

Using Ixia Chariot to Assess the Performance of Unreliable Networks

Introduction

When testing wireless networks it is necessary to test both the maximum performance of the network and the performance at the periphery of wireless coverage. In mobile wireless networks it is also necessary to test the performance of the network during handoffs between base stations or access points. Ixia Chariot has a new feature called *pair re-initialization* that allows the user to increase the application’s resiliency to network outages so that testing can be conducted smoothly across network transitions. This application note describes the configuration options that are available to the user when testing unreliable networks.

How do network failures affect Chariot tests?

Most Chariot users have seen something like this at least once in their testing:

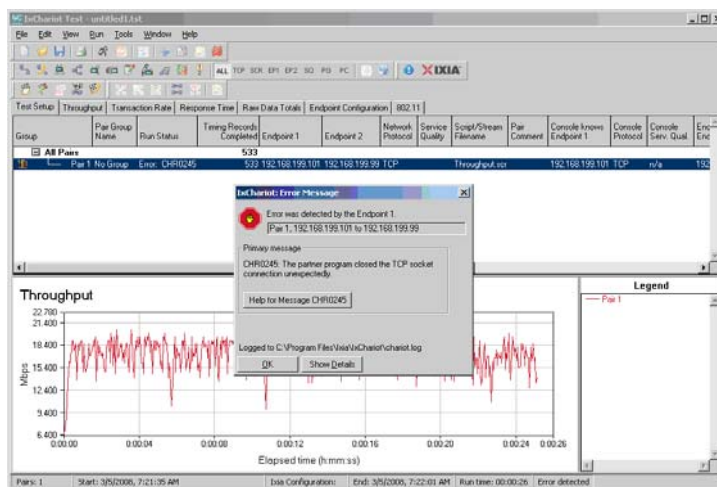


Figure 1. Common network-related Chariot error.

Chariot pairs fail when either the console or endpoint detects an error condition. Since TCP is connection-oriented and reliable, TCP errors are detected by the operating system and reported to the endpoint software agent. These failures can occur because of a timeout, high packet loss, unexpected socket closure (RST) or a network disconnect like a loss of signal. UDP and RTP pairs can fail for many of the same reasons, but their status is monitored by the endpoint software rather than the operating system. The Chariot User’s Guide has a covers the settings that apply to UDP and RTP pairs in Chapter 5 under Run Options and User Settings->Datagram Run Options.

Chariot error codes are grouped into numerical ranges by the type of error, see Table 1. Network errors are grouped in the CHR2XX series—these are the errors that can be handled by the pair re-initialization feature introduced in Ixia Chariot 6.40.

In older versions of Chariot, once a pair encountered an error it would automatically stop and return an error code. Depending on the test configuration, the entire test might have stopped when a single pair failed. Using *pair re-initialization*, users can instruct the Chariot console to automatically retry failed pairs during a test run instead of simply returning an error code.

Table 1. Error codes and categories.

Error Code	Error Type
CHR100-CHR199	Chariot endpoint software errors
CHR200-CHR299	Network-related errors
<i>CHR200</i>	<i>Connection attempt timed out</i>
<i>CHR202</i>	<i>An established connection failed</i>
<i>CHR204</i>	<i>No partner program is waiting for this connection</i>
<i>CHR245</i>	<i>The partner program closed the socket connection unexpectedly</i>
CHR300-CHR399	Chariot pair-specific errors
CHR400-CHR499	Chariot console software errors
CHR5600-5699	Ixia hardware configuration errors
CHR6XXX	Hardware Performance Pair errors
CHR7XXX	Miscellaneous errors

Pair Re-initialization Overview

Chariot pair re-initialization gives the user the option to ‘test through’ initialization and connection failures. Each time a failure is encountered, the pair can be instructed to re-attempt the connection any number of times after a user-defined pause. The pair re-initialization feature is disabled by default but it can be enabled via a checkbox in the ‘Run Options’ dialog. Pair re-initialization can occur during both the initialization and established phases of the connection. There are independent controls for errors that occur during each of these phases.

The term ‘connection’ typically implies a TCP socket, but Chariot can also reattempt UDP and streaming ‘connections’ with this feature. When running a non-streaming Chariot pair using the UDP or RTP protocol, i.e. `inquiry.scr`, Chariot uses a windowed-acknowledgement system to ensure reliable delivery much like ordinary UDP-based management protocols. These pairs follow an initialization and connection error sequence much like TCP. Streaming pairs have only one failure mode—the receiver times out waiting for data. By default, Chariot’s streaming receive timeout is 10 seconds. These topics are covered in detail by the Chariot User’s Guide available in the online help system.

When a pair performs a re-initialization during a test, you will see the following dialog box (Figure 2) at the end of the test. There is a CHRO469 warning that a pair re-init occurred during the test and it denotes the root cause; in this case a CHRO216 timeout on a streaming pair.

Group	Pair Group Name	Run Status	Time	per Unique Points(E2)
[-] Multicast Call G. 711				
[-] Pair 1	230.10.10.1:15001	Finished		n/a
[-] Pair 2	230.10.10.2:15002	Finished: Warning(s)		n/a
[-] Multicast Call G. 729				
[-] Pair 3	230.10.10.3:15003	Finished: Warning(s)		1
[-] Pair 4	230.10.10.4:15004	Finished: Warning(s)		n/a
[-] Unicast Call G. 711				
[-] Pair 5	No Group	Finished: Warning(s)		n/a
[-] Pair 6	No Group	Finished: Warning(s)		1
[-] Unicast Call G. 729				
[-] Pair 7	No Group	Finished: Warning(s)		n/a
[-] Pair 8	No Group	Finished: Warning(s)		1

IxChariot: Warning Message(s)

The following warning(s) occurred:

CHRO469: PAIR REINIT due to error CHRO216: Data not received within the timeo
 CHRO358: A substantial inactive time value was received for a timing record.

Message Help
 Show Details

To see the help message for a warning, double-click the warning, or select the warning and press the Message Help button.

To see additional details for a warning, select the warning and press the Show Details button.

OK

Figure 2. Pair re-initialized during test.

In the Chariot graphing pane you will see the graph line ‘cut out’ during periods when re-initialization occurred. This can be seen in Figure 3 below.

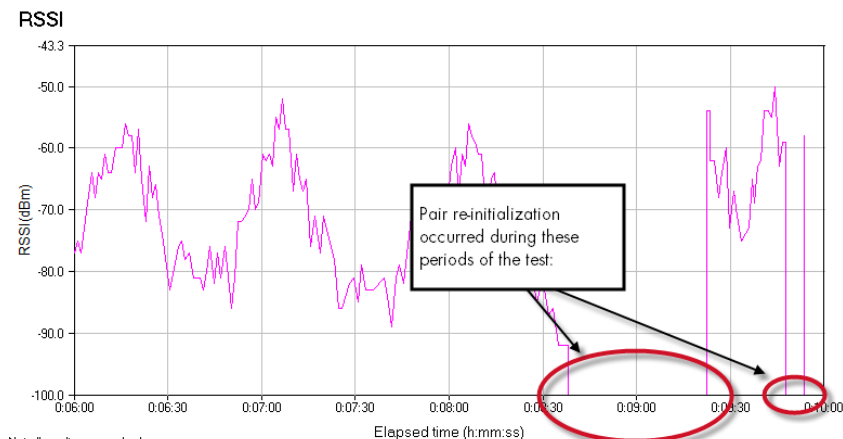


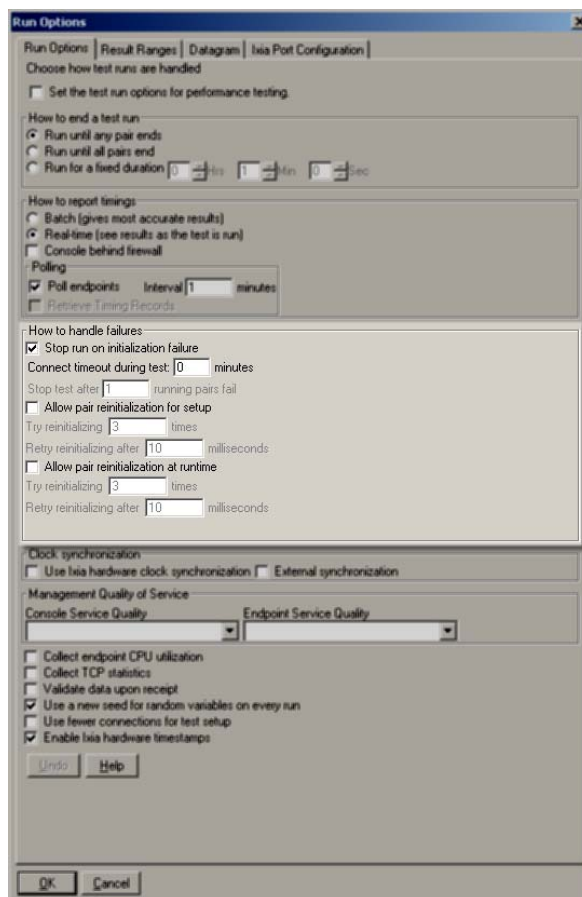
Figure 3. Throughput graph showing drops during re-initialization.

Finally, you can investigate the timing records in each re-initialized pair to see exactly how long the connection was unavailable. In Figure 4, you can see that this pair spent 39.726 seconds re-initializing itself during the test. Since this time is counted as ‘inactive time’ by Chariot, it will not affect the results of the pair itself, but it will affect the group averages which take into account the total duration of the test. These calculations are described in Chapter 9 of the User’s Guide.

Record Number	Elapsed Time (sec)	Measured Time (sec)	Inactive Time (sec)	Throughput (Mbps)	Bytes Sent by E1	Bytes Received by E2	Bytes Lost from E1 to E2	Total DGs Sent by E1	Total DGs Received by E2	Duplicate DGs Received by E2
258	286.331	1.174		0.007	1,180	1,000	180	59	50	0
272	305.160	1.163		0.007	1,160	1,000	160	58	50	0
273	306.495	1.335		0.006	1,300	1,000	300	65	50	0
274	307.795	1.299		0.006	1,320	1,000	320	66	50	0
275	310.701	2.906		0.003	2,920	1,000	1,920	146	50	0
276	312.029	1.327		0.006	1,300	1,000	300	65	50	0
277	313.610	1.580		0.005	1,540	1,000	540	77	50	0
278	313.730	2.115		0.004	2,020	1,000	1,020	101	50	0
279	356.617	1.161	39.726	0.007	1,180	1,000	180	59	50	0
280	357.638	1.681		0.007	1,660	1,000	660	54	50	0
281	360.025	2.326		0.003	2,320	1,000	1,320	116	50	0
282	361.088	1.062		0.008	1,060	1,000	60	53	50	0
283	362.150	1.062		0.008	1,060	1,000	60	53	50	0
284	363.387	1.236		0.006	1,240	1,000	240	62	50	0
285	364.483	1.095		0.007	1,100	1,000	100	55	50	0

Figure 4. Timing records showing a re-initialization.

Pair re-initialization Settings



Three options you can configure in the Run Options dialog box give you some control in failure-prone networks:

- *Stop run on initialization failure*

Initialization occurs when you click the Run button: the IxChariot Console contacts all Endpoint 1 computers, which in turn contact their Endpoint 2 partners. When you start a test, you are never completely sure whether all the endpoints can be reached. But if your test involves many different endpoints, you may want to run the test even if some of the endpoints are unavailable.

Checking the *Stop on initialization failure* box stops the run when any endpoint cannot pass all the initialization steps. If you leave the box cleared, the test will be run if at least one endpoint pair can be initialized. Those endpoints that cannot be initialized are omitted from the results and show errors.

- *Connect timeout during test*

You may be testing in noisy networks, where long connections are frequently dropped. IxChariot retries its connection attempts for the number of minutes you specify here. If that amount of time elapses and a connection still cannot be established, IxChariot declares a connection failure and issues the appropriate error message.

A connection attempt by an endpoint may consist of more than one `Sockets Connect` call, even if the timeout is set to zero. If the connection failure is due to a transient condition, such as network congestion, the endpoint will issue a fixed number of `Sockets Connect` calls per attempt. If the failure is due to a permanent condition, such as insufficient memory resources, the endpoint will issue one `Sockets Connect` call per attempt. This information applies to TCP connections only. See "Mapping Communication Commands to APIs" in the *IxChariot Script Development and Editing Guide* for more information about endpoint operating systems using TCP.

A value of 0 minutes means that connection failure is declared after the first unsuccessful series of connection attempts by the endpoints. The timeout option tracks errors encountered on `CONNECT_INITIATE` commands in a script; errors that occur on `SEND`, `RECEIVE`, or other commands will still cause a running test to stop. Thus, this timeout is most helpful in scripts with short connections.

- *Stop test after x running pairs fail*

You may want to let some pairs fail while the remainder of the pairs continue executing their scripts. IxChariot implements this option when the test enters the running state. Once in running state, IxChariot lets the specified number of pairs fail before terminating the test.

- *Allow pair re-initialization for setup*

When you select this option, IxChariot will attempt to reinitialize an endpoint pair that fails during the initialization phase of the test.

The re-initialization feature provides flexibility in test planning and execution. When running large-scale tests (up to 100,000 pairs), it is often desirable—if not necessary—for a test to proceed even if some of the pairs fail during initialization, and to allow IxChariot to reinitialize the pairs that fail.

When you select the *Allow pair re-initialization for setup* option, you must also set the following parameters:

- *Try reinitialization n times*

Specify the number of times that IxChariot should attempt to reinitialize the failed endpoint pair. The default is three attempts.

If IxChariot is not successful in reinitializing the pair after the specified number of attempts, initialization of the pair is considered to have failed. At this point, the pair will be included in the count of failed pairs (refer to the *Stop test after x running pairs fail* parameter above).

- *Try reinitializing after n milliseconds*

Specify the number of milliseconds to wait between reinitialization attempts. The default is to wait ten milliseconds between attempts.

- *Allow pair reinitialization at runtime*

When you select this option, IxChariot will attempt to reinitialize an endpoint pair that fails during the execution of the test.

The reinitialization feature provides flexibility in test planning and execution. When running large-scale tests (up to 100,000 pairs), it is often desirable—if not necessary—for a test to proceed even if some of the pairs fail during test execution, and to allow IxChariot to reinitialize the pairs that fail. As another example, in many WLAN tests it is not uncommon for a pair to fail if the performance endpoint moves out of the coverage area of the access point (AP) while the test is in progress. By using appropriate reinitialization options, you can allow IxChariot to reinitialize the pairs that fail.

When you select the *Allow pair reinitialization at runtime* option, you must also set the following parameters:

- *Try reinitialization n times*

Specify the number of times that IxChariot should attempt to reinitialize the failed endpoint pair. The default is three attempts.

If IxChariot is not successful in reinitializing the pair after the specified number of attempts, initialization of the pair is considered to have failed. At this point, the pair will be included in the count of failed pairs (refer to the *Stop test after x running pairs fail* parameter above).

- *Try reinitializing after n milliseconds*

Specify the number of milliseconds to wait between reinitialization attempts. The default is to wait ten milliseconds between attempts.

Recommended settings

The default re-attempt count is 3 and the default pause time is 10 ms. This pause time should be seen as a minimum setting under ideal conditions. If the test contains more than 10 pairs and there is a high number of reconnects then this should be raised to at least 500 ms. If you enable re-initialization at runtime with 10 attempts and a wait time of 500 ms, then on each runtime connection failure the pair will re-attempt the connection every 500 ms for 5 seconds. If you expect longer outages during the test then these parameters can be modified to fit the scenario.

How do real applications handle failures?

Network connections have two distinct states of operation: initialization, or *connect-ing*, and *connected*. The way an application handles network errors may differ depending on the state of the connection. Generally, applications can choose two options when they encounter a network error: return the error to the user for more input (“Would you like to try again?”) or automatically re-attempt the connection. Automatically re-attempting the connection may seem to be an obvious choice but it often takes many seconds for a TCP connection to fail so three automatic retries may force the user to wait 2-3 minutes or more before they can decide to cancel the operation. Chariot’s pair re-initialization options can be used to control error handling in both the initialization and connected states to accurately simulate real-world behavior.