ICON-PEP-3.0

Icon-PEP Administration Guide

Isode

Table of Contents

Chapter 1 Icon-PEP Overview..... 1

This chapter contains a general overview of Icon-PEP.

Chapter 2 Initial Setup of Icon-PEP..... 5

This chapter describes how to perform initial setup of Icon-PEP.

Chapter 3 Configuring Icon-PEP..... 10

This chapter describes how to configure Icon-PEP.

Chapter 4 Operating and Monitoring Icon-PEP..... 22

This chapter describes how to operate and monitor Icon-PEP.

Chapter 5 General management API..... 31

This chapter describes the web-API that allows external tools to manage Icon-PEP.

1 Software Version

This guide is published in support of Isode Icon-PEP 3.0. It may also be pertinent to later releases. Please consult the release notes for further details.

2 Readership

This guide is intended for an administrator to set up and operate Icon-PEP.

3 How to use this guide

It is recommended that all administrators read the entire manual.

4 Typographical conventions

The text of this manual uses different typefaces to identify different types of objects, such as file names and input to the system. The typeface conventions are shown in the table below.

Object	Example
File and directory names	*isoentities*
Program and macro names	**mkpasswd**
Input to the system	`cd newdir`

5 File System placeholders

A number of directory names are given in the text, and the actual locations (below) vary depending on the target platform.

Name	Place holder for the directory used to store...	UNIX
(ETCDIR)	System-specific configuration files.	*/etc/isode/icon-pep*
(SBINDIR)	Programs run by the system administrators.	*/opt/isode/icon-pep/sbin*
(LOGDIR)	Log files.	*/var/log/isode/icon-pep*

6 Support queries and bug reporting

A number of email addresses are available for contacting Isode. Please use the address relevant to the content of your message.

1. For all account-related inquiries and issues: customer-service@isode.com. If customers are unsure of which list to use then they should send to this list. The list is monitored daily, and all messages will be responded to.
2. To provide keys necessary to activate products, send the generated string to isode.support@isode.com along with information on what is being evaluated or what has been purchased.
3. For all technical inquiries and problem reports, including documentation issues from customers and evaluators: isode.support@isode.com. Customers should include relevant contact details in initial calls to speed processing. Messages which are continuations of an existing call should include the call ID in the subject line.
4. Customers may also submit support queries through the customer section of the Isode web site using the URL provided. Customers with silver or gold support may also submit support queries by telephone.
5. For all sales inquiries and similar communication: sales@isode.com.

Bug reports on software releases are welcomed. These may be sent by any means, but electronic mail to the support address listed above is preferred.

Isode sends release announcements and other information to the Isode News email list, which can be subscribed to from the address https://www.isode.com/news/.

7 Export controls

Icon-PEP may use TLS (Transport Layer Security) to encrypt HTTP traffic between the service and the web-browser, and also when querying an OAuth server. When TLS is enabled, Icon-PEP is subject to UK Export Controls. For some countries (at the time of shipping this release, these comprise all EU countries, United States of America, Canada, Australia, New Zealand, Switzerland, Norway, Japan), these Export Controls can be handled by administrative process as part of evaluation or purchase.

For other countries, a special Export License is required. This can be applied for only in context of a purchase order for Icon-PEP.

The TLS feature of 3.0 is enabled by a TLS Product Activation feature. This feature may be turned off, and 3.0 without this TLS feature is not export controlled. This can be helpful to support evaluation of 3.0 in countries that need a special export license. Note that when TLS is disabled, Icon-PEP is still able to forward TLS-encrypted connections such as HTTPS over S'5066 because no encryption or decryption occurs within Icon-PEP.

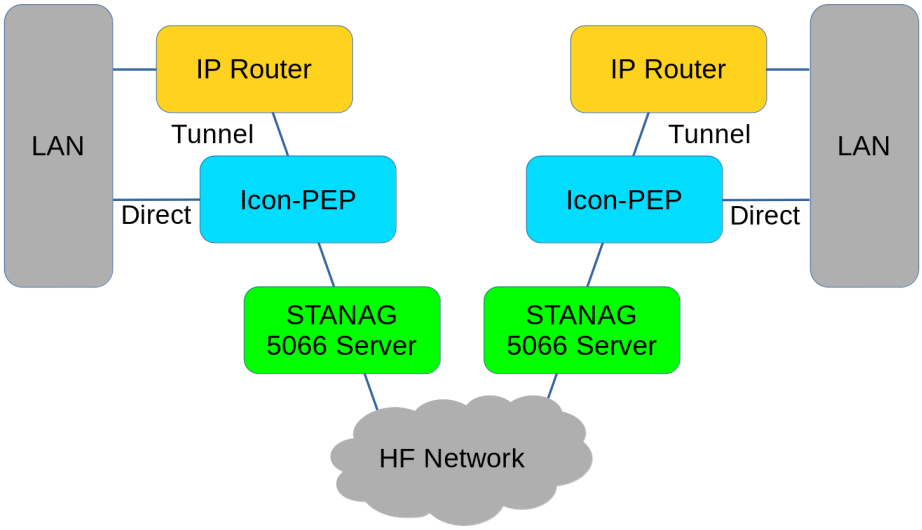
Isode strongly recommends that all operational deployments of Icon-PEP use the export-controlled TLS feature.

You must ensure that you comply with these Export Controls where applicable, i.e. if you are licensing or re-selling Isode products. All Isode Software is subject to a license agreement and your attention is also called to the export terms of your Isode license.

Chapter 1 Icon-PEP Overview

This chapter contains a general overview of Icon-PEP.

1.1 Icon-PEP Functionality

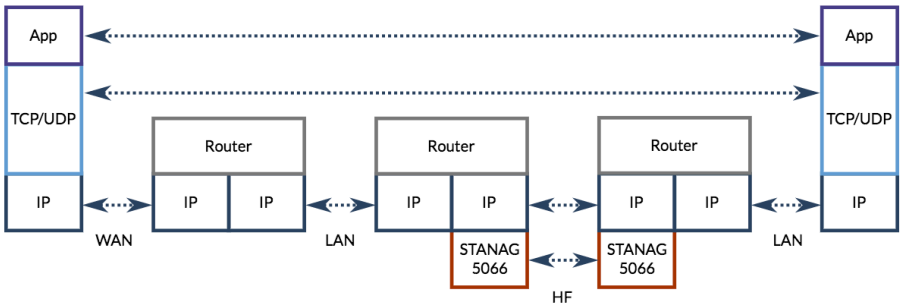


Icon-PEP enables provision of IP services over an HF network, with performance optimizations for TCP, TLS, HTTP/HTTPS (Web) and TAK connections.

Icon-PEP operates over the STANAG 5066 HF Link layer, and connects to a STANAG 5066 server such as Isode's Icon-5066 product. In "Tunnel" mode, Icon-PEP connects directly to an IP router, as illustrated above. Icon-PEP communicates with a peer server using standard protocols, to enable transfer of IP packets and proxied data streams between a pair of routers.

In addition, using the "Direct"-mode NAT or listener configuration options, connections for selected protocols may be made directly to/from the LAN without needing to go via an IP router. This is compatible with a "Tunnel" configuration on the remote. Alternatively both local and remote could be running "Direct" configurations.

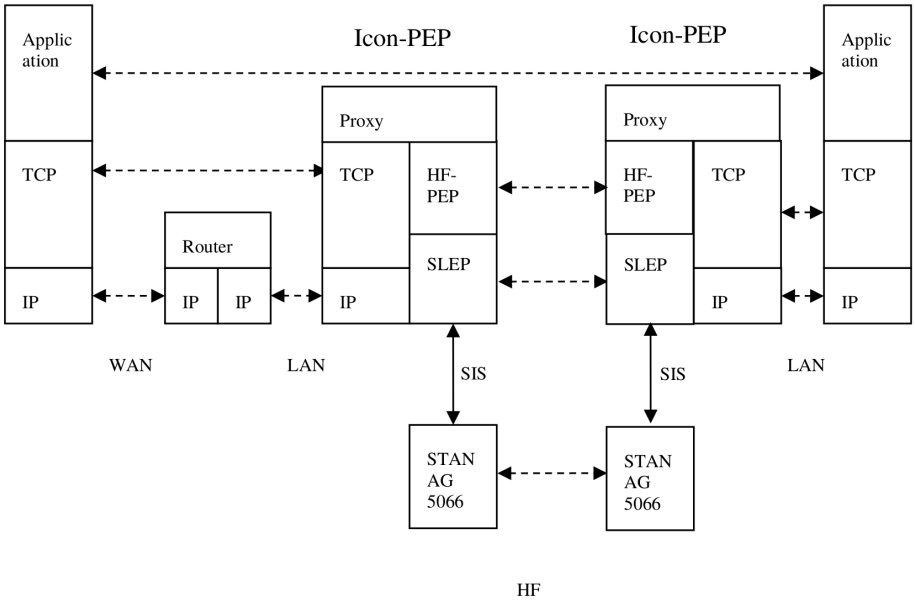
1.1.1 STANAG 5066 IP Client Service



STANAG 5066 Ed4 Annex U defines "IP Client" services for operating an IP subnet over HF, as shown in the diagram above. The approach is discussed in detail in the Isode white paper STANAG 5066 IP Client & Providing IP Services over HF Radio: https://www.isode.com/whitepaper/providing-ip-services-over-hf-radio.

Icon-PEP supports this protocol, which is suitable for some low volume services such as ICMP Ping, and it works acceptably for some specialized military applications and services such as DNS Lookup.

1.1.2 HF-PEP Service



Use of IP Client leads to poor performance for bulk applications and in particular for TCP and HTTP. To address this, Icon-PEP includes a PEP (Performance Enhancing Proxy) architecture to efficiently provide support.

In this TCP Proxy architecture, an application is communicating over TCP, running over IP in the normal manner. The TCP connection from each application is peered with Icon-PEP, rather than the other application. Icon-PEP then communicates using the HF-PEP protocol specified in STANAG 5066 Ed5 Annex X (based on S5066-APP9: https://www.isode.com/whitepaper/stanag-5066-application-protocol-series/).

HF-PEP operates over SLEP (SIS Layer Extension Protocol), specified in STANAG 5066 Ed5 Annex W (based on S5066-APP3: https://www.isode.com/whitepaper/stanag-5066-application-protocol-series/). SLEP provides the Stream Services used by HF-PEP. SLEP communicates over STANAG 5066, using the local STANAG 5066 SIS (Subnet Interface Service) to connect. STANAG 5066 peers communicate over an HF network, as shown.

Icon-PEP can multiplex TCP connections over a single HF link, so that a single STANAG 5066 SAP can be shared by multiple TCP connections and multiple applications running over TCP.

Further details on HF-PEP and performance measurements are provided in the Isode white paper Providing TCP Services over HF Radio: https://www.isode.com/whitepaper/providing-tcp-services-over-hf-radio.

1.1.3 Additional proxy layers for HF-PEP

Proxies for additional protocol layers above TCP are available to improve performance. These all follow S'5066 Annex X:

1. TLS: By issuing a server certificate to Icon-PEP, a TLS-encrypted stream may be terminated at Icon-PEP on the initiator side, avoiding the costly round-trips required by the TLS protocol. The responder-side Icon-PEP acts as a TLS client to connect to the final end-point TLS server.

2. TLS + client auth: Client authentication may be enforced on the initiator side. The client identity is passed over HF, allowing the correct client certificate to be selected on the responder side.
3. HTTP/2: An HTTP/2 proxy is available, layered over TLS, which maps the HTTP/2 substreams onto an HF-optimized protocol. Currently this is only used for TAK support.
4. TAK (Team Awareness Kit): Server-to-server TAK consists of protobuf over gRPC (Google Remote Procedure Call) over HTTP/2 over TLS. Icon-PEP supports TAK monitoring and TAK thinning, which reduces the traffic level to meet the bandwidth available. TAK status checks are intercepted and generated.

1.1.4 Connections to STANAG 5066 Server

Icon-PEP connects to one or more STANAG 5066 servers using one or more connections using the STANAG 5066 SIS protocol. Two types of SIS connection are supported:

1. IP Client (default SAP: 9)
2. HF-PEP (default SAP: 13)

1.1.5 Connections to Router (GRE)

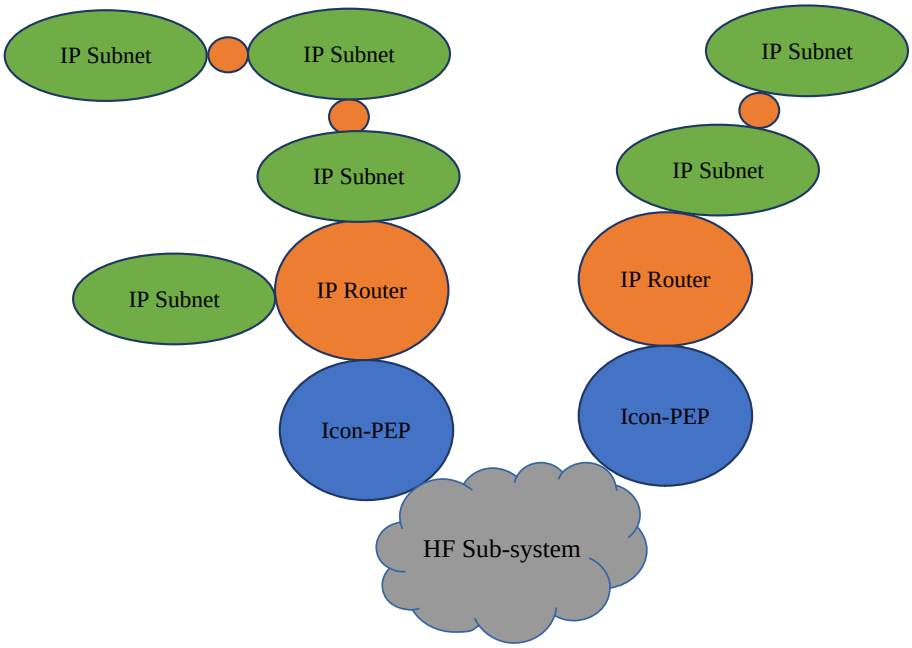
Icon-PEP communicates with one or more IP Routers using Generic Routing Encapsulation (GRE), specified in RFC 2784. GRE provides a simple mechanism to exchange packets between routers over IP, often described as a “GRE Tunnel”.

Icon-PEP terminates the GRE tunnel from a connected IP router. This means that there is no requirement for a peer system to use GRE. Routers commonly monitor GRE tunnel availability using ICMP Ping; Icon-PEP responds to these pings.

GRE is widely supported by Edge routers, and it is anticipated that Icon-PEP will generally connect to the deployed router. For subnets or single hosts that do not deploy a router, or where the deployed router does not support GRE, it is possible to use a software router, such as the one provided on many Linux systems or a product such as pfSense running in a virtual machine.

Alternatively, for stations that only need to handle specific TCP and UDP connections, the "Direct"-mode NAT or listener configuration may be sufficient, which does not require an IP router.

1.1.6 IP Routing and Deployment Model



The IP world comprises multiple IP Subnets, interconnected by IP routers. The routing configuration of an IP Router enables any IP packet to be correctly routed to its destination host via the IP Subnet to which the host is attached.

Icon-PEP core model enables connection of a pair of routers over HF. Icon-PEP simply takes a packet from the local router and sends it over HF to the peer IP router. This is an entirely mechanical function of switching data between STANAG 5066 and the local IP Router.

1.1.7 Icon-PEP Routing

The architecturally simple model described above requires that an IP Router connecting over HF to multiple peer routers requires an instance of Icon-PEP for every peer. This would add significantly to deployment complexity and in practice this approach is rarely needed.

Icon-PEP includes a static IP routing capability. This can be used for systems where all bar one node has a known fixed list of IP subnets. One node can be treated as a default route. This reflects a typical HF configuration, with multiple Mobile Units with a known set of IP Subnets, using a default route to a single shore system, which can have complex IP connectivity.

For performance reasons, it is desirable to avoid use of dynamic routing protocols over HF, so this approach is efficient.

1.1.8 STANAG 5066 Edition 5 support

Where the S'5066 server supports Edition 5 features such as Extended Flow Control, these features may be enabled in Icon-PEP. Icon-5066 will support Edition 5 Extended Flow Control in version 4.0.

When available, Extended Flow Control significantly increases the freshness of the data sent over HF. For protocols that support queue-jumping such as TAK with thinning, this means that the time for a location change to pass over HF will be significantly reduced.

Chapter 2 Initial Setup of Icon-PEP

This chapter describes how to perform initial setup of Icon-PEP.

2.1 Installation

Installation of Icon-PEP is covered in the release notes. Icon-PEP only runs on Linux at present, so may require a virtual machine if the host is not Linux.

2.2 Operation as a Service

Icon-PEP can be operated as a Linux systemd service using the **systemctl** service management capability. This is the recommended way to run Icon-PEP for both initial configuration and deployment. To start Icon-PEP as a service, use the following command:

systemctl start iconpepd

2.3 Operation from Command Line

Icon-PEP can be run from a Linux command prompt. This may be helpful in some cases for remote debugging. See Section 4.12, “Command Line Monitoring” for details of the text user interface (TUI).

First stop the Icon-PEP systemd service if it is running:

systemctl stop iconpepd

Then run Icon-PEP manually using the following command:

(SBINDIR)/**isode.iconpepd -T**

Use **Ctrl-C** to stop Icon-PEP when running on the terminal.

Alternatively Icon-PEP may be run with just terminal logging output using the following command:

(SBINDIR)/**isode.iconpepd -C**

A usage message showing other command-line options can be obtained by running Icon-PEP with **-?**:

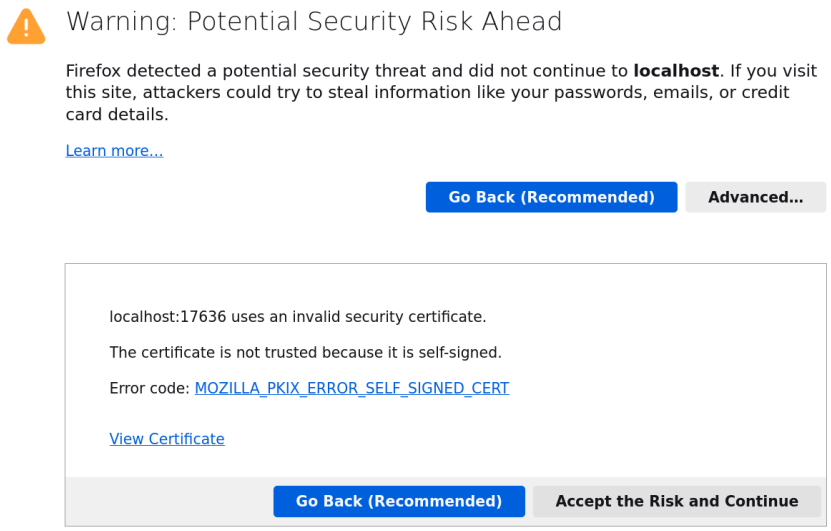
(SBINDIR)/**isode.iconpepd -?**

2.4 Connecting with a browser

By default Icon-PEP listens on all interfaces on port 17636 for HTTPS connections from a web browser. However if TLS has been disabled by the activation, then Icon-PEP will

expect HTTP connections instead of HTTPS. In addition, the listening interface and port may be changed in the configuration.

Note that if activation or a config modification change the protocol, IP address or port that Icon-PEP uses for web access, then it will be necessary to reload the page, or correct the URL to match the new settings, before continuing to interact with the web-UI.



If no private key or public certificate chain have yet been configured, then Icon-PEP will create a temporary self-signed certificate in order to support HTTPS. Your browser will warn you that this is not signed by a known authority. This is expected. To proceed you should select the "advanced" option in the browser and choose to trust the temporary certificate.

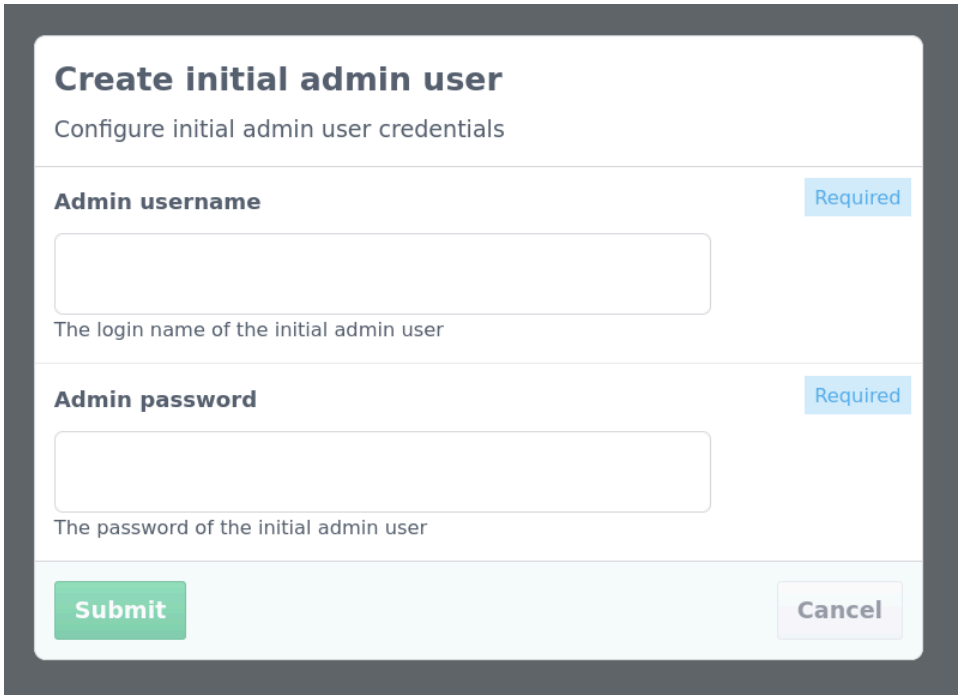
So in its initial default state, you may connect to Icon-PEP by entering `https://localhost:17636/` into your browser, replacing `localhost` with the name or IP-address of the machine hosting Icon-PEP if it is not `localhost`.

2.5 Recovering from a broken configuration

During the initial setup it is possible to lock yourself out of the web-UI by entering incorrect values. The files set up by initial configuration are found in *(ETCDIR)*. If you make a mistake and wish to revert to the unconfigured state, stop Icon-PEP, delete the relevant files under *(ETCDIR)* and then start up Icon-PEP once more.

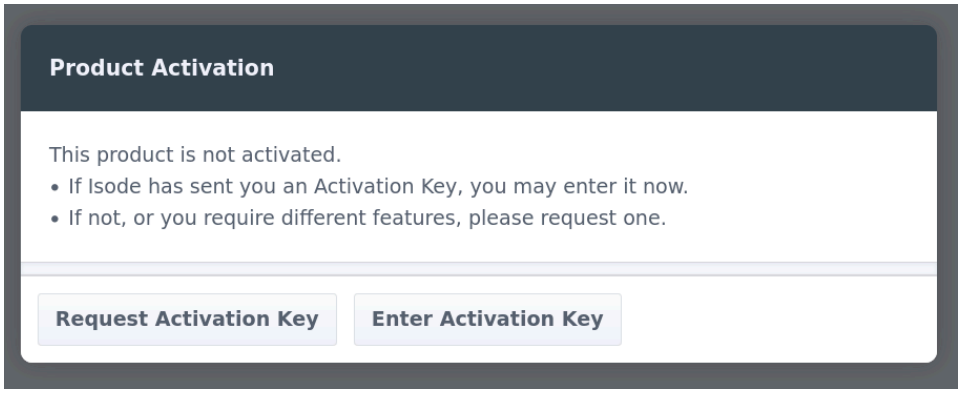
In the specific case of being locked out due to misconfiguring OAuth, you may recover from this situation without deleting the entire config as follows: First stop Icon-PEP, then edit the *(ETCDIR)/pep.json* file and change the `"enable_oauth"` setting from `true` to `false`, save it, and start up Icon-PEP again. Then reload the web-page and you should be able to connect as before and fix the settings.

2.6 Setting an admin password



Choose a username and password. If OAuth is not enabled, then these will be required for all future logins into this web-UI.

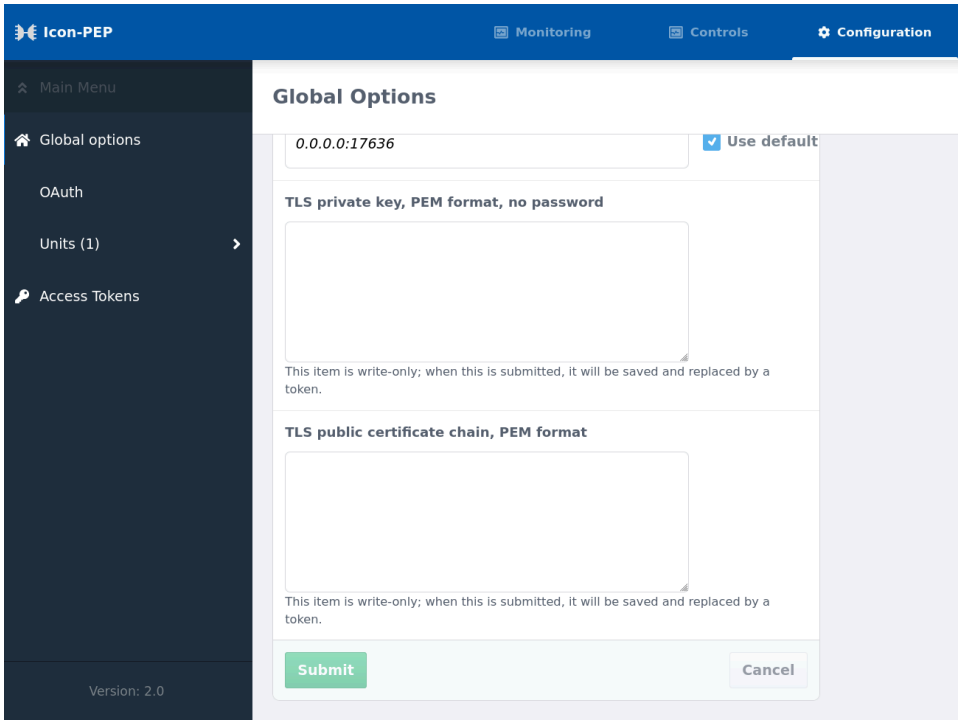
2.7 Product Activation



In this dialog, first choose **Request Activation Key**, and send the activation request to Isode. When the activation key has been provided, choose **Enter Activation Key** and paste in the key. This will activate Icon-PEP. If the activation does not include TLS, then you will have to change the browser's URL from `https:` to `http:` in order to continue using the web-UI.

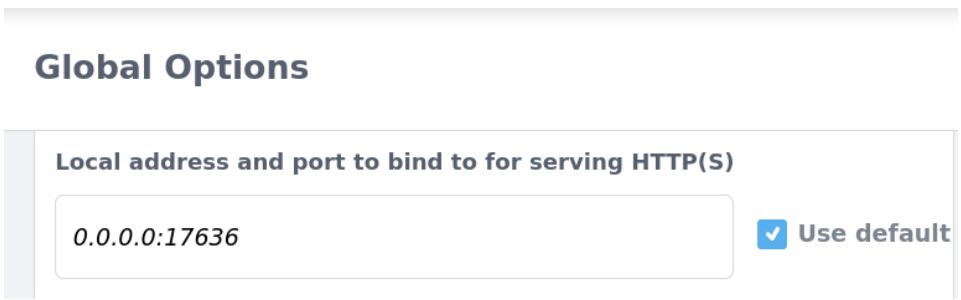
To check the activation status or to reactivate or deactivate Icon-PEP, go to the top-right menu and select **About Icon-PEP**. **Update Key** and **Deactivate** buttons provide the necessary functions.

2.8 Configuring a TLS key and certificate



By default, Icon-PEP will operate using a temporary self-signed certificate which changes each time Icon-PEP restarts its web service. If you wish to provide a proper private key and public certificate chain so that HTTPS can operate without a warning in the browser, then go to the top level of the **Configuration** under **Global Options** and paste in both the private key and the public certificate chain in PEM format, submit and then reload the page. Note that the private key must not be protected with a passphrase. The certificates are replaced by tokens in the UI, and the sensitive content is saved to disk in an encrypted and machine-locked format. If there are any problems with the PEM files then errors or warnings will be visible in the log file, found under *(LOGDIR)*.

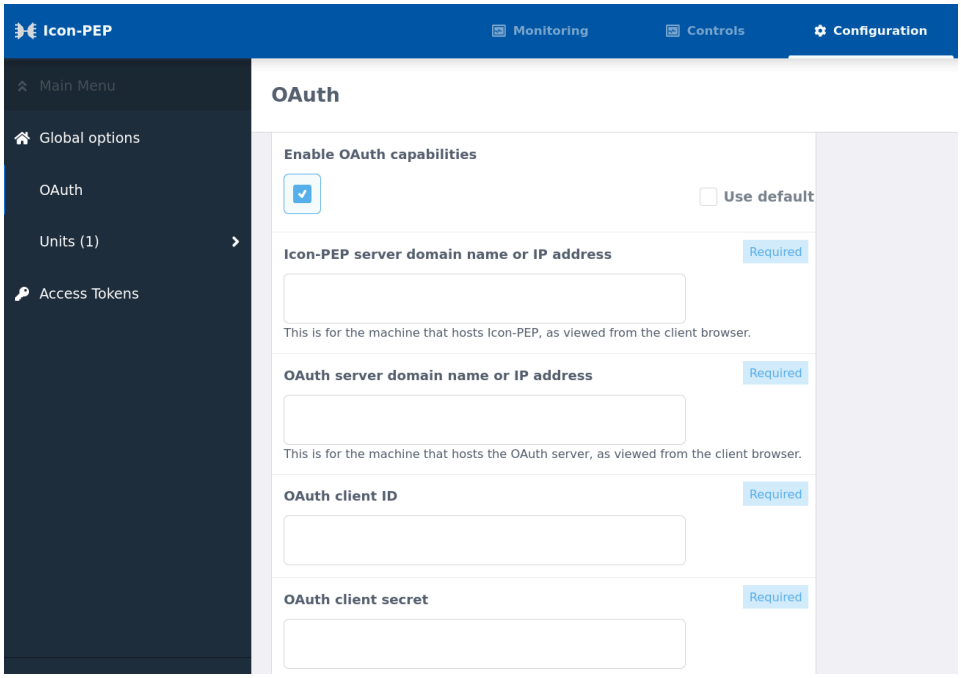
2.9 Changing HTTP(S) listening port



If you wish to have Icon-PEP listen for HTTP(S) connections on a different port, or to listen only on one interface, then go to **Configuration**, **Global Options** and visit the **Local address ...** field. The default IP of `0.0.0.0` listens on all interfaces. To listen on just one interface, use the IP address that the host machine has on that interface. After

submitting the new IP and port number, change the URL in the browser to match the same settings and reload the page.

2.10 OAuth configuration



OAuth centralizes login credential handling, meaning that no login-related secrets have to be kept by Icon-PEP. If you wish to use OAuth, fill in the configuration in the OAuth tab with details provided by the M-Vault OAuth server. Cobalt may be used as the interface for provisioning. After clicking **Submit** to apply an OAuth configuration, Icon-PEP will restart its web service, so it is necessary to reload the page in order to continue using the UI.

Chapter 3 Configuring Icon-PEP

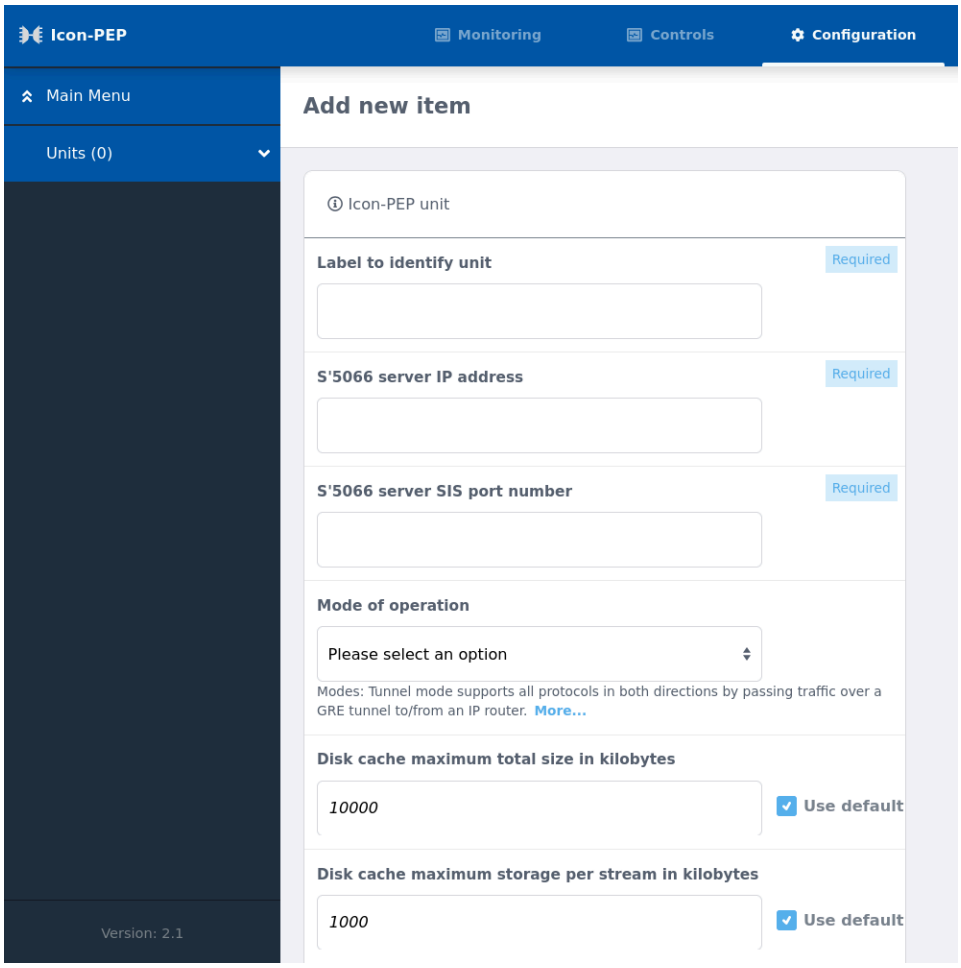
This chapter describes how to configure Icon-PEP.

Icon-PEP should be configured through the **Configuration** tab of the web-UI. The configuration is saved to a single JSON file stored in *(ETCDIR)/pep.json*

3.1 Units

Each "Unit" is like a separate independent running instance of Icon-PEP. Most sites will only require a single Unit. However a more complicated shore station using several S'5066 server nodes to access HF will need to set up a different Unit for each S'5066 server.

To add a Unit, go to the **Units** entry in the left-hand pane of the **Configuration** tab, and select **Add...**. The top-level configuration for a Unit can be made at Unit creation, or changed later if required. It is as follows:



3.1.1 Common parameters

These basic parameters may be requested by the top-level configuration form, depending on the mode selected:

S'5066 server IP address and **S'5066 server SIS port number** give the socket address for connecting to the S'5066 server.

Node address gives the S'5066 node address represented by the S'5066 server. Node addresses are expressed as a dotted-quad, like an IPv4 address. Where the first number of the dotted-quad is below 16, leading zero half-bytes are stripped off and the remaining half-bytes are taken to be the S'5066 node address. So the shortest S'5066 node address would be in the range 0.0.0.0 to 0.0.0.15, which corresponds to a single half-byte in S'5066 addressing. If the first number is 16 or over, that represents either a group address or an unnormalized node address, but these are unlikely to be required in this application.

Disk cache parameters control the disk caching for TCP streams. Since HF is a much slower medium than ethernet or the internet, servers may time out the connection when they detect that the data flow is too slow. To solve this problem, data is read at full speed from the TCP connection and temporarily stored on disk. It is then fed to the 5066 server at the speed that it can accept it. The disk space is freed up when the data has been sent or if the connection is closed or aborted. These two parameters limit how much data to cache on disk.

Mode of operation selects the overall mode of operation, as follows:

1. **Tunnel** is for connecting to the LAN over GRE tunnels via one or more IP routers. This mode supports all IP traffic, even if a proxy is not available for that traffic. It is the most flexible mode and maintains IP addressing end-to-end, but requires a software or hardware router.
2. **Direct** works without an IP router, and allows a subset of protocols to be exchanged with the local LAN directly, through either NAT-style connections out to the LAN, or using listeners for connections in from the LAN. It is more limited but requires less setup.
3. **Simple** is a simplified "Direct" configuration suitable for a NAT shore station that only needs to act as the responder for connections initiated by other S'5066 nodes.

Note that all of these three modes use the same underlying protocols over HF (e.g. IP Client for IP packets, or HF-PEP over SLEP for TCP streams). So it is fine to have one node using "Tunnel" mode, and another node using "Direct" or "Simple"; everything will still work.

3.1.2 Tunnel: Using one or more IP routers

This connects to the local network via a GRE tunnel to an IP router. There is one required field to fill in: **Local IP address to bind GRE listener to**. This should be the host machine's IP address on the network interface that will be used to communicate with the GRE router. This is the IP address that the router will be configured to send GRE traffic to.

"IP Client" (S'5066 Annex U) is automatically enabled, both for incoming and outgoing IP traffic. This relays IP packets directly over HF without any performance enhancement. There is one additional option:

3.1.2.1 Configure TUN-based TCP PEP

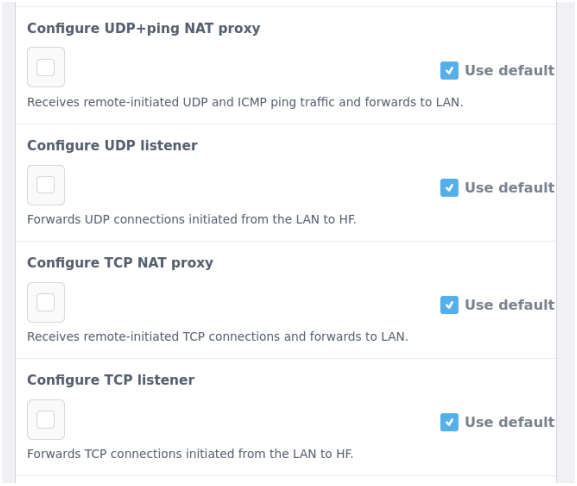
This enables performance enhancement for any protocol carried over TCP (including HTTP and HTTPS), both for incoming and outgoing connections. TCP connections are terminated by Icon-PEP and the raw stream data is passed over HF using an HF-optimised protocol. This takes only the TCP traffic, leaving all other traffic for IP Client to handle.

This has one required field to fill in: **TUN device IPv4 subnet**. The suggested value of `172.30.0.0/16` is fine unless that will clash with one of the subnets used on the local network. If IPv6 is required, this can be configured later.

("TUN" refers to one half of the TUN/TAP driver provided by Linux which allows transferring IP packets into and out of the Linux network stack, which is the mechanism by which TCP termination is implemented.)

3.1.3 Direct: Using NAT and/or listeners

There are the following options:



3.1.3.1 Configure UDP+ping NAT proxy

This may be used to forward IP Client UDP exchanges and ICMP pings incoming from HF to the local network. All other IP Client traffic is dropped. The requests are made directly from Icon-PEP. This gives the appearance of NAT: the packets are addressed from the local machine rather than the remote IP address. Note that this does not permit UDP or ping requests to be initiated from the local network and sent to remote networks, but responses to the incoming UDP packets or ping requests are captured and returned to the initiator over HF.

3.1.3.2 Configure UDP listener

This may be used to forward UDP packet exchanges initiated from the LAN over HF to a remote node. A listener port is opened, and the UDP clients should connect directly to this port on the local machine. The packets are forwarded over HF using the IP Client protocol, and any responses received back over HF are passed back to the UDP initiator.

This also supports bidirectional UDP multicast, either point-to-point over S'5066, or to a S'5066 broadcast node address. Where multicast is required, a UDP listener should be configured at both ends.

3.1.3.3 Configure TCP NAT proxy

This may be used to forward TCP PEP connections incoming from HF directly to the local network. The TCP connection is made directly from Icon-PEP, giving the appearance of NAT. Data is streamed both ways.

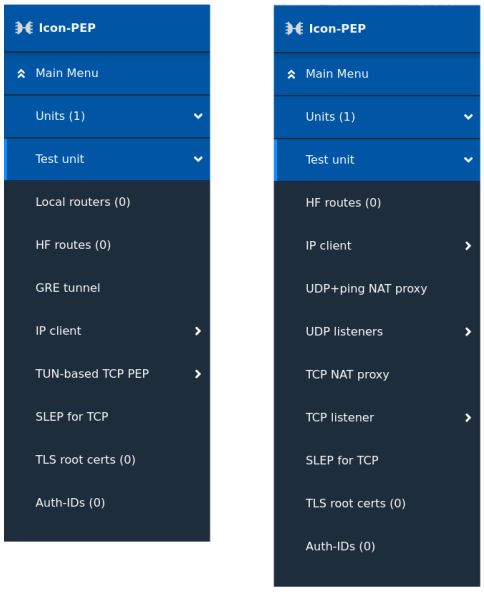
3.1.3.4 Configure TCP listener

This may be used to forward TCP connections from the LAN over HF to a remote node. A listener port is opened, and the TCP clients should connect directly to this port on the local machine. The stream is forwarded over HF using the same HF-PEP protocol as used by TUN PEP or TCP NAT proxy, so the remote node can be in either mode.

3.1.4 Simple: Simplified NAT shore proxy

This enables a simplified configuration for a shore station that does not permit outgoing connections from the shore to the mobile units. It only accepts incoming TCP, UDP and ICMP ping requests from mobile units, which it proxies out to the local network in the style of NAT. This configuration does not require HF routing tables to be set up, and will accept traffic from any valid S'5066 node or IP address.

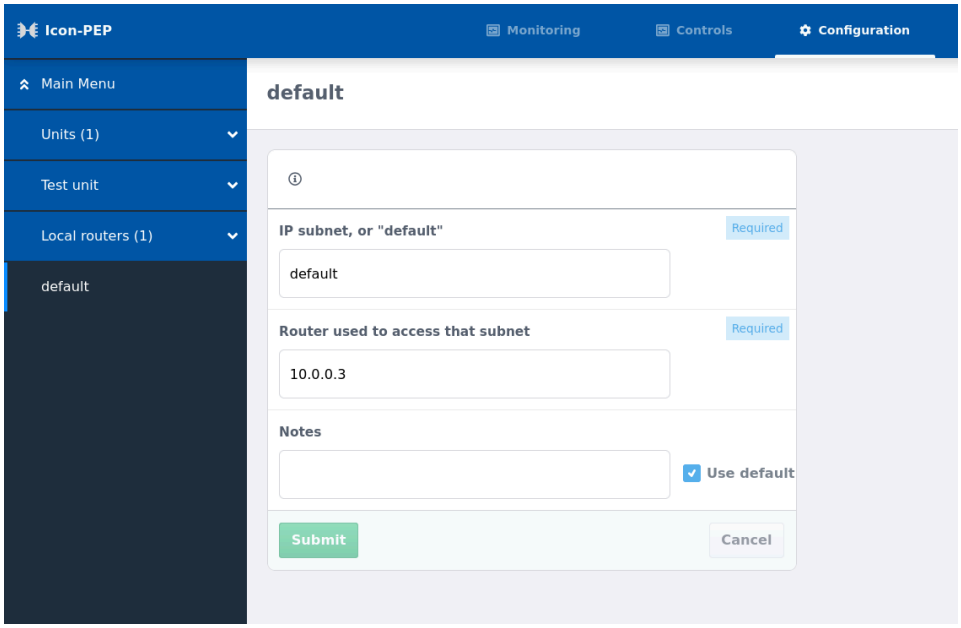
3.2 Configuration subsections



Only the subsections relevant to the current configuration will appear in the left bar. Most of the configuration can be left with the default values, and is not described here. Certain subsections such as SLEP have a lot of tunable parameters. In general it is safest to leave these parameters as they are unless a problem has arisen and been thoroughly analysed and Isode support agrees that modifying those parameters will help resolve it.

The following subsections however are required for correct operation, or may be necessary in certain deployments.

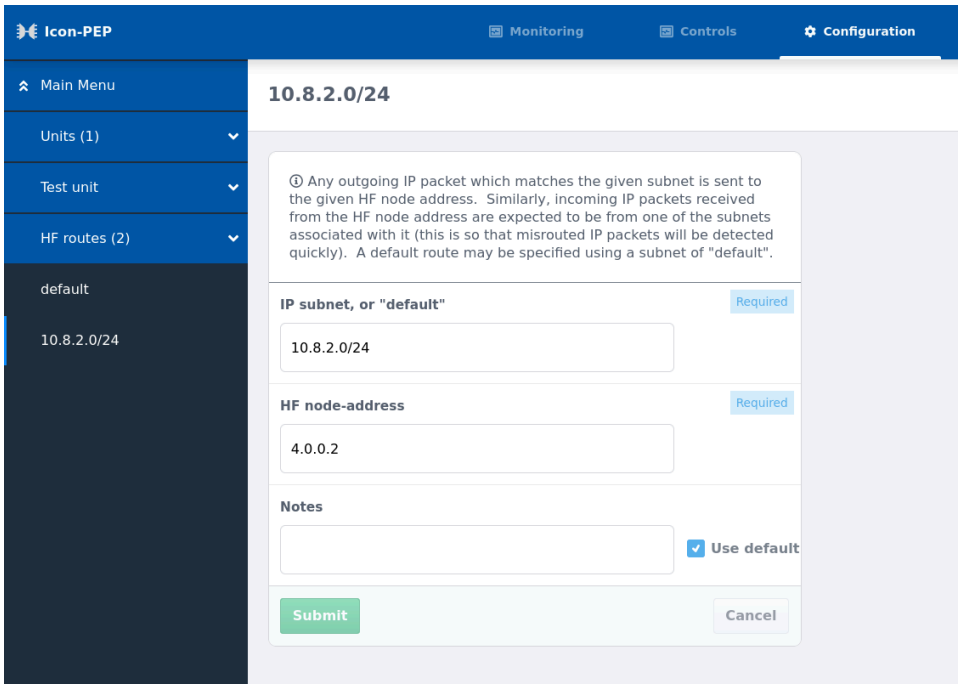
3.2.1 Local routers configuration



Routing to the local network over GRE tunnels is done via this table. Typically there will be just one entry, with `default` for the IP subnet, and the IP address of the GRE router for the **Router**. However if the local network configuration is more complicated,

it is possible set up several entries so that traffic for different IPv4 or IPv6 subnets is sent to different GRE routers accordingly.

3.2.2 HF routes configuration

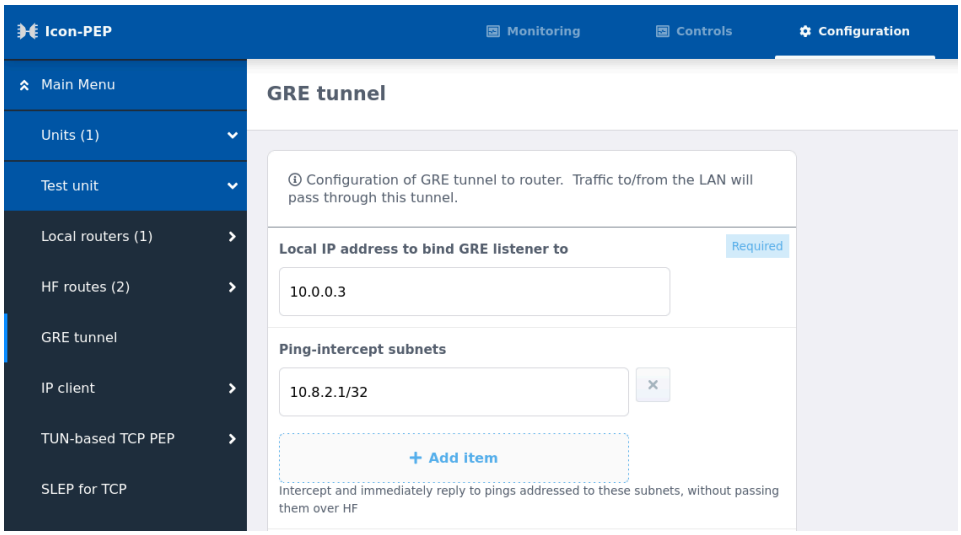


Routing over HF is determined by a fixed mapping table between IP subnets and S'5066 node addresses. When an IP address is looked up, if it matches ones of the subnets, then the corresponding node address is used to send out the traffic. If it matches none of the subnets but a "default" entry is present then the corresponding default node address is used. This "default" route would typically map to the node-address of a shore station that connects to the wider shore network. However if no route matches then the packet is regarded as unroutable and will be dropped, and a warning will appear in the logs.

Subnets are specified in standard IPv4 or IPv6 network address/netmask format, such as `10.0.2.0/24`. Each subnet is associated with a node identified by its S'5066 address. Multiple subnets may be associated with a node.

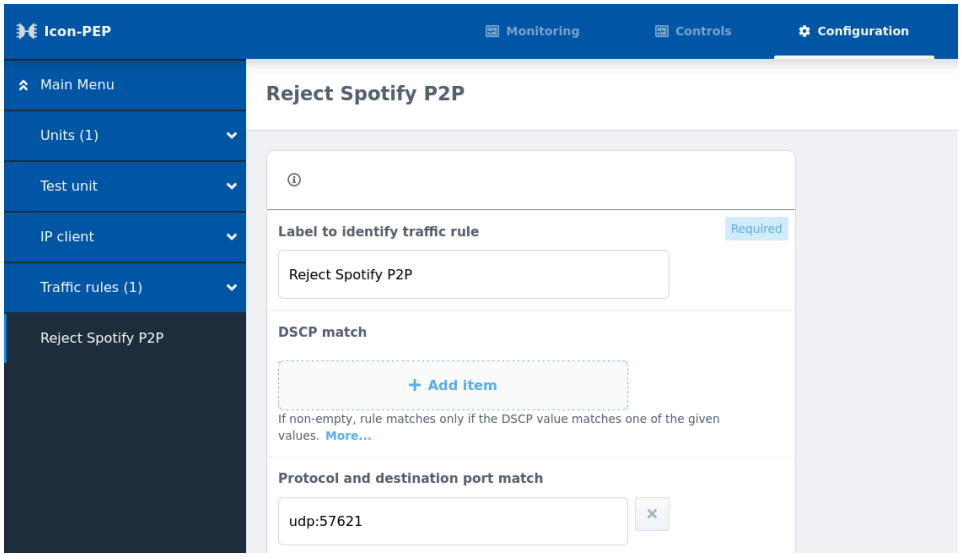
The routing table must include one or more entries for the local node as well as for all remote nodes. One model of deployment of routing tables is to copy the same routing table to all nodes, which avoids the possibility of packets being delivered somewhere where the return packets will be unroutable. The routing table is also used to check incoming and outgoing traffic. If the IP address on traffic doesn't correspond to the expected node address for that IP address, the traffic is rejected with a warning in the logs, since any return traffic along the same path would be unroutable.

3.2.3 GRE tunnel configuration



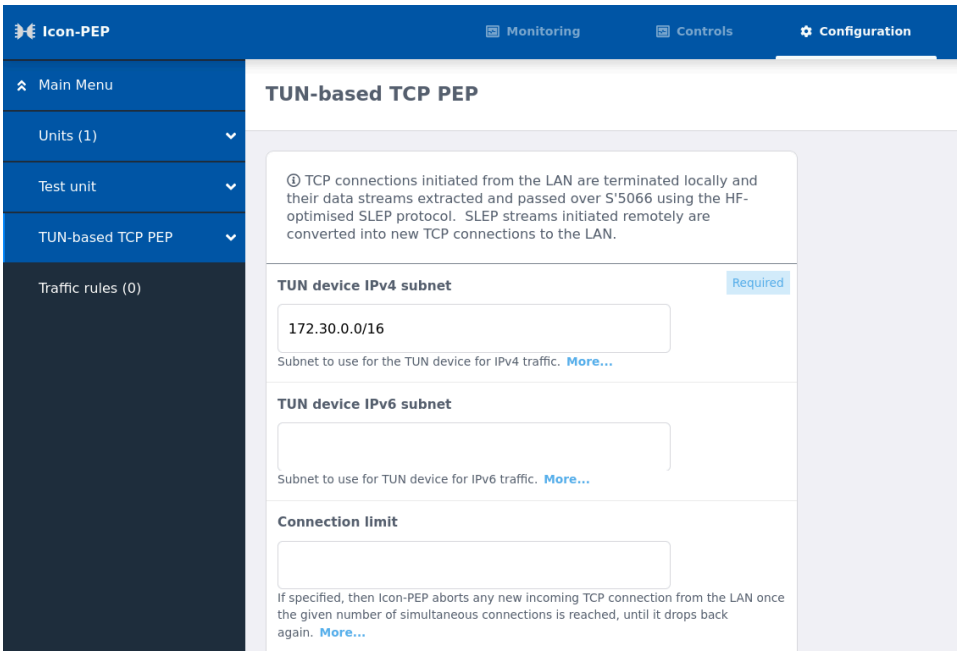
Some routers send pings over the GRE tunnel to check that the other end is alive, using “internal” addresses. It is inefficient to pass these over HF, so these should be intercepted and replied to locally. Set this up using the **Ping-intercept subnets** field. These subnets will typically cover a single IP address, so the subnet mask will be `/32`.

3.2.4 IP Client Traffic Rules configuration



Traffic rules allow blocking and selection of priority and other settings for outgoing IP Client packets. The first rule that matches is used. If no rule matches then the packet is allowed to pass with default settings. Each rule may match on any of DSCP, protocol and destination port, source IP subnet and/or destination IP subnet. On matching, the packet is dropped if **Drop packet** is enabled. Otherwise the provided SIS priority, ARQ and in-order settings are applied.

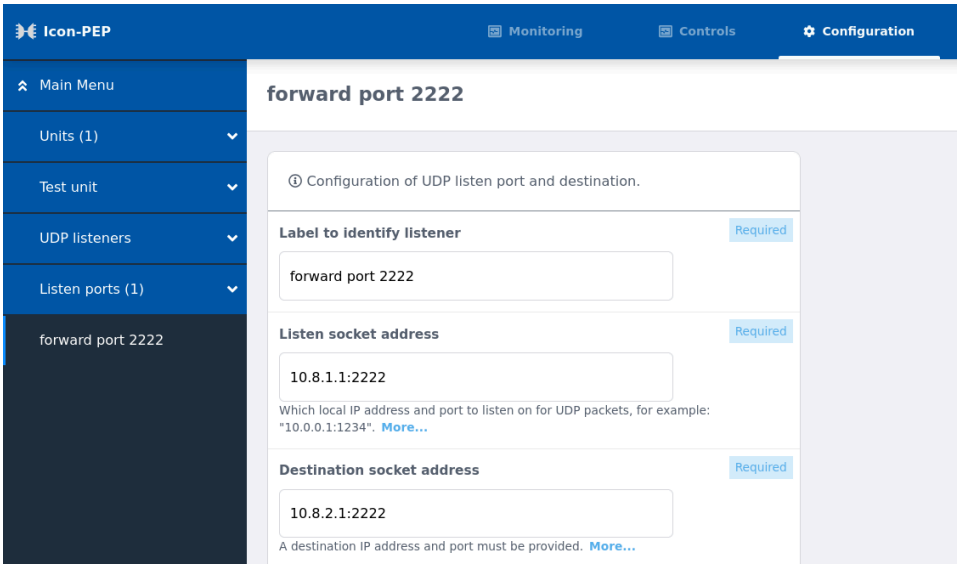
3.2.5 TUN-based TCP PEP



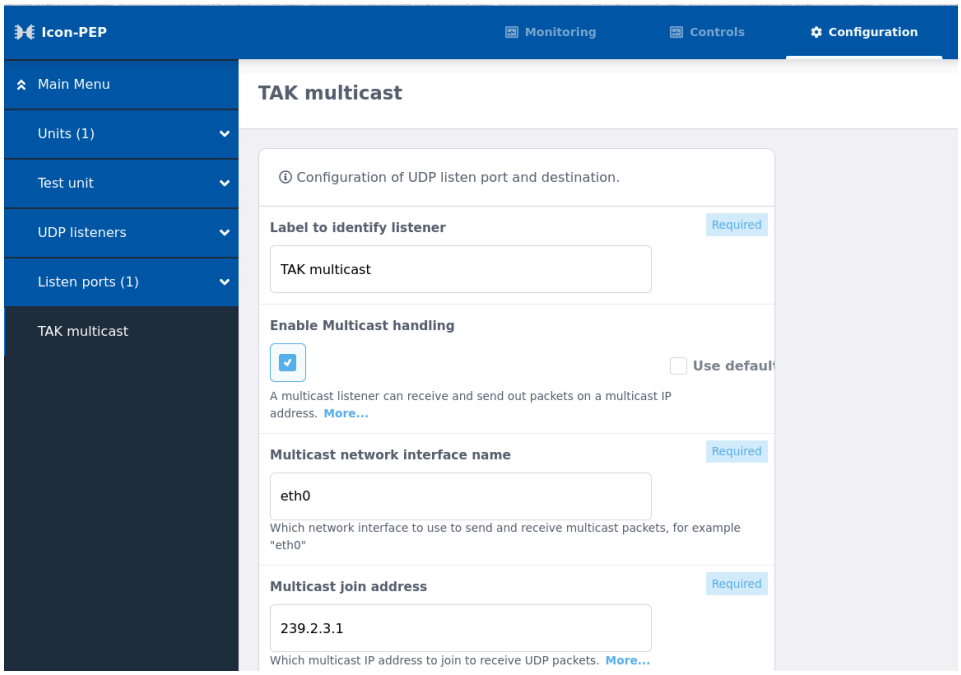
To support IPv6 TCP connections, provide a value for **TUN device IPv6 subnet**. Also you may wish to specify a **Connection limit** to limit the number of simultaneous outgoing TCP connections.

The **Traffic Rules** section allows blocking of outgoing TCP connections and selection of priority, compression, Auth-ID mapping and TLS proxy configuration. The first rule that matches is used. If no rule matches then the connection is allowed using default settings. Rules may match on destination port, source IP subnet and/or destination IP subnet.

3.2.6 UDP listeners

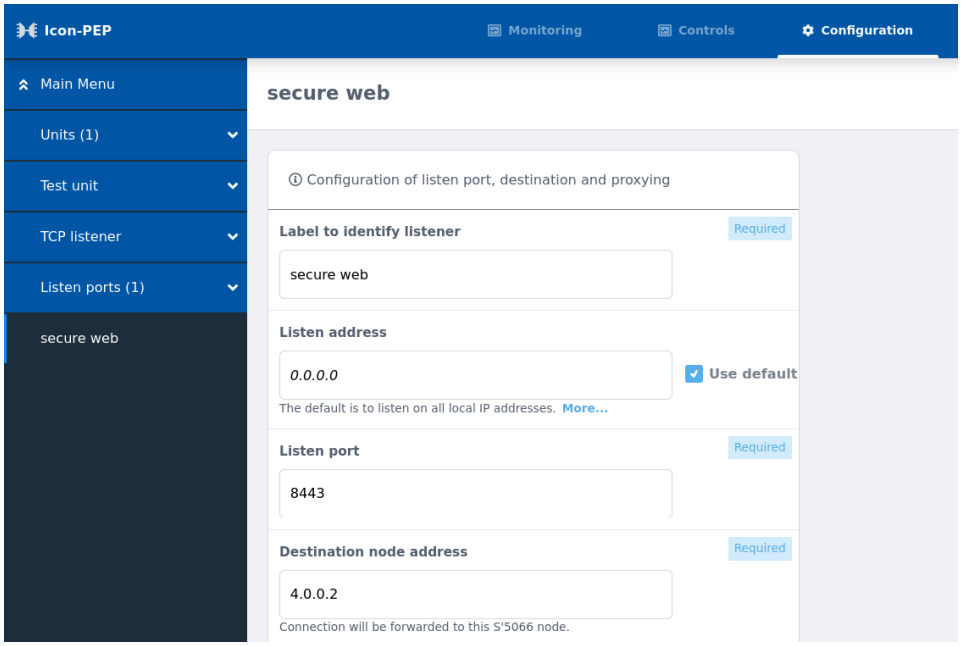


Each individual UDP listener port must be configured here, along with the destination port and address to which the UDP packets should be directed. The destination S'5066 node is determined by looking up the destination IP address in the HF routing table.



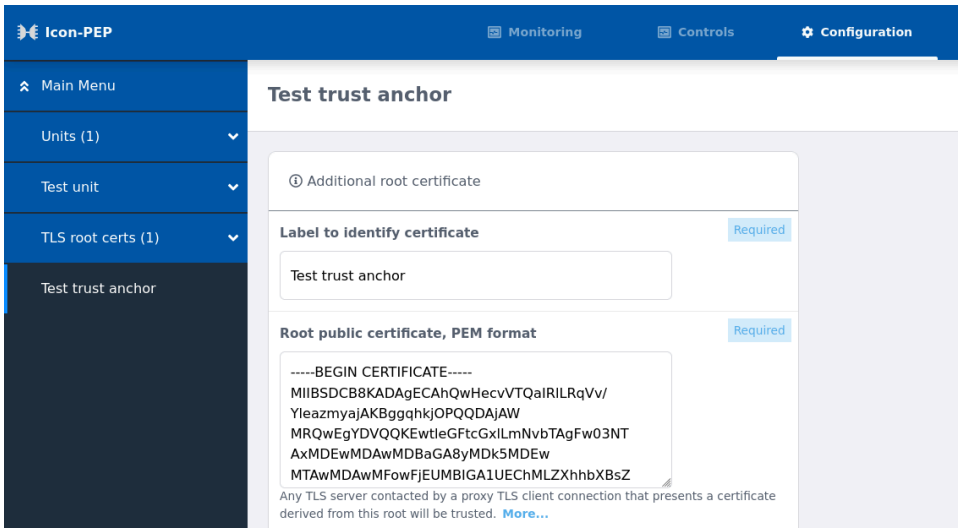
For multicast UDP, specify network interface, multicast IP address and port, and S'5066 destination node address. Multicast packet handling is bidirectional across HF. A similarly-configured UDP listener must be active at the remote node address(es) to pass on the multicast packets. S'5066 transmission may be point-to-point, or else broadcast using a S'5066 group address. ARQ is possible only for point-to-point links.

3.2.7 TCP listeners



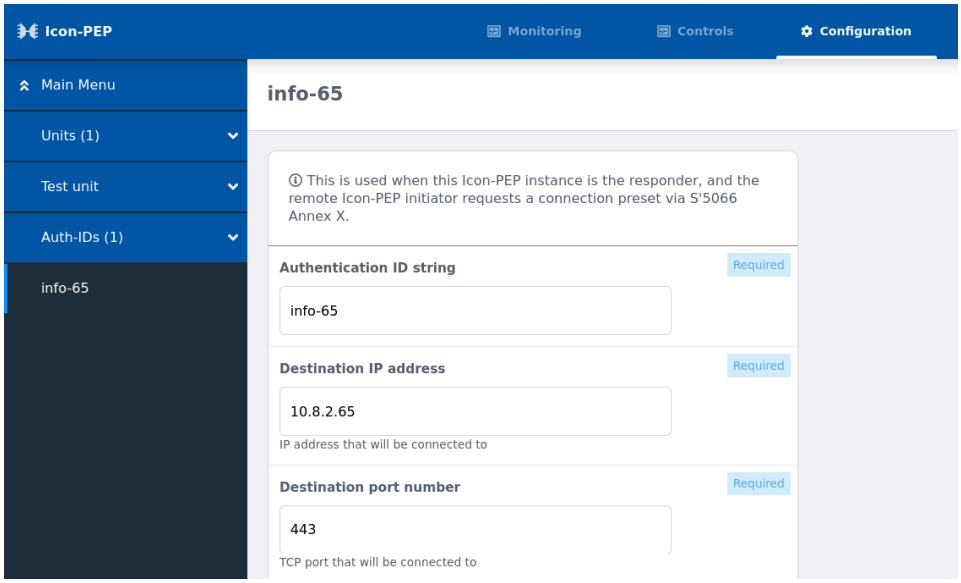
The **Listen ports** rules not only specify which ports to listen on, but also how to forward the data to the remote Icon-PEP instance, and which proxies to apply to this connection. This includes support for TLS (with optional client authentication) and HTTP/2.

3.2.8 TLS root certificates



These should be configured on the responder side, and are used to validate TLS client connections made by Icon-PEP to TLS servers. Icon-PEP makes TLS client connections when the TLS proxy is enabled on the remote, or when a local Auth-ID configuration specifies a TLS connection.

3.2.9 Auth-IDs



These are configured on the responder side to specify how the TCP connection out to the LAN should be made when the remote initiator requests a connection over HF with that Auth-ID. TLS can be enabled, along with SNI and ALPN values and client authentication if required. An HTTP/2 proxy is available to help with server-to-server TAK connections.

3.3 Proxies

Configuration of proxies is split between initiator and responder sides, so this section explains how they work together.

3.3.1 TCP proxy

When this proxy is in operation, the TCP protocol is terminated at Icon-PEP and the stream data contained within TCP is forwarded over HF using the HF-optimised protocols in S'5066 Annex X and Annex W. Compared to passing TCP/IP packets directly over IP Client, this uses fewer bytes and saves round-trips. It also enables the use of compression.

The protocol is used in the same way for both "Direct" and "Tunnel" modes of operation. For "Tunnel" mode it is configured by selecting **Configure TUN-based TCP PEP** at the top level of the Unit configuration. For "Direct" mode it is configured by selecting **Configure TCP NAT proxy** or **Configure TCP listener**.

3.3.2 TLS proxy

TLS is the underlying protocol providing cryptographic security for connections such as HTTPS, LDAPS and XMPP. If TLS is used as a layer in the protocol stack then this proxy can provide savings by avoiding passing the initial cryptographic exchange over HF, saving both bytes and round-trips. Unwrapping this protocol layer also enables compression to be applied for additional savings. This proxy is layered on top of the TCP proxy, i.e. it can only operate if the TCP stream is being proxied.

This proxy works by acting as a TLS server on the initiator node, and as a TLS client on the responder node. This means that the configuration is split between the two nodes.

3.3.2.1 Initiator side configuration

On the initiator side, the configuration is made in the following places: for "Tunnel" mode in **Traffic rules** found under **TUN-based TCP PEP**, or for "Direct" mode in **Listen ports** under **TCP listeners**. The settings are specific to each traffic rule or listener port. A different TLS server is started for each rule or port, so the handling can be different for each one.

3.3.2.2 Responder side configuration

On the responder side, there are two sections that may need to be configured within the Unit: **TLS root certs** must be configured if any of the TLS connections are going to be made to servers whose certificates were not issued by an authority recognised by the system-installed certificate list. **Auth-IDs** must be configured if the initiating Icon-PEP is configured to use Auth-IDs for any of its connections.

3.3.2.3 ALPN handling

ALPN (Application-Layer Protocol Negotiation) is an optional TLS mechanism used to negotiate the protocol carried within TLS. The TLS server informs the TLS client which protocols it supports, and the TLS client chooses one. For example the identifier "h2" refers to the HTTP/2 protocol. As the TLS negotiation is not passed over HF, this has implications for ALPN handling.

Icon-PEP supports ALPN negotiation, but the list of protocols supported by the final end-point server must be configured on the initiator-side Icon-PEP. The initiating TLS client chooses one of those ALPN protocols and then Icon-PEP passes that choice over HF to the remote Icon-PEP, which uses it when negotiating a TLS connection with the final end-point.

There is also the option to skip the ALPN negotiation with the initiator and override the ALPN identifier with a fixed value.

3.3.2.4 SNI handling

TLS allows the TLS client to provide a SNI (Server Name Indication). This is the domain-name that the TLS client is intending to connect to. For example, this is important when a single IP address is used to serve several websites. Icon-PEP passes

through the SNI over HF, and on the responder side uses it when making the TLS connection to the final end-point.

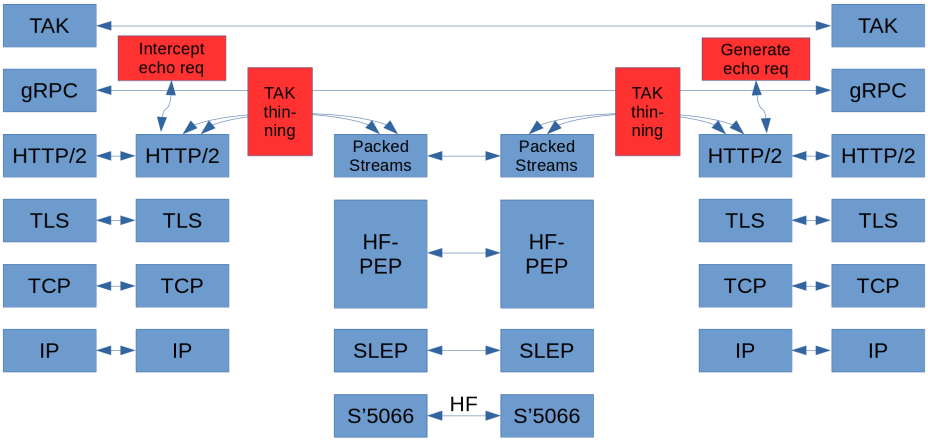
3.3.2.5 Auth-IDs and TLS client authentication

Authentication IDs (Auth-IDs) are a mechanism within S'5066 Annex X that replaces the connection details passed over HF (addresses, ports, TLS, ALPN, SNI) with a single identifier. The responding Icon-PEP looks up this identifier in a table to find the connection details it should use to forward the connection. It may be used for both TLS and non-TLS connections. However it is particularly useful for TLS connections where client authentication is required. The Auth-ID can be used to represent not just the connection details but also the identity of the client. The responding Icon-PEP can be configured to use a suitable client certificate so that the same logical identity is presented to the final end-point.

On the initiator side there are two ways to configure this. Firstly the **Map directly to Auth-ID** option may be used when client authentication is not required. A single Auth-ID must be provided which will be used to represent the connection over HF.

Alternatively, enabling **Require TLS client authentication** makes two new subsections appear where the client-auth trust roots and identities may be configured. The trust roots control which client certificates will be accepted. The identity list maps from SAN addresses within the client certificate to the Auth-IDs that are passed over HF.

3.3.3 HTTP/2 and TAK proxy



HTTP/2 is a web protocol that allows multiple web requests and bidirectional streams of data to be sent down a single TCP connection. It is always used with TLS, so this proxy stacks on top of the TLS and TCP proxies.

The purpose of this proxy is to unpack an HTTP/2 stream to allow the individual requests and streams to be examined and modified, before repacking all the streams into an HF-optimized protocol to be forwarded over HF. At the moment the only modifications supported are TAK (Team Awareness Kit) thinning, answering TAK protocol health-checks on the initiator side, and generating TAK health-checks on the responder side. This removes the overhead of TAK health-checks from the HF connection and thins the TAK traffic to match the HF bandwidth.

To enable the HTTP/2 proxy on the initiator side, go to the traffic rule or listen port config and select **Enable TLS proxy** and then **Enable HTTP/2 proxy**. The option **Answer TAK health-check requests sent from the LAN** then appears, which should be also enabled. The field **Delay before sending back a TAK health-check response, in seconds** should be set longer than the typical round-trip-time for messages over HF to avoid the TAK server breaking the connection. For example, a value of 220 worked in Isode testing.

To apply the HTTP/2 proxy on the responder side, an Auth-ID must be used for the connection. Within the Auth-ID mapping, enable **Use TLS for connection**, then **Enable HTTP/2 proxy**, then **Generate TAK health-check requests to the LAN**.

Packet thinning consists of discarding and reordering TAK messages to get the stream of events to fit the HF bandwidth available. The aim is graceful degradation of the service when the messages exceed the bandwidth, e.g. updates for all objects at longer intervals instead of updates that are heavily delayed or completely absent for long periods. These are the options:

1. **None: No thinning: Pass packets in order**: This passes packets in arrival order and periodically trims the queue to about a transmission's-worth, discarding the newest items.
2. **NewestFirst: Send most recent packets first, letting older packets expire**: The aim is to let newer information skip older information, and trim older information when the queue gets too long. This will work if all location updates come through at about the same interval. However data which is sent only occasionally may get lost.
3. **Update: Packets with new info queue-jump to update older packets of the same type**: The aim is to maintain an ordered queue, so that both frequent and infrequent updates get equal treatment, but to always send the most up-to-date information available. So if there are several location updates for the same object in the queue, they are merged and the newest information is sent when any one of them leaves the queue. This deduplication saves a lot of bandwidth without having to trim the queue (losing messages) so long as the number of tracked objects is within the capacity of the HF channel.

3.3.4 Multicast UDP TAK proxy

A UDP Listener must be set up at each node with multicast enabled and the multicast address/port and node addresses filled in. TAK monitoring and thinning is not yet available for UDP TAK traffic in Icon-PEP version 3.0.

There is a command-line tool provided to help debug multicast setups: *(SBINDIR)/multicast_tester*. Run as `multicast_tester -h` to see usage information. The tool always listens and always displays received multicast packets, and can optionally also send out test multicast packets. It can be run on the same machine as Icon-PEP, or on any other machine on the network. It is suggested that you check multicast connectivity over the local network and over HF using this tool if there are any difficulties getting a multicast application to work.

Chapter 4 Operating and Monitoring Icon-PEP

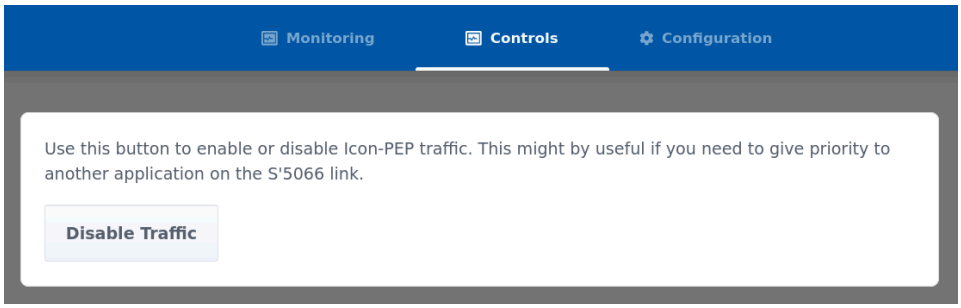
This chapter describes how to operate and monitor Icon-PEP.

4.1 Echo server

For testing, when TCP proxying is enabled on both the initiating and responding Icon-PEP servers, the responding server provides a built-in TCP echo server on port 7 of its remote IP address. When a TCP client such as **telnet** connects to port 7 of an IP address which is handled by another node, the traffic goes out over HF to that node, and then Icon-PEP on that node echoes the data back again over the proxied TCP stream, back over HF. This provides a test that transmission over HF to that node is operating correctly, without having to set up any services on that remote network to test against. For example:

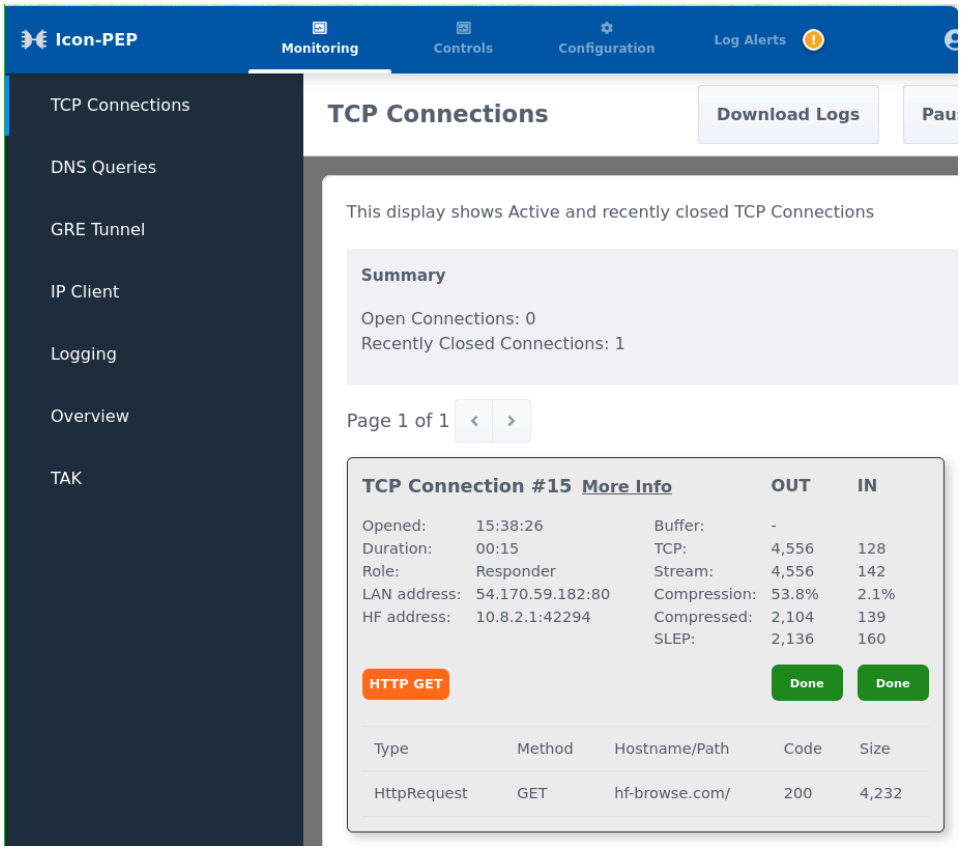
telnet 10.0.2.1 7

4.2 Enable/disable control



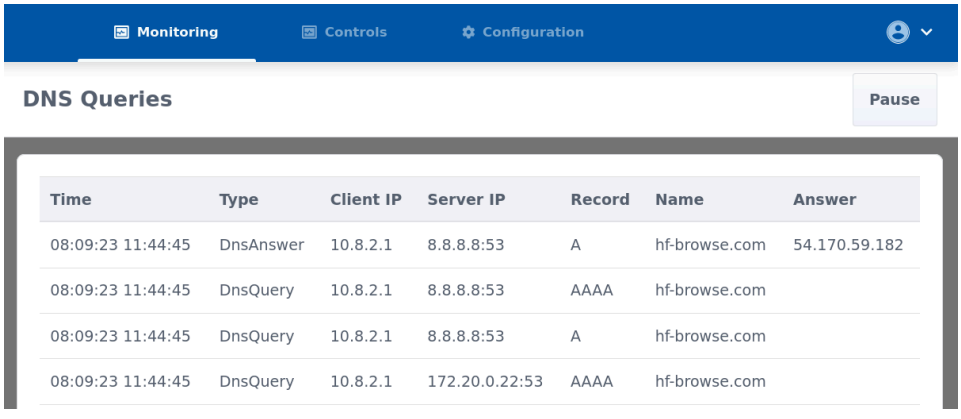
Under **Controls** there is a single control that may be used for enabling or disabling traffic. This may be useful when the S'5066 connection needs to be kept free for higher-priority traffic from another application. This stops Icon-PEP from sending new data. Any outstanding SLEP connections are shut down nicely, however, so there may still be a little bit of traffic as these connections terminate.

4.3 TCP monitoring



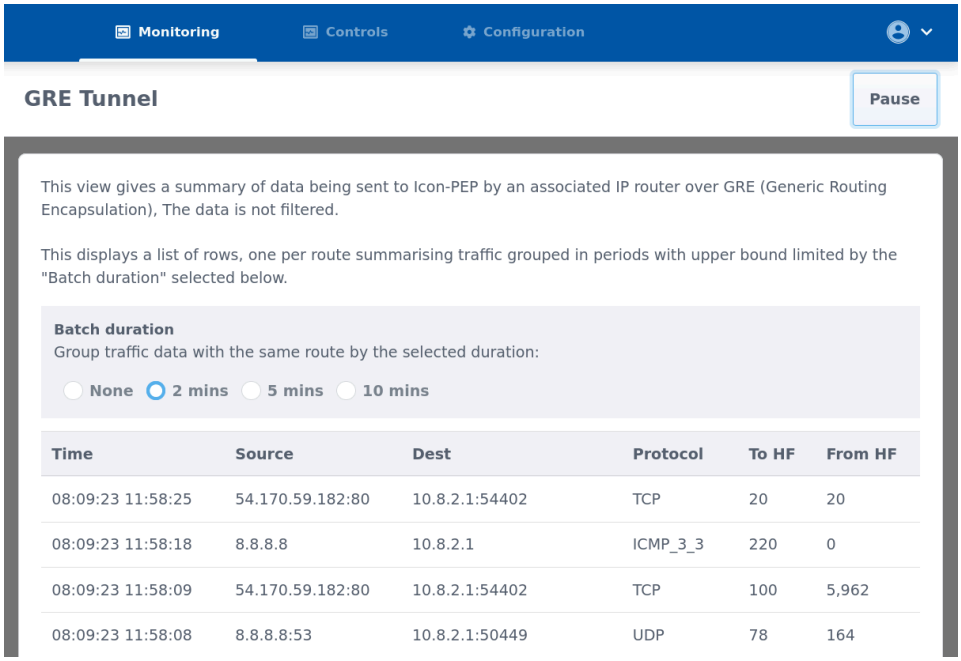
The TCP monitoring page shows the status of all TCP connections passing through when in the TCP-proxy modes of operation (TCP PEP, NAT TCP or TCP listener). For HTTP connections, details of the request or response are also available. Tooltips give more detail on the input and output stream states. **Pause** allows the display to be paused. When there are a large number of TCP connections active or recently closed, you may need to use the page arrows to find the connection that you are interested in. The "More info" link shows a detail view, including all the logging associated with this connection.

4.4 DNS monitoring



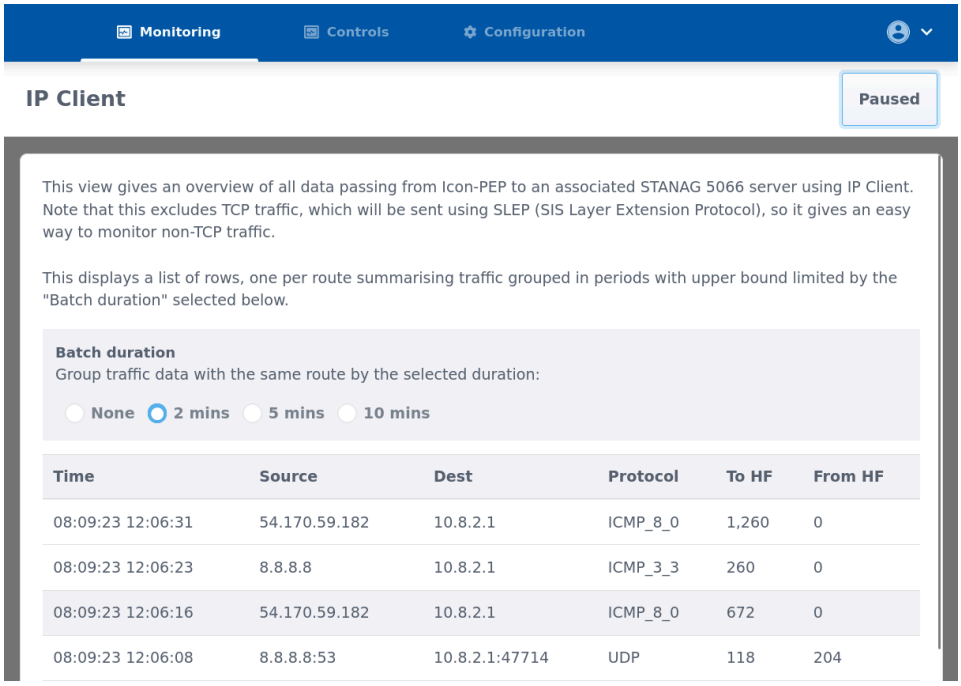
The DNS monitoring page shows the contents of all DNS requests and responses passing through the server. DNS usually uses UDP, but for larger requests it may also use TCP. TCP will be monitored only if the server is in a TCP-proxy mode of operation. The **Pause** button allows the display to be paused.

4.5 GRE tunnel monitoring



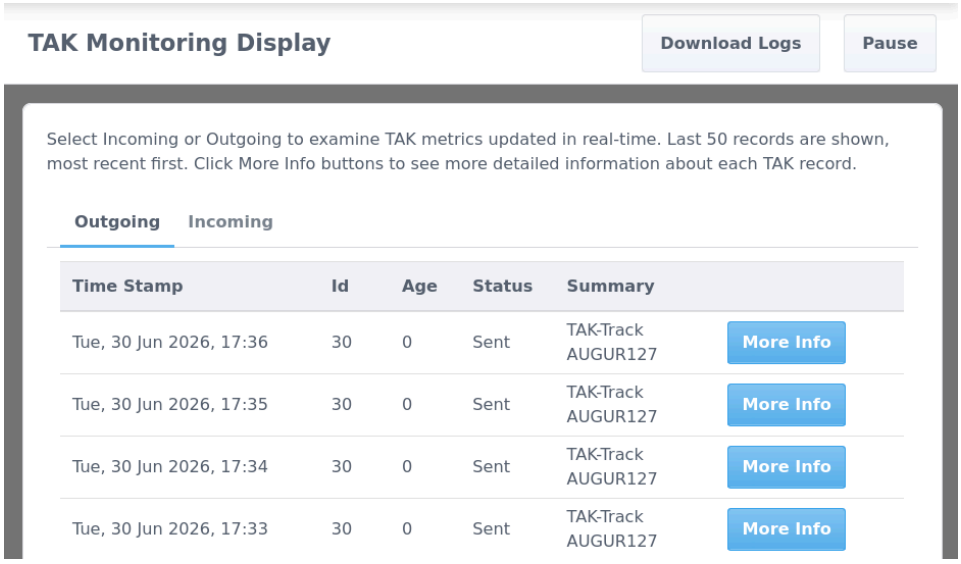
This shows a summary of all traffic passing over the GRE tunnel. This may include traffic which is subsequently filtered out by traffic rules. The **Pause** button allows the display to be paused.

4.6 IP Client monitoring



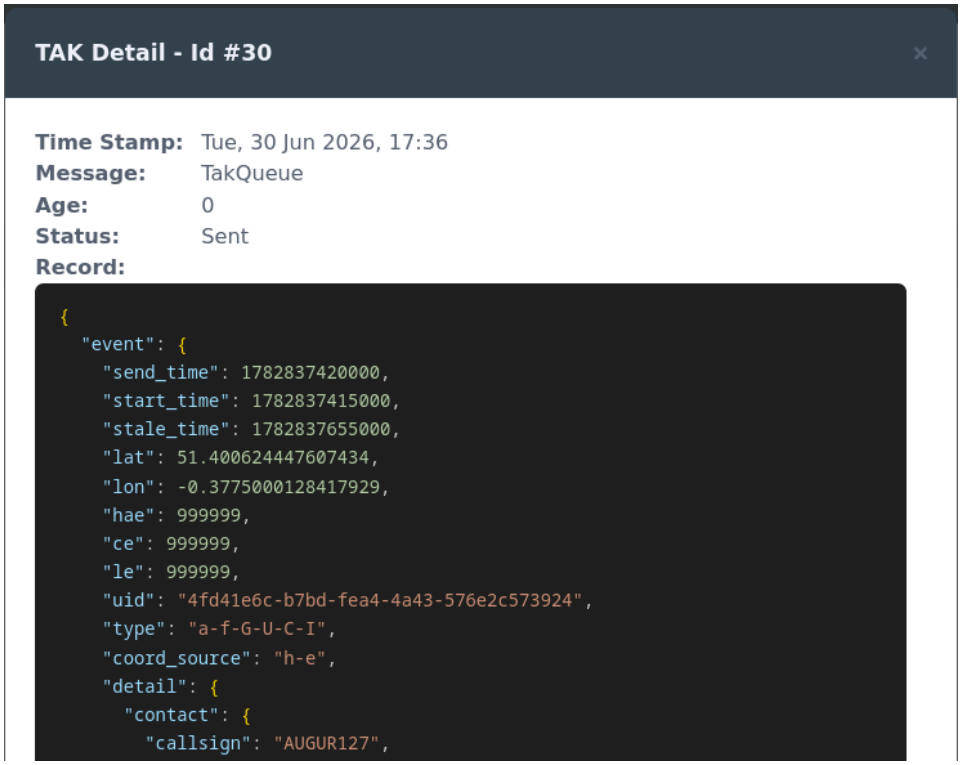
This shows a summary of all IP Client traffic passing through the server. If TCP proxying is enabled, then this will not include TCP traffic. The **Pause** button allows the display to be paused.

4.7 TAK monitoring



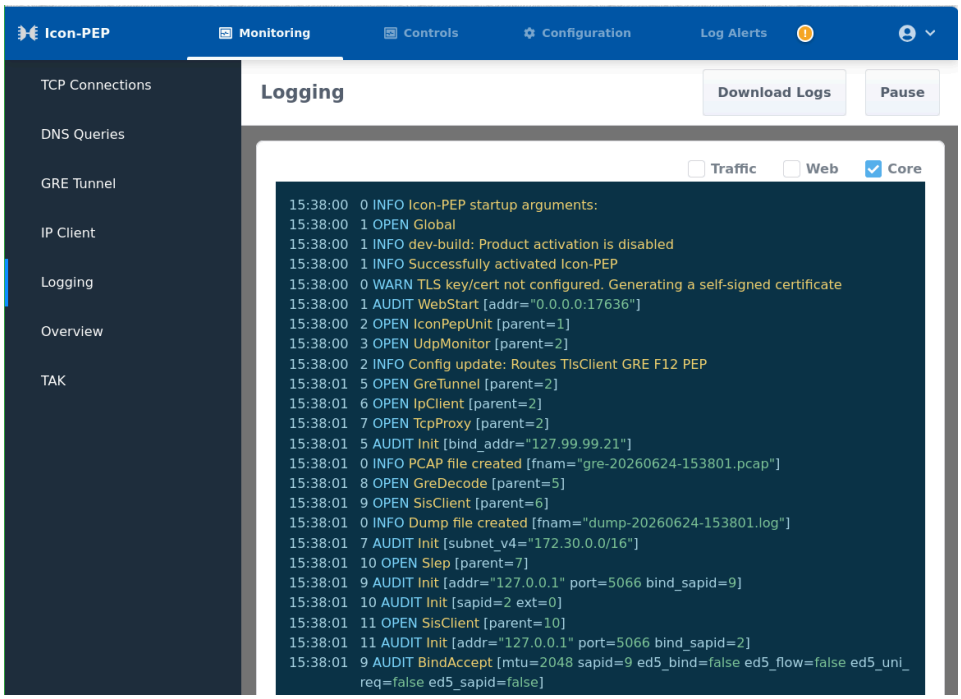
When the TCP TAK proxy is in operation, this shows incoming and outgoing TAK traffic. For outgoing messages, it also shows the final decisions taken by the enabled TAK thinning strategy. Messages may be marked with **Sent** if they were committed to HF, **Discarded** if the message was dropped due to the queue having grown too long, or

Replaced if the message has been replaced by a newer message. The age of the message is indicated in seconds.



The **More Info** button shows a detail display which includes a dump of the message contents.

4.8 Logging display

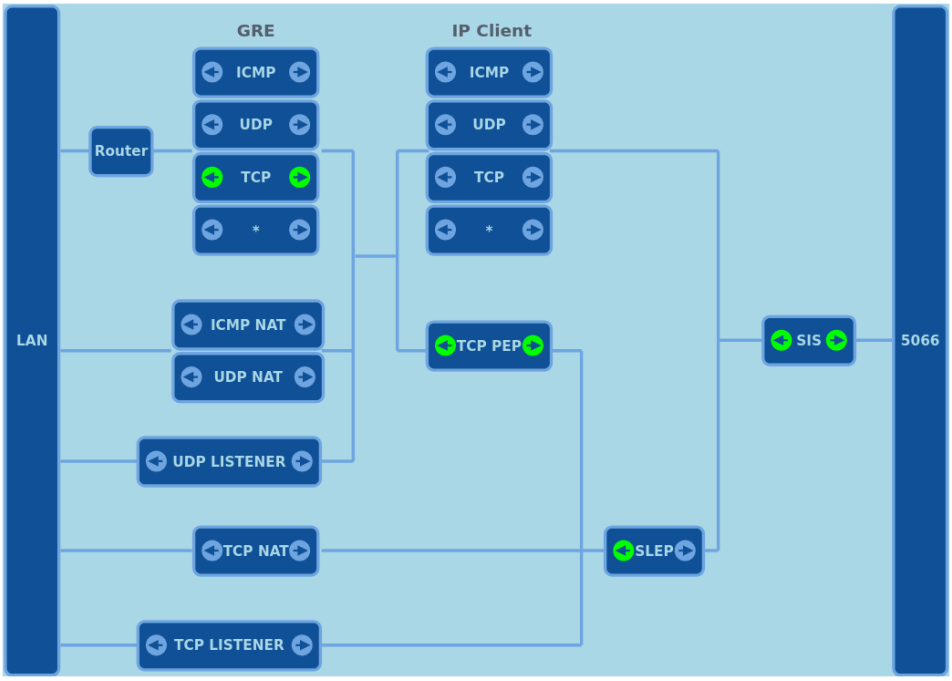


This shows logging from Icon-PEP. By default only **Core** logging is shown. In addition you may show traffic logging and web logging. Traffic logging consists of metrics used

to drive the other monitoring displays. Web logging shows logging associated with the web connections to the UI front-end or other HTTP clients. The **Pause** button allows the display to be paused.

The **Download Logs** button downloads a ZIP containing today's logs and the current config. It is a quick way to package up all the information required to diagnose a problem. No crypto assets are included, but it will include dump files for today if they were enabled.

4.9 Overview display



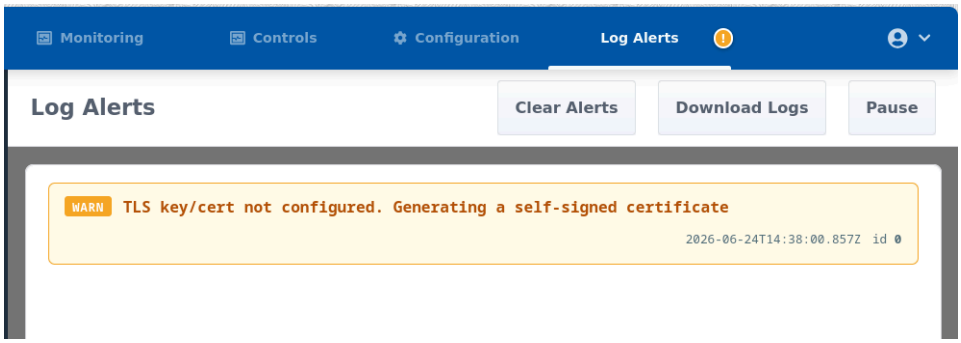
The purpose of this display is to give an overview of all parts of the processing chain, and to show whether traffic has flowed through them, and in which direction. This can help understand where the problem is if things aren't working in initial setup. The arrows become highlighted when traffic is detected. The **Reset** button can be used to clear all the highlighted indications. Only new traffic since **Reset** was pressed will cause them to be highlighted again. **Pause** may be used to pause the display.

SIS shows SIS traffic to/from the S'5066 server (e.g. Icon-5066). The two protocols used over HF are **IP Client** (for IP packets) and **SLEP** (for proxied TCP streams). Some of the different types of traffic are broken out for **IP Client**, with the rest put under the asterisk.

On the LAN side, depending on the configuration, traffic may pass over a GRE tunnel (broken out into several protocols), or via NAT proxies or listeners. In the middle, the **TCP PEP** component converts IP packets carrying a TCP stream into the raw stream of data to be carried over SLEP.

Note that there may occasionally be counter-intuitive indications, for example traffic showing in one component, but not an associated component in the same path. This occurs because each component may be working at a different protocol layer. For example, there may be IP packets and SLEP PDUs going back and forth to make a new TCP connection, but no actual stream data passed just yet.

4.10 Log Alerts tab



This gives alerts when warnings or errors occur in the logs, with a visible indicator showing the severity. The **Clear Alerts** button can be used to clear the display and indicator once the alerts have been read. The purpose is to get quick feedback if things are going wrong, especially whilst making configuration changes.

4.11 Logging files

Logging appears in dated files under *(LOGDIR)*. Files roll over daily. In the following example, some lines have been wrapped to fit the page width.

```
20230831-172208.034    0 INFO  Timezone is +0000
20230831-172208.034    0 INFO  Icon-PEP startup arguments: --T
20230831-172208.035    1 OPEN  Global
20230831-172208.036    2 OPEN  Tui [parent=1]
20230831-172208.043    1 INFO  Successfully activated Icon-PEP
20230831-172208.045    3 OPEN  Terminal [parent=2]
20230831-172208.136    1 AUDIT WebStart [addr=0.0.0.0:17636]
20230831-172208.138    4 OPEN  IconPepUnit [parent=1]
20230831-172208.138    5 OPEN  UdpMonitor [parent=4]
20230831-172208.138    4 INFO  Config update: Routes GRE F12 PEP
20230831-172208.239    6 OPEN  WsListen [parent=1]
20230831-172208.239    7 OPEN  GreTunnel [parent=4]
20230831-172208.240    8 OPEN  AnnexF12 [parent=4]
20230831-172208.240    9 OPEN  TcpProxy [parent=4]
20230831-172208.240    7 AUDIT Init [bind_addr=127.99.99.22]
20230831-172208.240    0 INFO  PCAP file created
  [fnam=gre-20230831-172208.pcap]
20230831-172208.240   10 OPEN  GreDecode [parent=7]
20230831-172208.241   11 OPEN  SisClient [parent=8]
20230831-172208.241    0 INFO  Dump file created
  [fnam=dump-20230831-172208.log]
20230831-172208.241    9 AUDIT Init [subnet_v4=172.31.0.0/16]
20230831-172208.241   12 OPEN  Slep [parent=9]
20230831-172208.249   11 AUDIT Init [addr=10.222.0.1 port=6066]
20230831-172208.249   12 AUDIT Init [sapid=2 ext=0]
20230831-172208.249   13 OPEN  SisClient [parent=12]
20230831-172208.250   13 AUDIT Init [addr=10.222.0.1 port=6066]
20230831-172208.251   11 AUDIT BindAccept [mtu=2048]
20230831-172208.251   13 AUDIT BindAccept [mtu=2048]
20230831-172208.251    9 AUDIT Bound [mtu=2048]
20230831-172242.859   10 INFO  Incoming GRE packet type:
  -!key -!seq -!checksum
```

```
20230831-172313.296    5 AUDIT DnsQuery [client=10.8.2.1
  server=172.20.0.66:53 rec=A name=www.isode.com tcp=false]
20230831-172358.827    5 AUDIT DnsAnswer [client=10.8.2.1
  server=8.8.8.8:53 rec=A name=www.isode.com answer=46.32.229.212
  tcp=false ttl=600]
20230831-172502.900   16 OPEN  TcpConn [parent=9]
20230831-172503.104   17 OPEN  SlepStream [parent=12]
20230831-172503.105   17 AUDIT Initiator [node=4.0.0.1 xfid=31364]
20230831-172503.105   16 AUDIT FromLan [key=1 node=4.0.0.1
  xfid=31364]
20230831-172503.105   16 AUDIT ConnKey [hf_addr=46.32.229.212
  hf_port=80 lan_addr=10.8.2.1 lan_port=36826]
20230831-172503.106   16 AUDIT HttpRequest [method=GET
  host=www.isode.com path=/ src=10.8.2.1:36826 dst=46.32.229.212:80]
20230831-172510.375   17 AUDIT InState [state=Ready]
20230831-172510.377   16 AUDIT HttpResponse [code=301
  host=www.isode.com path=/ src=46.32.229.212:80 dst=10.8.2.1:36826
  length=162]
20230831-172537.384   17 AUDIT OutState [state=Flushing]
20230831-172537.384   17 AUDIT Compression [input=142 output=140
  percent=98.59]
```

The logging columns are as follows:

`20230831-172208.034`
: Date as YYYYMMDD, time as HHMMSS and subsecond time in milliseconds.

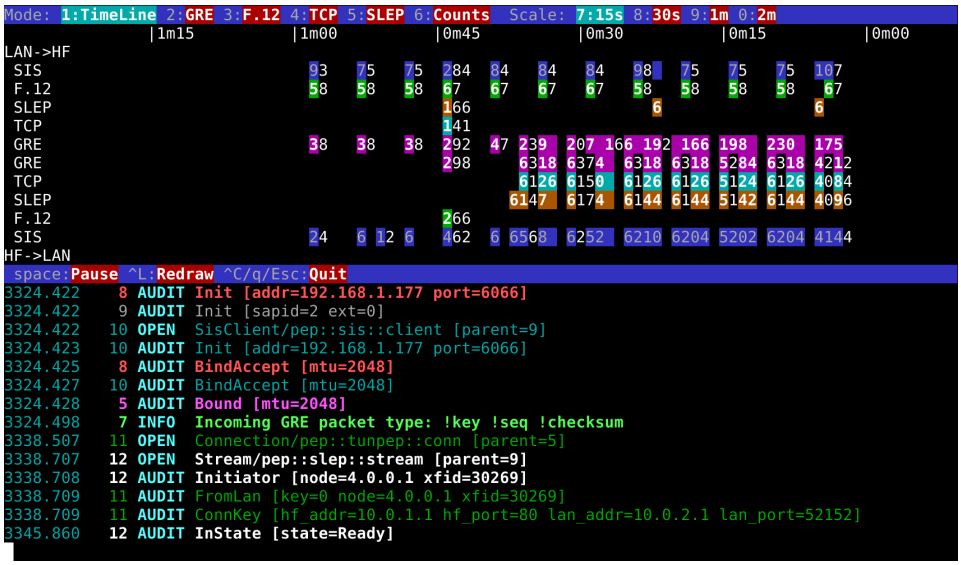
`13`
: Logging span number. Each instance of a component has a different logging span number. All logging from that component, between `OPEN` and `CLOSE`, is logged against the same span.

`AUDIT`
: Type of logging record: `OPEN` and `CLOSE` record start and end of a span, `AUDIT` logs machine-readable data, `TRACE`, `DEBUG`, `INFO`, `WARN` and `ERROR` log human-readable information at various severity levels.

`OutState [state=Flushing]`
: Data associated with the logging record. The values within square brackets are machine-readable key-value pairs.

4.12 Command Line Monitoring

Figure 4.1. Command Line Monitoring



The command mode of Icon-PEP provides a simple character based GUI to visualize data flowing over GRE, TCP, SLEP, IP Client and STANAG 5066 SIS. This gives a front-panel display to Icon-PEP that may be useful during initial setup or whilst debugging a configuration remotely where it is inconvenient to use the web-based monitoring display. There are the following options, accessed by keyboard control:

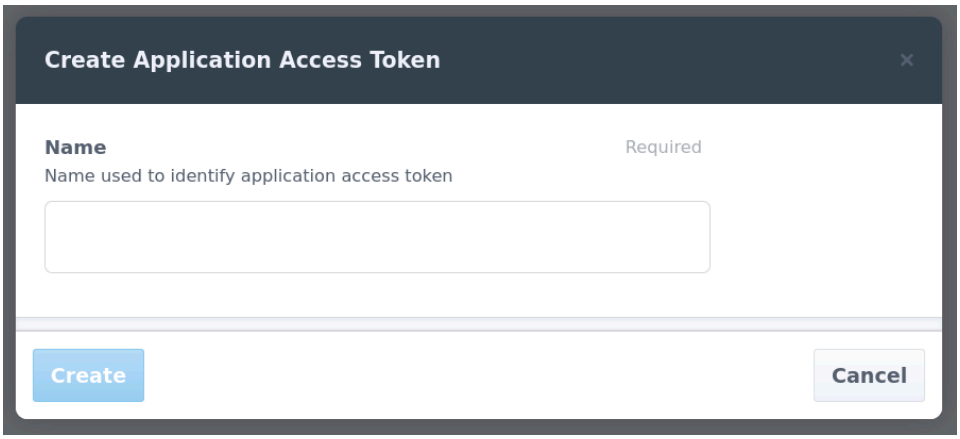
1. Timeline (shown above), gives a graphical flow of inbound and outbound traffic at each level. Traffic moving from the LAN towards HF appears in the upper part, and from HF towards the LAN in the lower part.
2. GRE. Data flowing at the Generic Routing Encapsulation layer (GRE).
3. IP Client (previously Annex F.12). Data flowing over STANAG 5066 Annex U IP Client.
4. TCP. Data flowing in TCP Tunnel.
5. SLEP. Data from TCP Tunnel flowing over STANAG 5066 SLEP Streaming Service.
6. Counts. Show summary of packet counts at each layer.

Chapter 5 General management API

This chapter describes the web-API that allows external tools to manage Icon-PEP.

This is where interfaces will be exposed for the use of Icon-Topo or Red/Black. Icon-Topo is a product to support Mobile Unit (MU) mobility between HF Networks. See the Icon-Topo product page: https://www.isode.com/products/icon-topo.html. Red/Black is a management product that monitors and controls devices and servers. See the Red/Black product page: https://www.isode.com/products/red-black.html.

5.1 Enabling access to the API



Access to the API requires an access token. Go to the **Access Tokens** section under **Configuration** and select **Add token**. The UI will provide a token which must be copied and saved in the configuration of the external tool immediately, and not saved anywhere else. Along with the IP address and port of Icon-PEP, this will enable access by the external tool to Icon-PEP.

5.2 Supported actions

The following actions may currently be performed:

1. Update the entire HF routing table.
2. Change the default HF route to direct traffic to the provided S'5066 node.
3. Abort all traffic to/from the provided S'5066 node.

This interface could be supported for other uses than Icon-Topo or Red/Black if required. Contact Isode support if this is of interest.