F5 Sites V

SIGN IN

Knowledge

K8082: Overview of TCP connection setup for BIG-IP LTM virtual server types Updated Date: May 31, 2023

→ Al Recommended Content

Published Date: Oct 29, 2015

Applies to:

Topic

The BIG-IP virtual server type specifies the attributes for a virtual server. For example, a Standard virtual server has a different set of attributes and is used to process traffic differently than a Forwarding virtual following is a description of the connection setup characteristics for BIG-IP LTM virtual server types:

Note: This article is not applicable to all virtual servers types, such as non-TCP virtual server types, or those that do not process user traffic.

server. The virtual server type can be found in the Configuration utility by navigating to Local Traffic > Virtual Servers, clicking a specific virtual server, and then viewing the Type drop-down box. The Standard virtual server

Performance Layer4 virtual server

Performance HTTP virtual server

Forwarding IP virtual server

Reject virtual server

The BIG-IP LTM TMOS operating system implements a full proxy architecture for virtual servers configured with a TCP profile. By assigning a custom TCP profile to the virtual server, you can configure the BIG-IP LTM system to maintain compatibility to disparate server operating systems in the data center. At the same time, the BIG-IP LTM system can leverage its TCP/IP stack on the client side of the connection to

In a full proxy architecture, the BIG-IP LTM system appears as a TCP peer to both the client and the server by associating two independent TCP connections with the end-to-end session. Although certain client information, such as the source IP address or source TCP port, may be re-used on the server side of the connection, the BIG-IP LTM system manages the two sessions independently, making itself

The Standard virtual server requires a TCP or User Datagram Protocol (UDP) profile, and may optionally be configured with HTTP, FTP, or Secure Sockets Layer (SSL) profiles if Layer 7 or SSL processing is required. The TCP connection setup behavior for a Standard virtual server varies depending on whether a TCP profile or a TCP and Layer 7 profile, such as HTTP, is associated with the virtual server.

The TCP connection setup behavior for a Standard virtual server operates as follows: the three-way TCP handshake occurs on the client side of the connection before the BIG-IP LTM system initiates the TCP

LTM sends SYN-ACK TCP segment to Client

Client sends SYN request to LTM Standard Virtual Server

A Standard virtual server processes connections using the full proxy architecture. The following TCP flow diagram illustrates the TCP handshake for a Standard virtual server with a TCP profile:

Client sends ACK packet to acknowledge receiving the SYN ACK segment from the LTM seq_num = 1, ack_num = 1 Client-side TCP handshake is complete. LTM chooses Pool member SYN LTM initiates SYN request with new ISN to pool member seq_num = 100 SYN_ACK Pool member sets SYN and ACK bits in the TCP header. Sends TCP segment sent to LTM seq_num = 0, ack_num = 101 ACK LTM acknowledges receiving SYN ACK segment from pool member seq_num = 101, ack_num = : Client begins sending data Data Client sends data seq_num = 1 Figure: TCP handshake for a Standard virtual server with a TCP profile

Pool Member

Standard virtual server with Layer 7 functionality If a Standard virtual server is configured with Layer 7 functionality, such as an HTTP profile, the client must send at least one data packet before the server-side connection can be initiated by the BIG-IP LTM

> Pool Member Client sends SYN request to LTM Standard Virtual Server with Layer 7 functionality

segment from the LTM seq_num = 1, ack_num = 1 Client-side TCP handshake is complete. The LTM waits for data packet from Client before selecting pool member HTTP_GET Client sends HTTP GET request LTM chooses pool member and initiates TCP connection

LTM sends SYN-ACK TCP segment to Client

Client sends ACK packet to acknowledge receiving the SYN ACK

SYN LTM initiates SYN request with new ISN to pool member seq_num = 100 SYN ACK Pool member sets SYN and ACK bits in the TCP header. Sends TCP segment sent to LTM seq_num = 0, ack_num = 101 ACK LTM acknowledges receiving SYN ACK segment from pool member seq num = 101, ack num = 1 LTM sends data packet to pool member HTTP_GET LTM sends GET request to pool member Figure: TCP handshake for a Standard virtual server with Layer 7 functionality Performance Layer4 virtual server The Performance Layer4 virtual server type uses the Fast L4 profile. Depending on the configuration, the virtual server uses the PVA ASIC chip with the PVA Acceleration mode defined as one of the following: full, assisted, or none. Irrespective of the PVA acceleration mode used in the profile, the Performance Layer4 virtual server processes connections on a packet-by-packet basis. Note: Platforms such as the VIPRION and 8900 do not contain a PVA ASIC chip, and all FastL4 connection processing is done in Traffic Management Microkernel (TMM). For more information about whether your platform contains the PVA ASIC chip, refer to the appropriate BIG-IP Platform Guide. For information about how to locate F5 product manuals, refer to Finding product documentation in K98133564: Tips for searching AskF5 and finding product documentation. The Performance Layer4 virtual server packet-by-packet TCP behavior operates as follows: The initial SYN request is sent from the client to the BIG-IP LTM virtual server. The BIG-IP LTM system makes the

The following TCP flow diagram illustrates the TCP handshake for a Performance Layer4 virtual server:

server

Node

 $seq_num = 0$ SYN_ACK Node sends SYN-ACK TCP packet back to LTM seq_num = 0, ack_num = 1 SYN_ACK LTM sends SYN-ACK TCP packet to Client

Client sends ACK packet to acknowledge receipt of the SYN

LTM sends SYN request to node

ACK packet from the LTM

Client sends SYN request to LTM The Forwarding IP virtual

ACK LTM sends ACK packet to Node seq_num = 1, ack_num = 1 Data Client sends Data packet to LTM Data LTM sends Data packet to node Figure: TCP handshake for a Performance Layer4 virtual server Performance HTTP virtual server The Performance HTTP virtual server type uses the Fast HTTP profile. The Performance HTTP virtual server with the Fast HTTP profile is designed to speed up certain types of HTTP connections and reduce the number of connections opened to the back-end HTTP servers. This is accomplished by combining features from the TCP, HTTP, and OneConnect profiles into a single profile that is optimized for network performance. The Performance HTTP virtual server processes connections on a packet-by-packet basis and buffers only enough data to parse packet headers. The Performance HTTP virtual server TCP behavior operates as follows: The BIG-IP system establishes server-side flows by opening TCP connections to the pool members. When a client makes a connection to the Performance HTTP virtual server, if an existing server-side flow to the pool member is idle, the BIG-IP LTM system marks the connection as non-idle and sends a client request over the connection. Performance HTTP virtual server with idle server-side flow The following TCP flow diagram illustrates the client connection to the Performance HTTP virtual server when an idle server-side flow is found (in this case the idle flow was created by the BIG-IP system): BIGIP LTM Client Pool Member

SYN_ACK LTM sends SYN-ACK TCP packet back to Client seq_num = 0, ack_num = 1 ACK Client sends ACK packet to acknowledge receiving the SYN ACK segment from LTM seq_num = 1, ack_num = 1

LTM sends HTTP GET request over idle server-side flow

Pool member sends SYN-ACK TCP segment back to LTM

LTM sends ACK packet to acknowledge receiving the SYN ACK

Client sends SYN packet to Performance HTTP virtual server

LTM sends SYN packet to the pool member

segment from pool member

Client sends HTTP GET request

Figure: Client connection to the Performance HTTP virtual server when an idle server-side flow is found Performance HTTP virtual server with no idle server-side flow If no idle server-side flow is found, the BIG-IP system creates a new server-side TCP connection and sends a client request over the connection. The following TCP flow diagram illustrates the client connection to the Performance HTTP virtual server when no idle server-side flow is found: BIGIP LTM Pool Member Client SYN Client sends SYN request to LTM Performance HTTP virtual server $seq_num = 0$ SYN_ACK LTM sends SYN-ACK TCP segment to Client seq_num = 0, ack_num = 1 ACK Client sends ACK packet to acknowledge receiving the SYN ACK segment from the LTM seq num = 1, ack num = 1 Client-side handshake complete. No idle server-side flows available, so LTM chooses pool member and initiates TCP connection SYN LTM initiates SYN request with new ISN to pool member seq_num = 100 SYN ACK Pool member sets SYN and ACK bits in the TCP header. Sends TCP segment sent to LTM $seq_num = 0, ack_num = 101$ ACK LTM acknowledges receiving SYN ACK segment from pool member seq_num = 101, ack_num = 1

virtual server shares the same IP address as a node in an associated VLAN. Before creating a Forwarding Layer 2 virtual server, you must define a VLAN group that includes the VLAN in which the node resides. The Forwarding Layer 2 virtual server processes connections on a packet-by-packet basis. The Forwarding Layer 2 virtual server operates on a packet-by-packet basis with the following TCP behavior: the initial SYN request is sent from the client to the BIG-IP LTM virtual server. The BIG-IP LTM passes the SYN request to the node in the associated VLAN based on the routing decision. The source MAC address is preserved and the destination MAC changes based on routing.

Node sends SYN-ACK TCP packet back to LTM

Client sends ACK packet to acknowledge receipt of the SYN

LTM sends SYN-ACK TCP packet to Client

Client sends SYN request to LTM The Forwarding Layer 2 virtual

Figure: Client connection to the Performance HTTP virtual server when no idle server-side flow

Client sends GET request to LTM

LTM sends SYN request to node

ACK packet from the LTM

LTM sends ACK packet to Node

Client sends Data packet to LTM

LTM sends Data packet to node

LTM sends SYN request to node

ACK packet from the LTM

LTM sends Data packet to node

Node sends SYN-ACK TCP packet back to LTM

LTM sends SYN-ACK TCP packet to Client

LTM sends GET request to pool member

seq_num = 1, ack_num = 1 Client sends data Data

BIGIP LTM

LTM evaluates the packet for processing, looking only at the destination IP address.

SYN

seq num = 0

SYN_ACK

seq_num = 0, ack_num = 1

Data

Figure: TCP handshake for a Forwarding Layer 2 virtual server Forwarding IP virtual server The Forwarding IP virtual server type uses the Fast L4 profile. An IP forwarding virtual server forwards the packet directly to the next hop IP address specified in the client request. Therefore, when the BIG-IP

Client sends SYN request to LTM The Forwarding IP virtual

Client sends ACK packet to acknowledge receipt of the SYN

Figure: TCP handshake for a Forwarding IP virtual server

ACK LTM sends ACK packet to Node seq num = 1, ack num = 1 Client sends data Data Client sends Data packet to LTM

Node

server

The Reject virtual server type causes the BIG-IP system to immediately reject any traffic destined for the virtual server IP address. The Reject virtual server operates using the following TCP behavior: the initial SYN request is sent from the client to the BIG-IP LTM virtual server. The BIG-IP LTM virtual server immediately closes the connection by sending a TCP reset to the client. The following TCP flow diagram illustrates the TCP behavior for a Reject virtual server: BIGIP LTM Node Client SYN Client sends SYN request to the LTM Reject virtual server seq num = 0 SYN request sent from the client to the BIG-IP LTM Reject virtual server. The LTM immediately closes the connection by sending a TCP reset to the client RST LTM sends TCP Reset packet to client $seq_num = 0$ Figure: TCP behavior for a Reject virtual server **Related Content** • K55185917: Overview of BIG-IP virtual server types (12.x - 17.x) • K14163: Overview of BIG-IP virtual server types (11.x) • K12272: Overview of BIG-IP virtual server types (10.x) • K5017: Overview of BIG-IP virtual server types (9.x) **Al Recommended Content**

How can we improve this content?

Submit (©) Contact Support FOLLOW US **In** f **HAVE A** Support and Sales > **QUESTION?**

F5.com **DevCentral** MyF5

Read Support Policies Create Support Case Leave Feedback [+]

SUPPORT TASKS

Forwarding Layer 2 virtual server

Standard virtual server

provide independent and optimized TCP connections to client systems. transparent to the client and server.

Standard virtual server with a TCP profile handshake on the server side of the connection.

BIGIP LTM Client SYN $seq_num = 0$ SYN_ACK seq_num = 0, ack_num = 1

ACK

Note: The BIG-IP LTM system may initiate the server-side connection prior to the first data packet for certain Layer 7 applications, such as FTP, in which case the user waits for a greeting banner before

system. sending any data.

The TCP connection setup behavior for a Standard virtual server with Layer 7 functionality operates as follows: the three-way TCP handshake and initial data packet are processed on the client side of the connection before the BIG-IP LTM system initiates the TCP handshake on the server side of the connection. A Standard virtual server with Layer 7 functionality processes connections using the full proxy architecture. The following TCP flow diagram illustrates the TCP handshake for a Standard virtual server with

Layer 7 functionality: BIGIP LTM Client SYN $seq_num = 0$ SYN_ACK

seq num = 0, ack num = 1 ACK

Note: Starting with BIG-IP in 12.0.0, you can apply an HTTP profile to the Performance Layer4 virtual server. However, the addition of the HTTP profile does not alter how the Performance Layer4 packets are processed. Client SYN $seq_num = 0$ LTM evaluates the packet for processing, looking only at the destination IP address.

load balancing decision and passes the SYN request to the pool member.

BIGIP LTM

ACK

SYN

seq_num = 0

HTTP_GET

SYN

The LTM establishes an idle server-side flow by opening a TCP connection to the pool member

SYN

 $seq_num = 0$

SYN_ACK

seq_num = 0, ack_num = 1

ACK

seq num = 1, ack num = 1

HTTP_GET

HTTP_GET

The following TCP flow diagram illustrates the TCP handshake for a Forwarding Layer 2 virtual server:

SYN

 $seq_num = 0$

SYN_ACK

seq_num = 0, ack_num = 1

ACK

Data

Node

server

BIGIP LTM

seq_num = 0, ack num = 1 seq_num = 1, ack_num = 1 Client sends data

Client initiates a connection to the Performance HTTP virtual server LTM finds idle server-side flow, marks the connection as non-idle and sends client request over it

Client sends data Forwarding Layer 2 virtual server The Forwarding Layer 2 virtual server type uses the Fast L4 profile. The Forwarding Layer 2 virtual server does not have pool members to load balance and forwards packets based on routing decisions. The

HTTP_GET

SYN

 $seq_num = 0$

SYN_ACK

ACK

LTM passes the SYN request to the node in the associated VLAN based on the destination Layer 2 address seq_num = 0, ack_num = 1 seq_num = 1, ack_num = 1

Client

LTM system evaluates the packet for processing, the system looks only at the destination IP address. The Forwarding IP virtual server processes connections on a packet-by-packet basis. The Forwarding IP virtual server operates on a packet-by-packet basis with the following TCP behavior: the initial SYN request is sent from the client to the BIG-IP LTM virtual server. The BIG-IP LTM virtual server passes the SYN request to the next IP address in the associated VLAN, based on the destination IP address. The following TCP flow diagram illustrates the TCP handshake for a Forwarding IP virtual server:

Client

SYN

seq_num = 0

SYN_ACK

seq_num = 0, ack_num = 1

ACK

seq_num = 1, ack_num = 1

Reject virtual server

• Support Solution - K000148375: CFE troubleshooting steps are not working with an error - "Failover initialization failed: undefined" • Security Advisory - K000141008: RADIUS authentication vulnerability CVE-2024-3596 • Security Advisory - K000148584: Intel Xeon Processor vulnerability CVE-2024-23918 • Security Advisory - K000148583: Intel QAT Engine for OpenSSL vulnerability CVE-2024-33617

↑ Return to Top

* Was this information helpful?

May we contact you directly regarding this feedback?

ABOUT F5 EDUCATION

Newsroom

Careers

Corporate Information

Investor Relations

Contact Information

Certification LearnF5 Free Online Training **Communication Preferences**

Training