovery Statistics

Check the output of the following commands to see if the relevant MAPI counters are incremented.

```
WAE#show stat auto-discovery
Auto discovery structure:
      Allocation Failure:                          0
      Allocation Success:                          12886550
      Deallocations:                               12872245
      Timed Out:                                   1065677
.
.
.
Auto discovery Miscellaneous:
      RST received:                                87134
      SYNs found with our device id:               0
      SYN retransmit count resets:                 0
      SYN-ACK sequence number resets (syncookies): 0
      SYN-ACKs found with our device id:           0
      SYN-ACKs found with mirrored options:        0
      Connections taken over for MAPI optimization: 0     <----- MAPI & Serial Inline cluster sta

WAE#show stat filtering
Number of filtering tuples:                          44892
Number of filtering tuple collisions:                402
Packets dropped due to filtering tuple collisions:   3
Number of transparent packets locally delivered:     287133100
Number of transparent packets dropped:               0
Packets dropped due to ttl expiry:                   0
Packets dropped due to bad route:                    589
Syn packets dropped with our own id in the options:  0
In ternal client syn packets dropped:                 0
Syn packets received and dropped on estab. conn:     1
Syn-Ack packets received and dropped on estab. conn: 22016
Syn packets dropped due to peer connection alive:    0
Syn-Ack packets dropped due to peer connection alive: 4
Packets recvd on in progress conn. and not handled:  0
```

```
Packets dropped due to peer connection alive:          1806742
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
SYN packets sent with non-opt option due to MAPI:      0    <----- MAPI & Serial Inline Cluster s
Internal Server conn. not optimized due to Serial Peer: 0
Duplicate packets to synq dropped:                     8
```

Enabling Debug Logging

If looking at the dynamic policies and the filtering and auto-discovery statistics does not help, then enable debug logging so that a technical support engineer can troubleshoot what is happening with MAPI accelerated connections in a serial inline cluster.

Enable debugging by running the following commands:

```
WAE#debug policy-engine connection
WAE#debug auto-discovery connection
WAE#debug filtering connection
WAE#debug connection acl
```

As always, disk logging needs to be enabled and the logging level for disk has to be set to debug.

NOTE: Debug logging is CPU intensive and can generate a large amount of output. Use it judiciously and sparingly in a production environment.

Troubleshooting Interception Access Lists

This section describes how to troubleshoot the following problems related to interception ACLS:

- Connections are not optimized
- Connections are not being bypassed as expected

Connections Are Not Optimized

If connections are not optimized as expected, it could be due to the following causes.

1. The interface may be down. If it is an inline interface, all traffic will be bypassed in hardware. Use the following command to check the interface status:

```
WAE#show interface inlinegroup 1/0
Interface is in intercept operating mode.       <------ Interface must be in intercepting mode
Standard NIC mode is off.
```

2. If the interface is up, check the state of the connections, and if they are in pass-through, check the reason using the following command:

```
WAE#show stat connection pass-through
Current Active Optimized Flows:                    9004
   Current Active Optimized TCP Plus Flows:        9008
```

```
   Current Active Optimized TCP Only Flows:          0
   Current Active Optimized TCP Preposition Flows:   0
Current Active Auto-Discovery Flows:                 10294
Current Reserved Flows:                              100
Current Active Pass-Through Flows:                   2994
Historical Flows:                                    443
Local IP:Port         Remote IP:Port        Peer ID          ConnType
155.155.14.9:21       199.199.1.200:28624   N/A              PT App Cfg
155.155.13.92:21      199.199.1.147:26564   N/A              PT App Cfg   <----- Pass-through rea
```

3. If the reason appears as "PT Interception ACL", then it is due to the interception ACL denying the SYN packets.

You can look at the following output to drill down into the ACL to see which condition matched:

```
WAE#show ip access-list
Space available:
   49 access lists
  499 access list conditions
Standard IP access list test
  1 permit any (1296 matches)
    (implicit deny any: 0 matches)
  total invocations: 1296
Interface access list references:
 None Configured
 Application access list references:
  INTERCEPTION                  Standard        test
    Any IP Protocol
```

Connections Are Not Being Bypassed as Expected

If connections are not being bypassed as expected, make sure that the interception ACL configuration took effect using the following command:

```
WAE#show ip access-list
Space available:
   49 access lists
  499 access list conditions
Standard IP access list test
  1 permit any (1296 matches)
    (implicit deny any: 0 matches)
  total invocations: 1296
Interface access list references:
 None Configured
 Application access list references:
  INTERCEPTION                  Standard        test
    Any IP Protocol
```

Check the hit counts from the above output to see if they are incrementing as expected.

Enabling Debug Logging

If everything appears correct by using the commands above but there is still an issue, enable the following debug logging and look for the policy-engine decision on the SYN packet of interest.

```
WAE#debug policy-engine connection
```

As always, disk logging needs to be enabled and the logging level for disk has to be set to debug.

NOTE: Debug logging is CPU intensive and can generate a large amount of output. Use it judiciously and sparingly in a production environment.