APPLICATION NOTE

Using Port Control Protocol with IxLoad

Introduction

An important component of today`s networks comprises a very high number of residential gateways with NAT 44 capabilities (Network Address Translator IPv4/IPv4). If other devices such as Carrier Grade NATs (CGNs), firewalls, and enterprise NAT64 are considered as well, then the probability to get a packet translation in an end-to-end communication is extremely high. Therefore a practical need of having an increased level of control over such devices (that perform translations) emerged. Port Control Protocol (PCP), currently defined by IETF draft, draft-ietf-pcp-base-29, provides the mechanism for a host located behind a NAT or firewall to define how downstream packets (that is, packets that are first initiated from the external network side) are handled by the NAT or firewall device.

Why does PCP matter?

Using PCP, applications operating behind a NAT or firewall will be able to accept inbound traffic (remote peers initiated sessions) by creating specific mappings into the NAT or firewall`s translation table: correspondence between an external IP address and port to an internal IP address and port.

This functionality allows a wide range of devices that act as application servers to operate flawlessly behind a NAT or firewall (e.g. security cameras, storage servers, HTTP servers, etc.). An important benefit of PCP is that it helps to reduce the overhead traffic and battery consumption (in the case of mobile devices) caused by the common keepalive traffic generated by the application client devices (also located behind NAT) to ensure that their session is kept active by the NAT.

How does PCP work?

PCP implementation follows a client–server architecture using UDP as the underlying transport protocol. The PCP client typically resides on the host “behind” a NAT or firewall while the PCP server sits on the NAT or firewall device itself as shown in Fig 1:

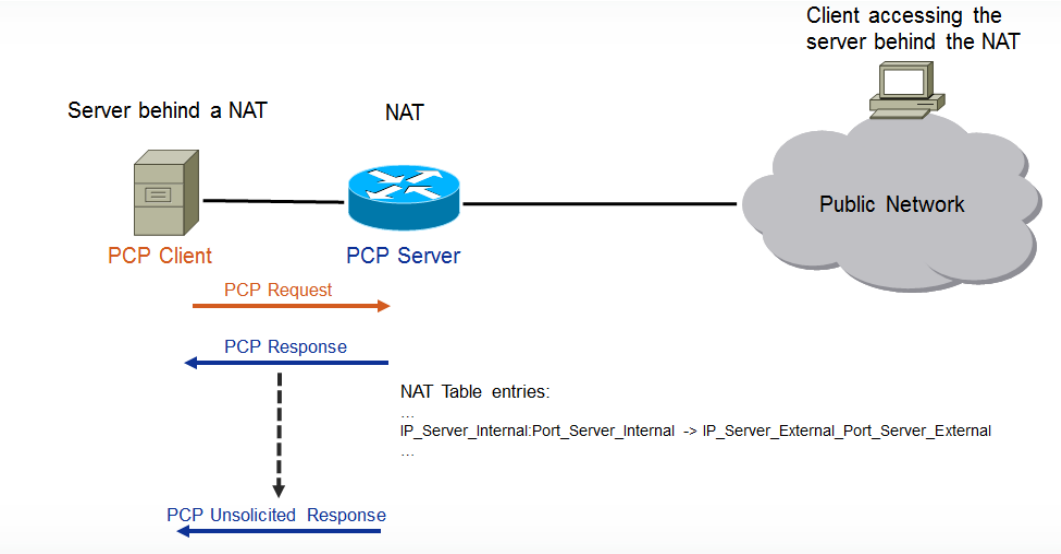


Figure 1. PCP deployment scenario overview

The PCP client is responsible for sending PCP requests containing detailed information about the mapping it wants to create or manage. In turn the PCP server answers using PCP response messages for the previously-received PCP request. In some situations the server can send unsolicited responses to clients, informing them of the new state of their mappings.

Opcodes, a seven-bit field in the PCP packet header, specify the PCP operation to be performed. According to IETF draft, draft-ietf-pcp-base-29, the following are definitions of existing ***Opcodes***:

- ***MAP Opcode***: controls forwarding from a NAT device (or firewall) to an Internal Host by creating an explicit dynamic mapping between an Internal IP Address:Port and an External IP Address:Port. The purpose of an MAP mapping is to receive inbound traffic from any remote endpoint, not from only one specific remote endpoint.
- ***PEER Opcode***: creates a new dynamic outbound mapping to a remote peer's IP address and port, or extends the lifetime of an existing outbound mapping. Mapping created or managed by PEER Opcode behaves almost exactly like an implicit dynamic mapping created as a side-effect of a packet (e.g., TCP SYN) sent by the host.
- ***ANOUNCE Opcode***: used mainly as a rapid recovery mechanism. When the PCP server loses its state (for example, if it is rebooted), it sends the ANNOUNCE response to its PCP clients (using either a multicast address if a multicast network exists on its local interface or, a unicast message if configured with the IP address of PCP client).

As per IETF draft, draft-ietf-pcp-base-29, PCP can be used in various deployment scenarios, including:

- Basic NAT [RFC3022]
- Network Address and Port Translation [RFC3022], such as commonly deployed in residential NAT devices
- Carrier-Grade NAT [I-D.ietf-behave-lsn-requirements]
- Dual-Stack Lite (DS-Lite) [RFC6333]
- Layer-2 Aware NAT [I-D.miles-behave-l2nat]
- Dual-Stack Extra Lite [RFC6619]
- NAT64, both Stateless [RFC6145] and Stateful [RFC6146]
- IPv4 and IPv6 simple firewall control [RFC6092]
- IPv6-to-IPv6 Network Prefix Translation (NPTv6) [RFC6296]

How PCP works in IxLoad

IxLoad is a scalable solution for testing converged multiplay services, application delivery platforms, and security devices and systems. IxLoad emulates data, voice, and video subscribers over a variety of access protocols such as DHCP, IPsec, PPPoE, L2TP, 6RD, and DSLite to ensure quality of experience (QoE). IxLoad also applies malware and distributed denial of service (DDoS) attacks for security effectiveness and accuracy testing.

IxLoad 6.10 EA is the first product to market that offers a comprehensive, scalable, and flexible PCP test solution, which allows emulation of PCP clients under three main PCP test scenarios:

1. PCP Client running over IP
2. PCP Client running over Emulated Router (acting as a third-party for IP hosts)
3. PCP Client running over DSLite

IxLoad`s PCP client implementation can be used to assess PCP-Server-capable devices under test (DUTs) in the following scenarios:

- Functional testing to assure proper PCP functionality and compliance
- Performance testing to assess maximum concurrent sessions and maximum session initiation rate capabilities
- Application protocol performance running over previously-established PCP Sessions

POINTS TO NOTE

1. IxLoad can emulate PCP Client functionality over a wide range of IxLoad stack plugins: PCP client over IP, PCP client over Emulated Router, PCP client over DS-Lite.
2. IxLoad does not support PCP server emulation. Thus, for a PCP test, a PCP-server-capable DUT is required.
3. The PCP Helper plugin is mandatory when you want L4/7 traffic to use the address mappings that are negotiated by the protocol. The PCP Helper plugin is not defined by the standard, it is an Ixia-designed component that is inserted as a network plugin into the L4/7 client network stack. Its only function is to assure that L4/7-generated packets are correctly forwarded between the L4/7 activities (please refer to below scenario example for a thorough explanation).

Scenario

The next section will go through the steps required to configure a *PCP Client running over IP* test as shown in Figure 2.

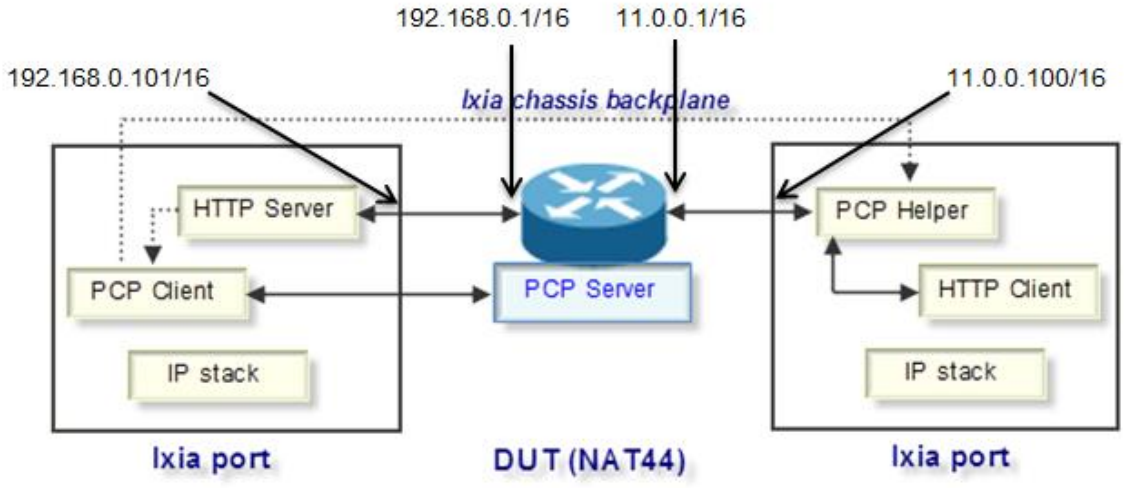


Figure 2. PCP Client running over IP

This scenario will simulate an HTTP Server activity located behind a NAT device (DUT). The HTTP server will be serving web pages to HTTP clients. IxLoad will emulate a private network that will host an HTTP Server activity (acting as PCP clients) and a public network that will host HTTP Clients. Between the public and private network there will be a DUT (PCP Server) performing all required translations from one network to another.

1. Start IxLoad and add two NetTraffics (originate and terminate). For the Terminate NetTraffic, add an HTTP Client activity and for the Originate NetTraffic, add HTTP Server activity:

Figure 3. Originate and Terminate NetTraffics

2. Configure the IP stack parameters for the client and server networks as shown below:

HTTP Server

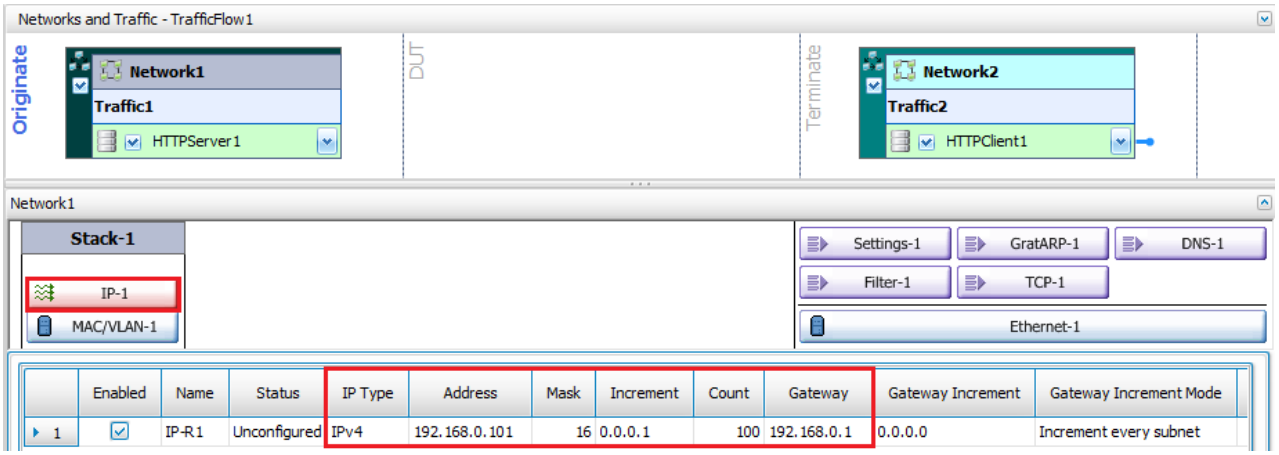


Figure 4. HTTP Server IP configuration

HTTP Client

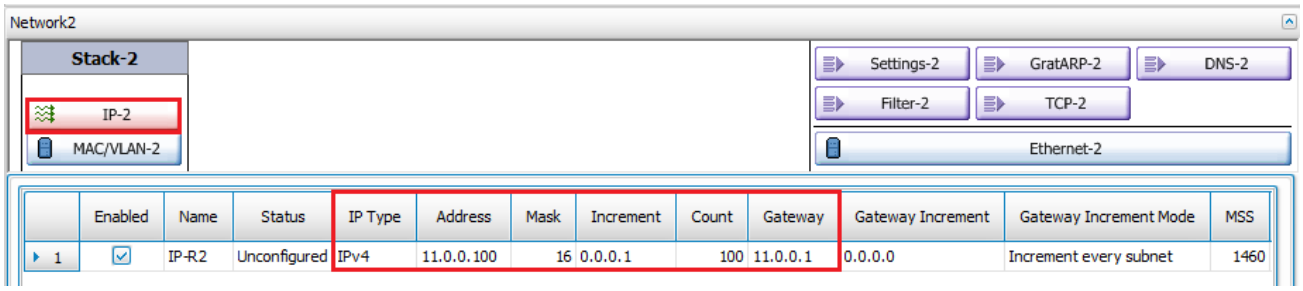


Figure 5. HTTP Client IP configuration

Special attention is required when configuring the above IP addresses and parameters:

- The HTTP Server (acting as PCP Client), which is located behind the NAT device, has a private IP address and its gateway is pointing to the private interface of the NAT device. Configure the ***Count*** parameter for the HTTP Server IP Stack as the number of PCP Clients to emulate.

- The HTTP Client, which is located on the public network, has a public IP address and its gateway is pointing to the public interface of the DUT.
- Make sure that the DUT is properly configured according to above specifications (the public IP is 11.0.0.1/16, while the private IP is 192.168.0.1/16).

To make sure that the application (http) traffic will follow the correct path between HTTP Clients and the HTTP Server according to the IP mapping(s) performed by the NAT device, we need to configure the ***PCP Helper*** plugin for the client HTTP IP stack. All HTTP-Client-generated packets will be forwarded by the ***PCP Helper*** plugin to the proper IP:Port to reach the HTTP Server, according to the existing PCP mapping.

3. To add the ***PCP Helper*** plugin, first select the IP stack layer of the HTTP client NetTraffic (terminate NetTraffic):

Figure 6. Terminate IP Stack

4. From the ribbon menu, click on the ***Add Plugins*** button; then, from the drop down, select ***Add above***. In the new list that will appear, select ***PCP Helper***:

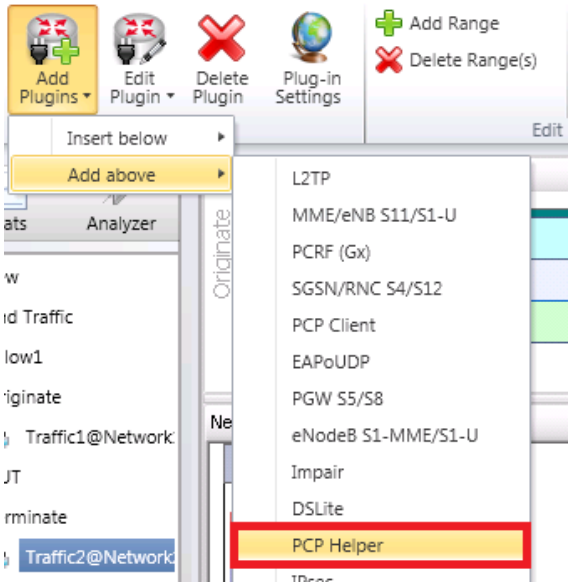


Figure 7. PCP Helper plugin

5. The newly-added ***PCP Helper*** plugin will be graphically shown as an icon at the IP Stack level:

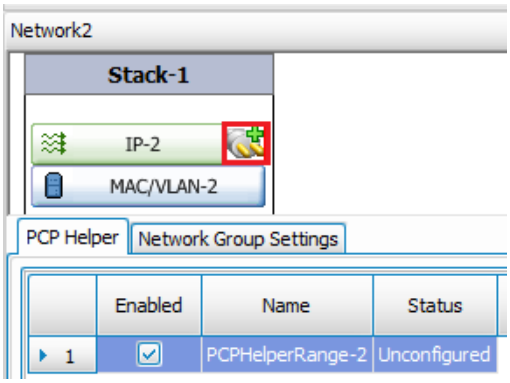


Figure 8. PCP Helper icon

The PCP Helper plugin does not require additional configuration since its primary use is to assist the HTTP Client activity to correctly forward the packets according to the relevant IP address mapping done by the DUT. It is enough just to add it as a network plugin as shown in Figure 8.

Once done with the ***PCP Helper,*** proceed with configuring the ***PCP Client*** plugin.

6. To add the ***PCP Client*** plugin, first select the IP stack layer of the HTTP Server NetTraffic (originate NetTraffic):

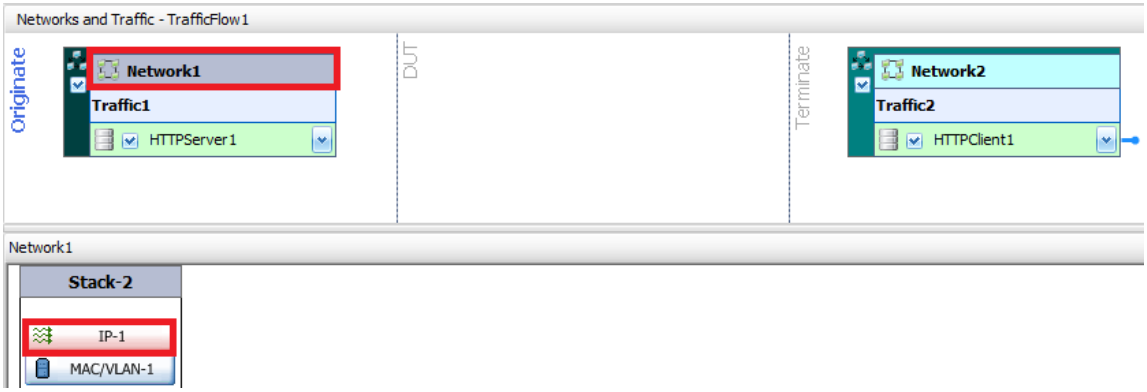


Figure 9. Originate IP Stack

7. From the ribbon menu, click on ***Add Plugins*** button; then from the drop down, select ***Add above***. In the new list that will be shown, select ***PCP Client***:

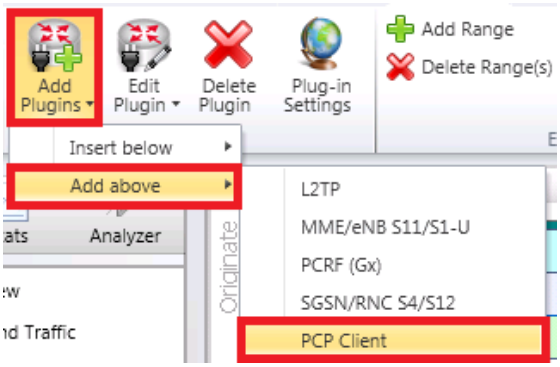


Figure 10. PCP Client plugin

8. Select the newly-added ***PCP Client*** stack element and go to the ***Network group Settings*** tab. In the new window, configure the ***Associated networks*** field with the corresponding network (such as Network2) on which you have configured the ***PCP Helper*** plugin:

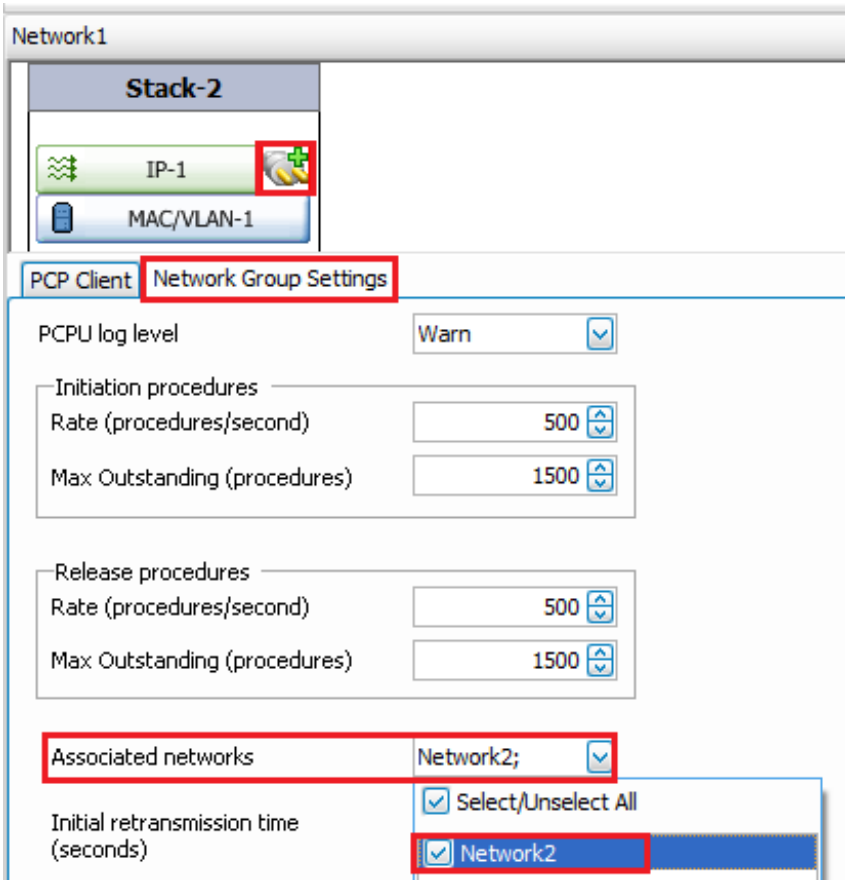


Figure 11. PCP Client Network Group Settings

Also in the ***Network Group Settings*** tab, other important PCP parameters can be configured such as:

- Initiation procedures:
  - Rate (procedures/second): procedure (MAP and PEER) initiation rate that can be initiated on this network group per second
  - Max Outstanding (procedures): maximum number of "in progress" procedures (MAP and PEER) that can be sent by all PCP Clients during startup
- Release procedures:
  - Rate (procedures/second)
  - Max Outstanding (procedures)
- Initial retransmission time
- Maximum Retransmission Count
- Maximum Retransmission Time
- Maximum Retransmission Duration

9. Next we need to configure the PCP Server IP address (DUT private IP address to be used as the destination for the PCP Client`s requests). This can be done from the ***PCP Client*** tab (***PCP Client Stack*** element):

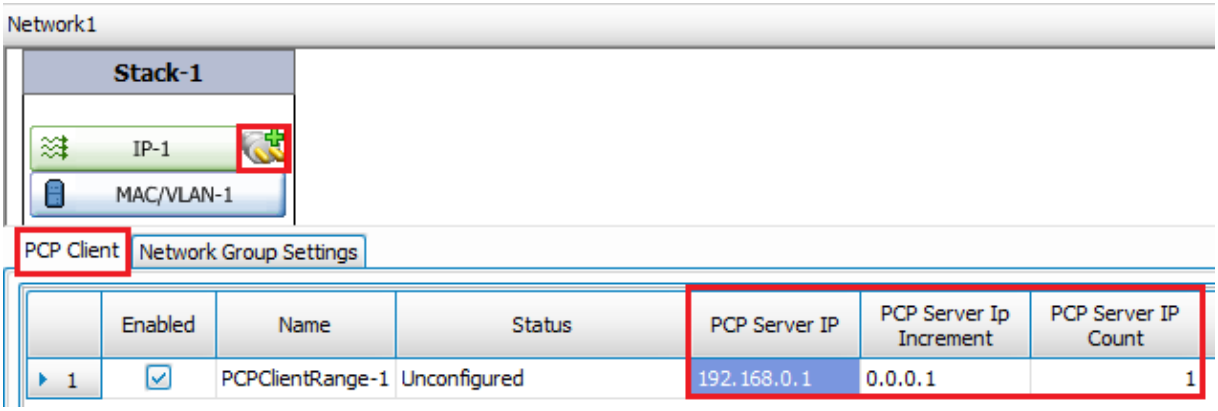


Figure 12. PCP Client configuration

10. Other PCP-related parameters can be found in the **Option** tab as shown in figure 13:

- Requested Lifetime
- Send PREFER_FAILURE
- Send Filter
- Suggested External IP
- Suggested External Port
- Send Zero Port in Delete Request
- Don`t Renew Mappings
- Force Session Lifetime

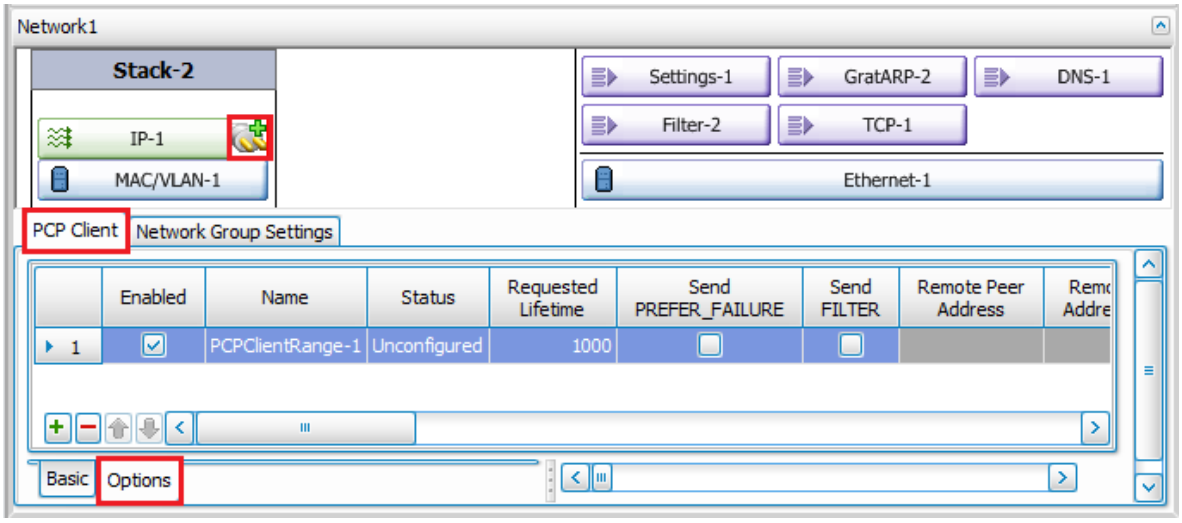


Figure 13. PCP Client, Option Tab

11. For the HTTP client, configure a GET request command of 1024k, and choose as destination *Traffic1_HTTPServer1:80*:

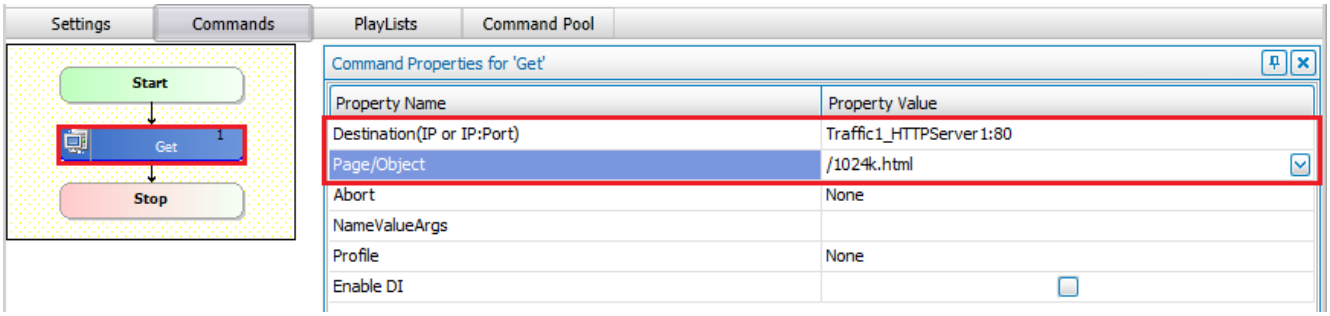


Figure 14. HTTP Client GET Command

12. Add client and server Ixia ports from the **Port Assignments** pane for each NetTraffic. [Verify that these ports are properly connected to the correct ports on the DUT – see figure 2]
13. Set the test objective to 100 Simulated Users in **Timeline and Objective.**

The test objective configured above refers to the simulated HTTP Clients. The actual number of PCP Clients to be emulated is specified by the **IP Count** parameter for the HTTP Server IP Stack (Originate NetTraffic).

14. To see PCP-related statistics (as with any other network plugin) you first need to ***Enable Network Diagnostics*** from ***Test Options***:

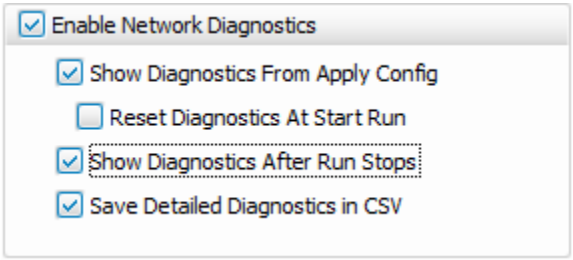


Figure 14. Enable Network Diagnostics options

15. Start the test by pressing the green play button.

Analyzing the results

As soon as Ixia test ports get configured, the view will automatically switch to Stat View. Below you can find a brief description of all PCP-related statistics:

- ***PCP – General*** is the most important statistic that you should begin with. It provides insight information into the PCP message exchange, like MAP Requests Initiated/Succeeded/Failed, MAP Renewals Initiated/Succeeded/Failed, Map Updated with Success/Failure, etc.

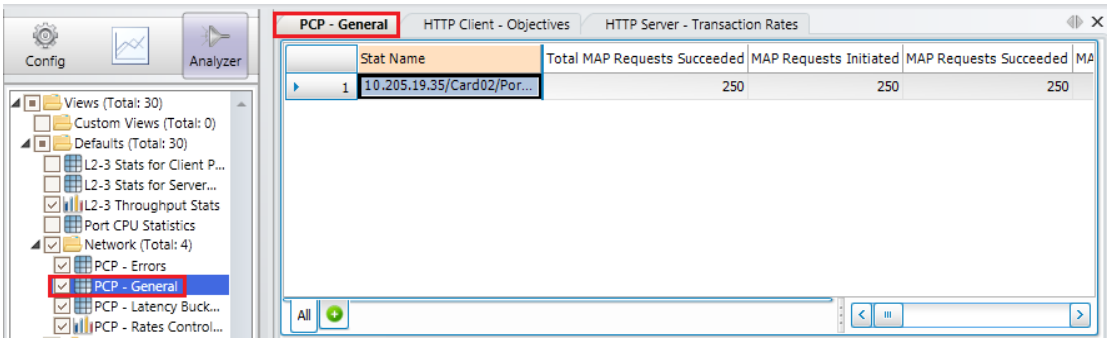


Figure 15. PCP – General stat view

- ***PCP – Rates Control – All Ports*** extremely useful, especially while assessing the PCP Server`s (DUT) performance, such as: MAP Rate, Activation Rate, Deactivation Rate, Renewal Rate.

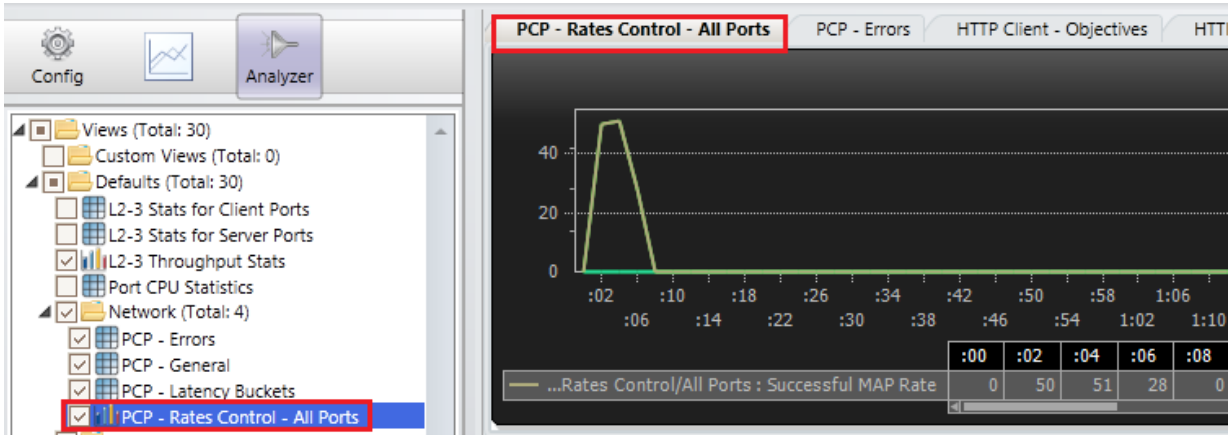


Figure 16. PCP – Rates Control stat view

- ***PCP – Latency Buckets*** statistic provides buckets of time intervals in which the specified number of MAP requests initiated by the PCP client were completed.

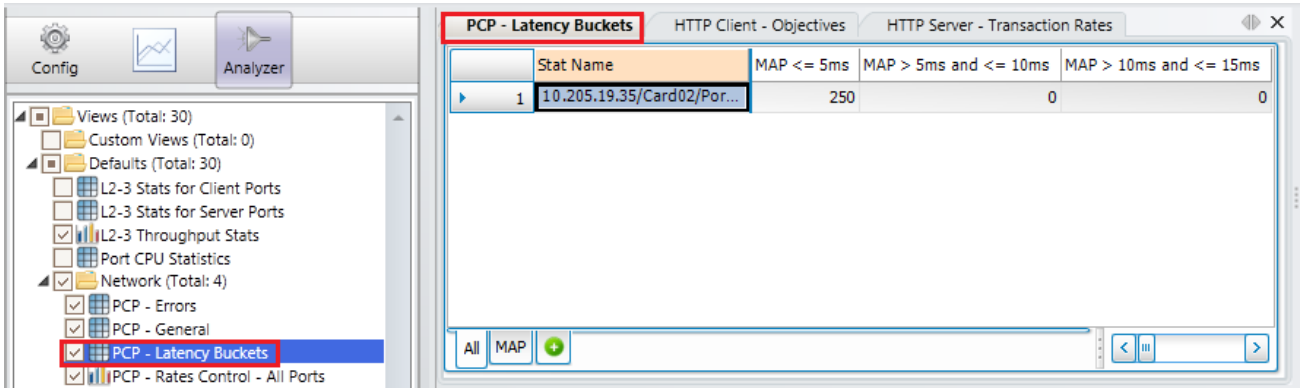


Figure 17. PCP – Latency Buckets stat view

- ***PCP – Errors*** indicates the error result codes that are returned by the PCP Server (DUT).

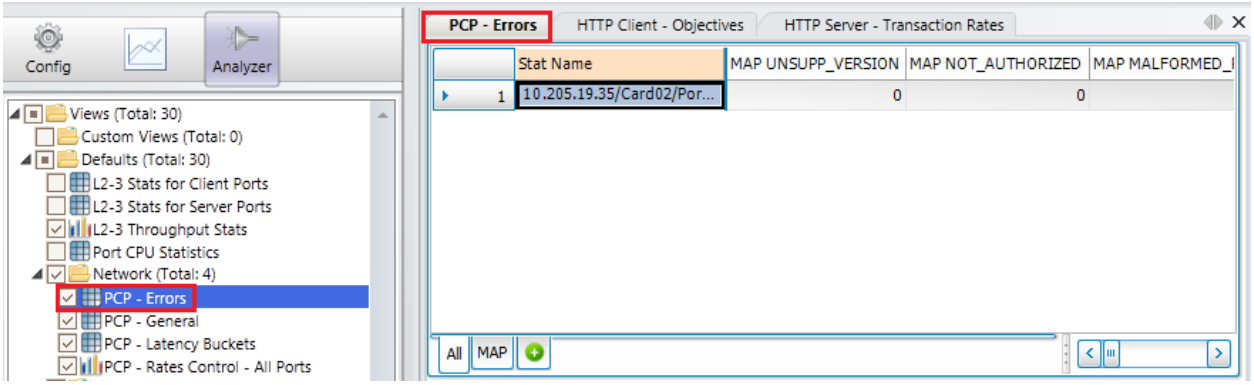


Figure 18. PCP Error stat view

- Once you have verified PCP-related statistics, look at the HTTP stats to verify L7 traffic traversing between the client and server, such as Simulated Users, Throughput, Transactions, Failures, etc.

Figure 19. HTTP Statistics

Tips and Tricks

For all PCP-related stat views, you have the option to drill down and monitor the following Conditional statistics:

- Per-session statistics: Statistics for each session created from a specific range.
- Per-range statistics: Aggregations of the per session statistics.

First select the relevant statistics you are interested in, and then press right click:

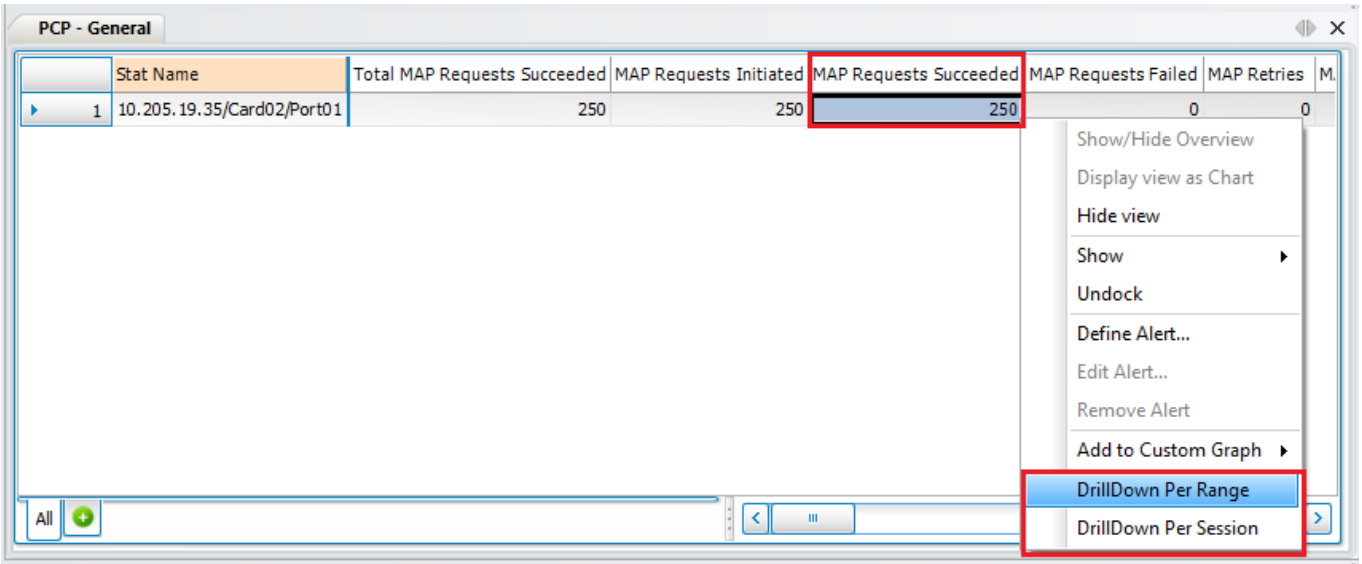
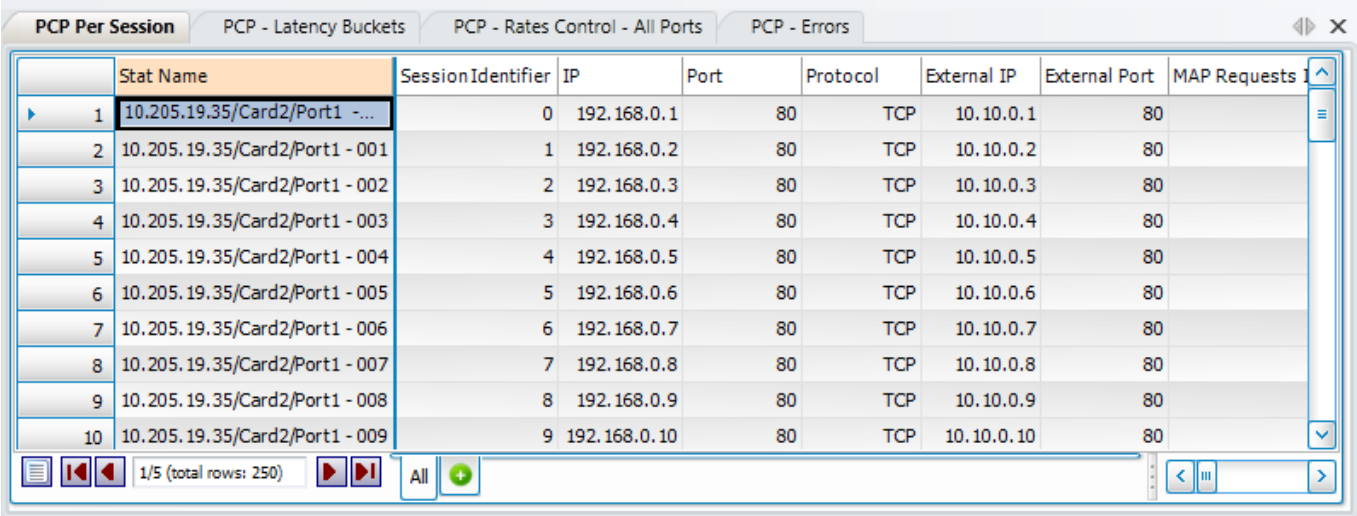


Figure 20. DrillDown Per Range/Session options

Once the desired drill-down type is selected (per range or per session), a new view will be generated: **PCP Per Session**.



	Stat Name	Session Identifier	IP	Port	Protocol	External IP	External Port	MAP Requests
1	10.205.19.35/Card2/Port1 -...	0	192.168.0.1	80	TCP	10.10.0.1	80	
2	10.205.19.35/Card2/Port1 - 001	1	192.168.0.2	80	TCP	10.10.0.2	80	
3	10.205.19.35/Card2/Port1 - 002	2	192.168.0.3	80	TCP	10.10.0.3	80	
4	10.205.19.35/Card2/Port1 - 003	3	192.168.0.4	80	TCP	10.10.0.4	80	
5	10.205.19.35/Card2/Port1 - 004	4	192.168.0.5	80	TCP	10.10.0.5	80	
6	10.205.19.35/Card2/Port1 - 005	5	192.168.0.6	80	TCP	10.10.0.6	80	
7	10.205.19.35/Card2/Port1 - 006	6	192.168.0.7	80	TCP	10.10.0.7	80	
8	10.205.19.35/Card2/Port1 - 007	7	192.168.0.8	80	TCP	10.10.0.8	80	
9	10.205.19.35/Card2/Port1 - 008	8	192.168.0.9	80	TCP	10.10.0.9	80	
10	10.205.19.35/Card2/Port1 - 009	9	192.168.0.10	80	TCP	10.10.0.10	80	

Figure 19. PCP Per Session conditional view

If required, additional filtering and sorting can be performed. Say one is interested in monitoring all the PCP sessions that have a mapping with external port 80. First select **PCP Per Session** view (on which the filtering/sorting will apply on) then click on the **Filter/Sort** button from the ribbon:

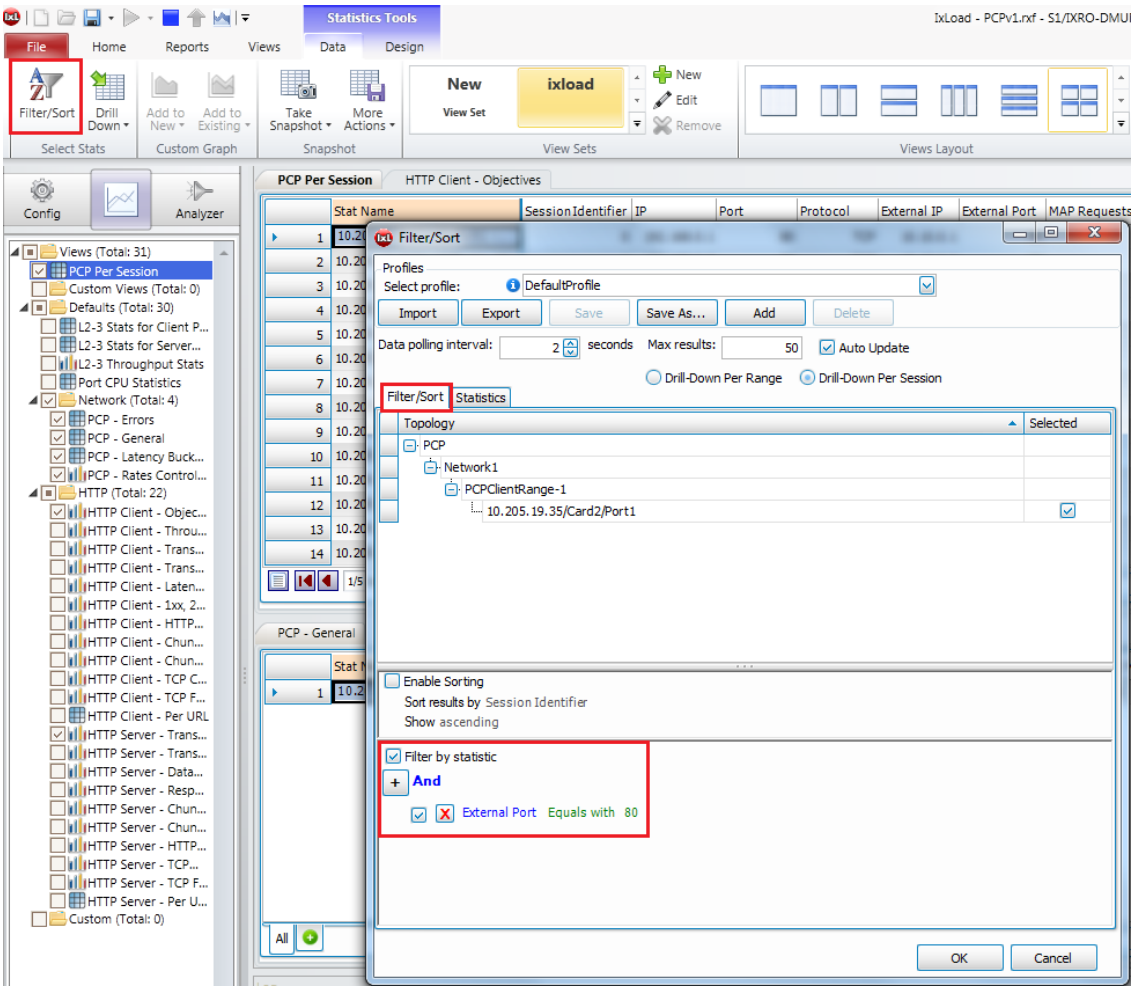


Figure 19. Filter/Sort dialog

A new dialog will open from where one can configure various sorting and filtering rules: in our case we defined a new filter rule: External port Equals with 80. From now on, the PCP Per Session view will display statistics only for those sessions matching our newly-created rule.

Conclusions

The PCP protocol implementation on residential, enterprise, and carrier-grade routers plays an extremely important role in the forwarding decisions of application traffic. Potential issues related to the PCP Server service running on those devices, directly impacts the performance, capacity, and resilience of the applications traversing those networks. IxLoad provides the flexibility and scale required to validate the performance and capacity of CGNs with realistic application traffic, allowing service providers and enterprise network operators to discover functional and performance issues earlier, before the production deployment.