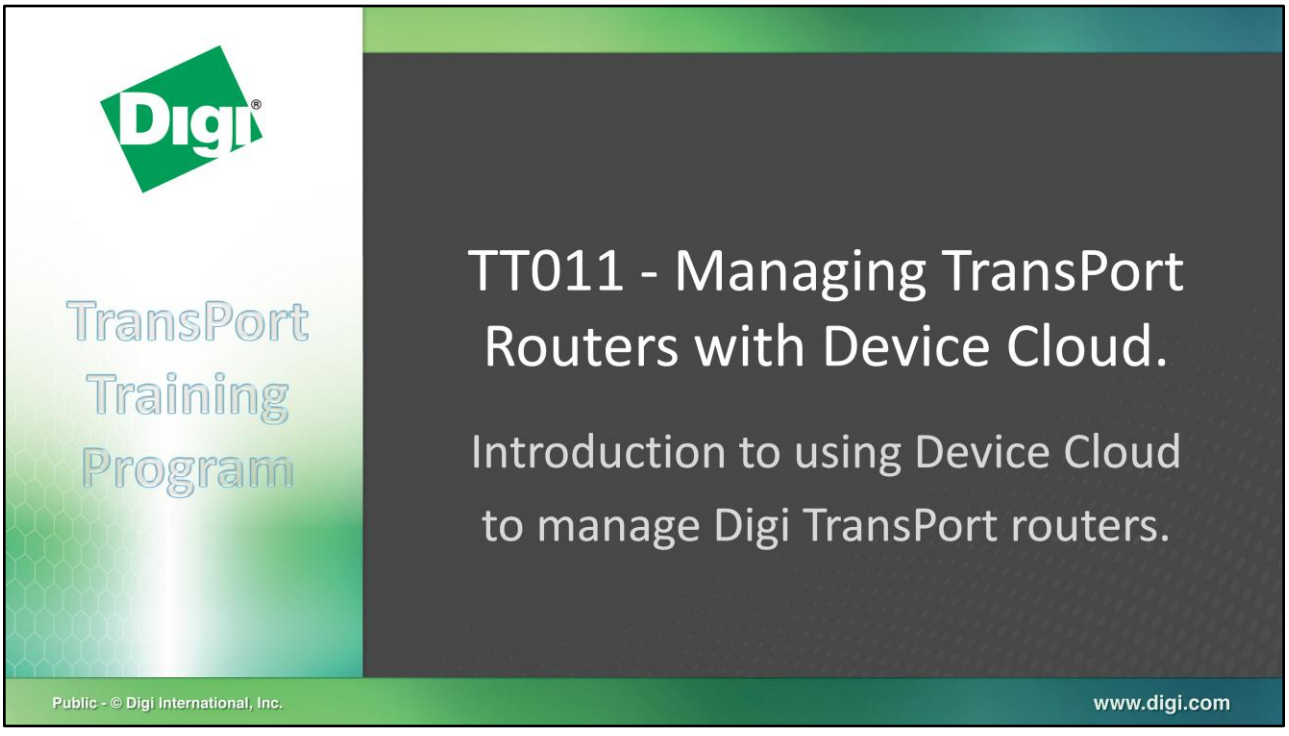
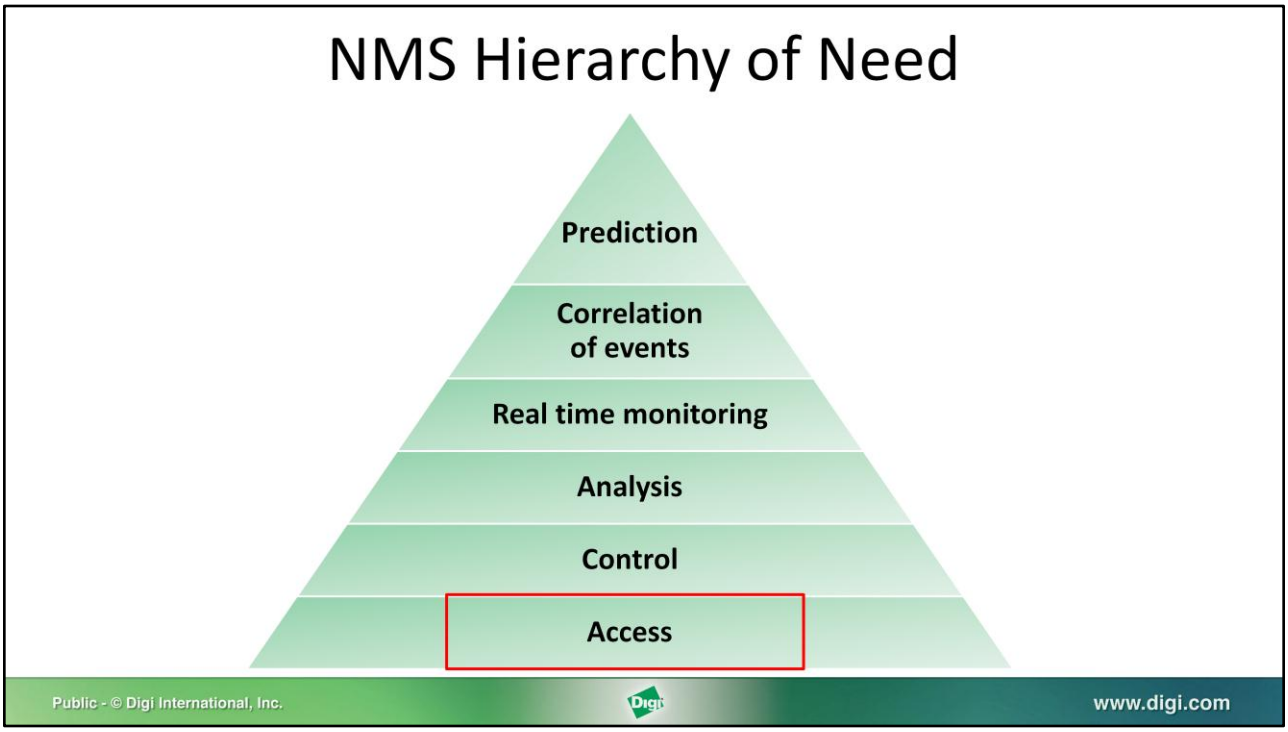

TransPort
Training
Program
TT011 - Managing TransPort Routers with Device Cloud.
Introduction to using Device Cloud to manage Digi TransPort routers.
Public - © Digi International, Inc.
www.digi.com

Agenda

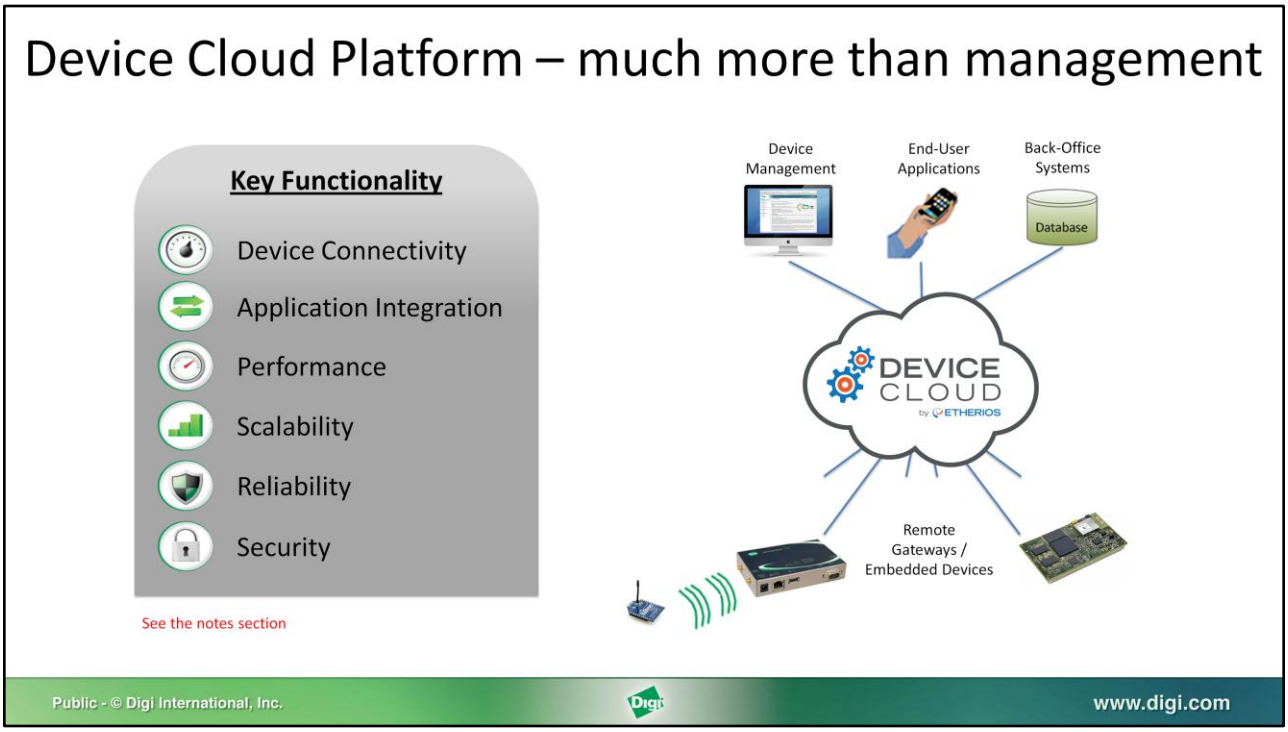
- NMS Hierarchy of Need
- Device Cloud Platform
- Access
- Control
- Analysis
- Monitoring
- Hand On Practical Session



This is a Hierarchy of need triangle for a **N**etwork **M**anagement **S**ystem. The largest and more fundamental needs are at the bottom. Only once the needs at the bottom have been met is it worth considering needs higher up the triangle.

We're going to concentrate on the first four "needs" as these are the most generally desired. We will see how these needs can be met on Digi TransPort routers using Device Cloud.

Access – getting connected
Control – changing configuration, firmware & rebooting.
Analysis – viewing performance in the past tense – reviewing what has happened
Real time monitoring – Real time performance information – alarms etc



Device cloud is much more than a network management tool. The platform's primary original purpose was to collect data from end devices and make it available to the application that needs it in a truly scalable way.

Device cloud is only available via the Internet as a service offering. It is not a software package that can be self hosted.

Let's look at some key functionality:

Device Connectivity

Get's you or your software connected to your device. This is how we achieve the first layer in our NMS hierarch of need – the Access. Normally each device maintains a constant TCP/SSL connection to Device Cloud. To do this the device must speak a protocol called EDP. (**E**therios **D**evice **P**rotocol).

Application Integration

A very complete set of API's exist to build an interface between Device Cloud and an application. These are implemented as RESTful web services. These APIs allow the data collected from Devices to be pulled or pushed to an application that needs them. In addition they allow an application or script or program to send messages to

devices. Finally these APIs can be used by an application, script or program to manage devices. For example to change the configuration of firmware that is running on routers. In the context of this module, in essences it allows other IT systems or even scripts to utilise the power of Device Cloud for managing Digi TransPort routers.

Performance

Performance is a key requirement of a data collection platform like Device Cloud. During testing with 1 Million simulated devices a figure of 2500 transactions per second was achieved. Performance is less critical for a NMS however the full performance capabilities of Device Cloud are available for the management functionality.

Scalability

Device cloud has been designed from the ground up to be scalable. The expansion of registered devices does not require planning and/or lead time by the end customer. Device cloud is architected in a manner that allows Digi to increase the instances of the various components required to meet the demand.

Reliability

Reliability is measured as percentage system up time. The current service level is 99.9% uptime but Digi is moving toward 99.99%

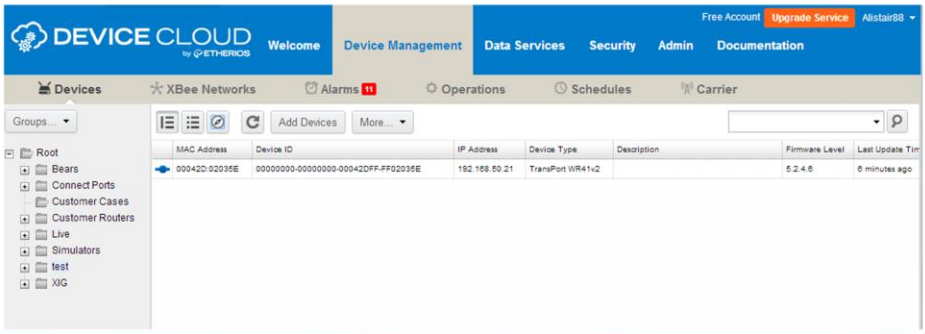
Security

It goes without saying, security is very important for any cloud system. Device Cloud has a security policy of currently 175 security controls. These control everything from the technology and algorithms used to encryption and authenticate (in most cases SSL) through to the requirement to tightly control access to the rooms hosting the physical infrastructure.

These security controls are based upon NIST, NERC CIP, ISO27001, CSA. Our datacenter is SSAE-16 certified (new SAS-70 type II). In addition Device Cloud is ISO27001 certified. Digi is working towards a PCI-DSS RoC (Payment Card Industry – Report on Compliance) so that Device Cloud can be used as an “in scope component” of a PCI-DSS certified system.

Device Cloud User Interface

- http://login.etherios.co.uk (UK instance) or http://login.etherios.com (US instance)
- Most Management activities performed under the Device Management Tab.
- Device Cloud User Guide available under the Documentation Tab – READ THIS
- Sign up for free developer edition (5 devices only) here: http://www.etherios.com/products/devicecloud/developerzone



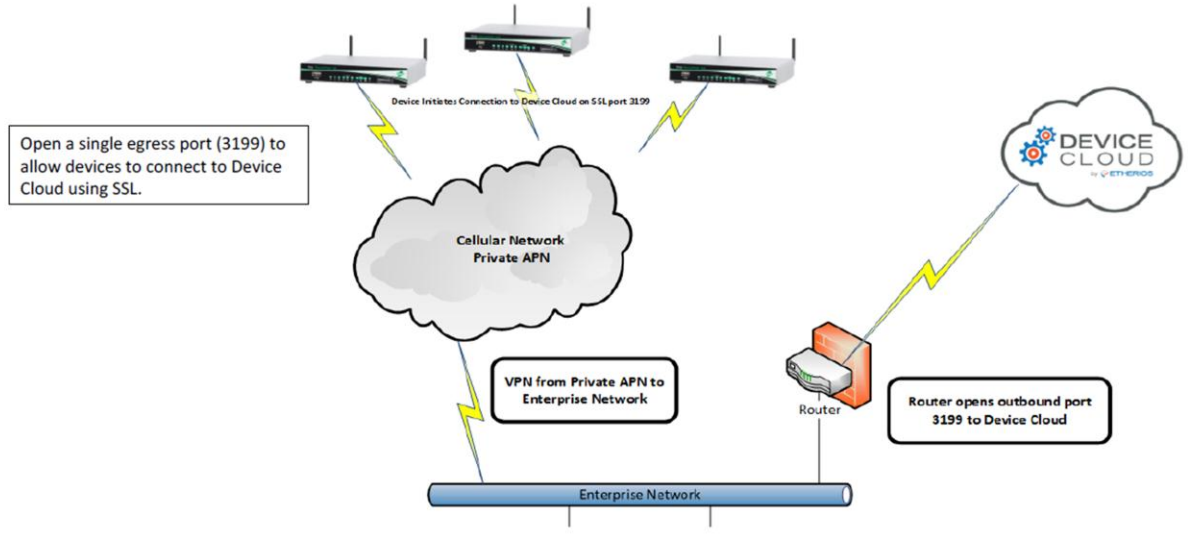
Access - Connectivity

- Normally router maintains constant SSL connection to cloud.
- SMS can be used instead for both configuration, message passing and control of the SSL connection. (SMS cannot be used for a firmware upgrade.)
- The SSL connection is EDP (Easy Device Protocol) which has a configurable keep alive interval to keep the link active. (Needed to traverse firewalls and NAT routers.) Typical keep alive data usage is:

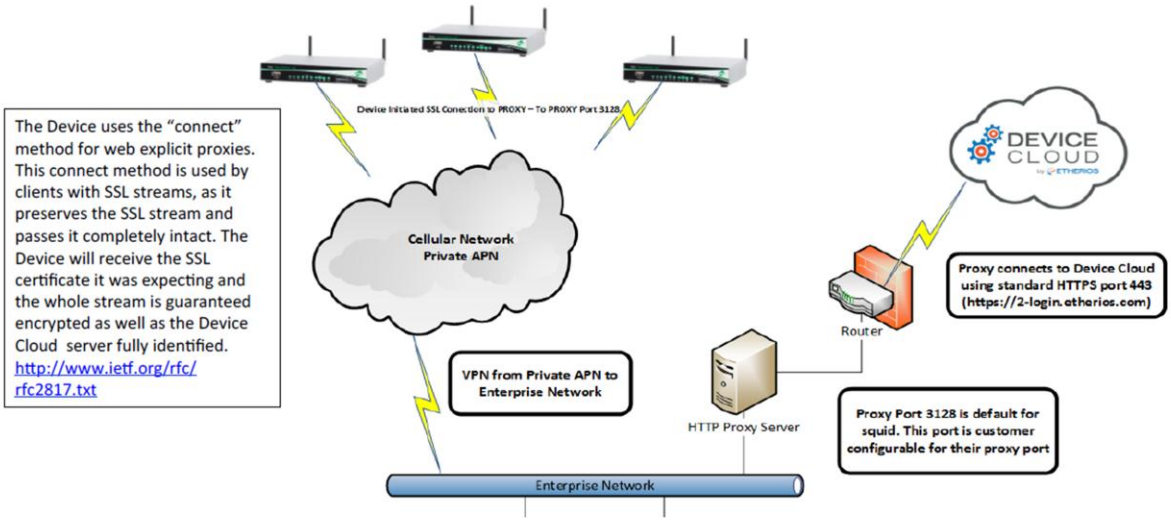
5 mins KA = 279 bytes x 12 x 24 x 30 = ~2.3 MB
10 mins KA = 279 bytes x 6 x 24 x 30 = ~1.2 MB
20 mins KA = 279 bytes x 3 x 24 x 30 = ~590kB
30 mins KA = 279 bytes x 2 x 24 x 30 = ~400kB]

- Options for connecting from private networks (e.g. private APN) follow

Open Single Egress Port

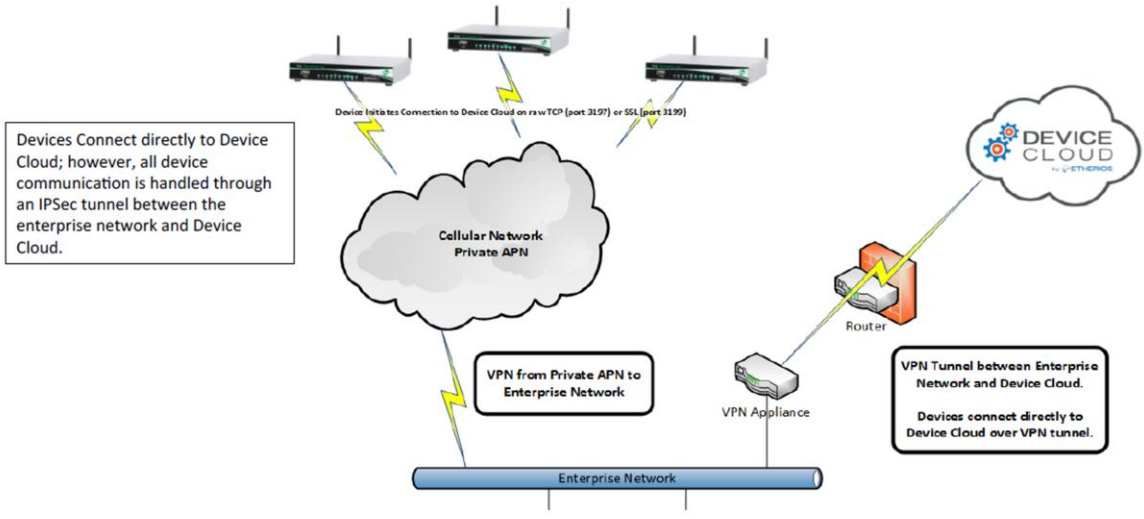


Utilise Standard Web Proxy



Coming later this calendar year. SarOS needs updating to add this support.

IPsec VPN Tunnel

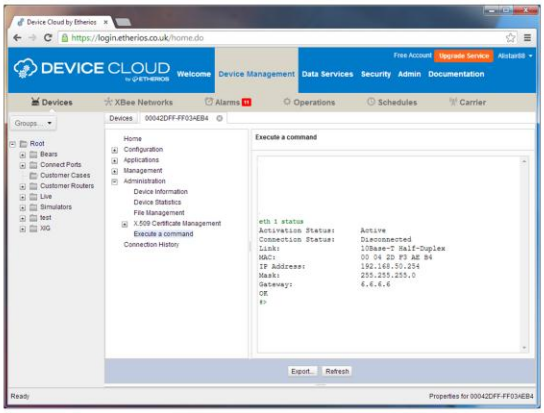


Costly option

Control

- Properties screen is an “alternate” user interface built inside of Device Cloud.
- Cached data shown if router is off-line.
- Useful to make changes on one router at once.
- Click – scroll – type allows settings to be changed.
- File system allows files to be transferred to/from router. (Including firmware and config files.)
- Execute a command
- Right click – update firmware

DEMO !



Control

Provide a live walk through of the above features – demonstration.

Configuration Control – XML RCI Method

- Right click - export configuration in RCI format
- Edit the RCI with an XML editor
- Right click - deploy this configuration to other routers
- Use the UI to generate an XML RCI file with just the changes required. Resulting file can be rolled out to other routers
- Alternative is to deploy individual configuration files or schedule CLI commands required to make changes.

See the notes section

There are several methods that can be use to manage configuration changes on TransPort routers via Device Cloud. For example you can issue CLI commands and work with the SarOS configuration files (config.da0, sregs.dat etc)

This is the XML RCI method. Hopefully most of you are familiar with XML?

RCI stands for Remote Command Interface – in this context it's basically a XML configuration file for the Transport Router wrapped up in an RCI instruction to apply the configuration.

Right click - export configuration in RCI format

Right clicking on a device or router in the device list will allow us to download an XML file from the router. This is the major part of the configuration of the router.

Edit the RCI with an XML editor

It's possible to edit this file with an XML editor or even a text editor. This might be required in order to make a configuration change to the file. Alternatively you might need to strip out most of the content of file leaving in just the parts of this configuration you wish to apply to other routers.

Right click - deploy this configuration to other routers

Once the new XML file has been created, you can apply it to other routers via multi-select and right click.

Use the UI to generate an XML RCI file with just the changes required.

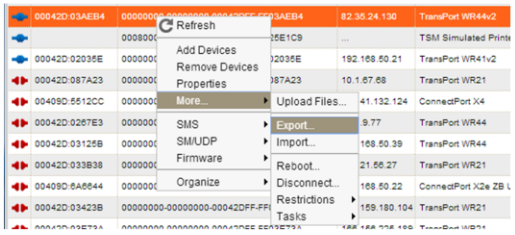
It's also possible to user the device properties user interface to generate an XML file which just contains the set of changes you wish to apply – presumably to multiple other sites. We'll cover this a little later.

Alternative is to deploy individual configuration files or schedule CLI commands required to make changes.

An alternative configuration management method is to use Device Manager to deploy prepared individual SarOS configuration files and/or issue SarOS Command Line Interface Commands.

Right click - export configuration in RCI format

Right click → More → Export. This results in a single .XML file e.g. "00042DFF-FF03AEB4.xml" that contains most of the router configuration.

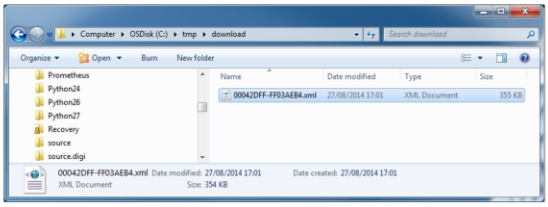


It contains the

- main configuration (config.dax and pwds.dax files)
- serial port configuration (sregs.dat file)

It does NOT contain:

- The firewall script (fw.txt)
- X.25 profiles (x3prof)
- Routing protocol configuration files (usually bgp.conf & ospf.conf)
- User created files and scripts (e.g. Python scripts)



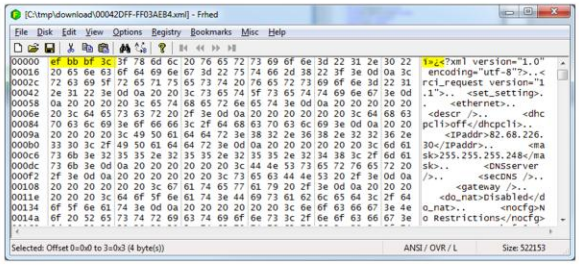
If your configuration has any elements listed in the "NOT contain" list these must be dealt with separately.

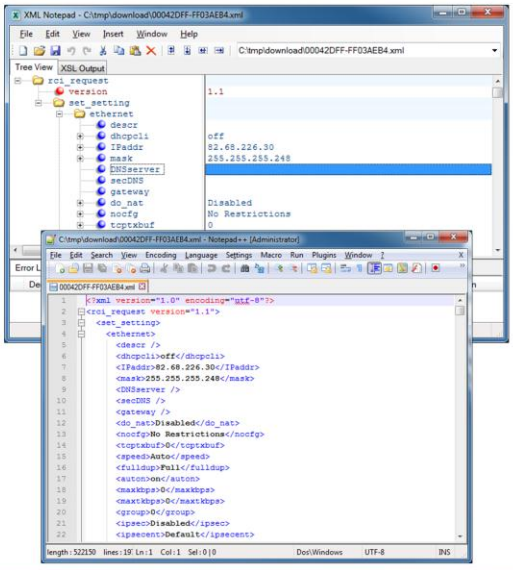
Edit the RCI with an XML editor

Study the format of the XML file. This is RCI.

Note that Microsoft XML notepad can help re-format the XML file to make it more readable by humans. But adds invalid binary characters to the beginning of the file that must be removed e.g. by FrHed (Free Hex Editor)

If necessary, edit the XML file with an XML editor to change settings or remove site specific components.





Right click - deploy this configuration to other routers

Use:

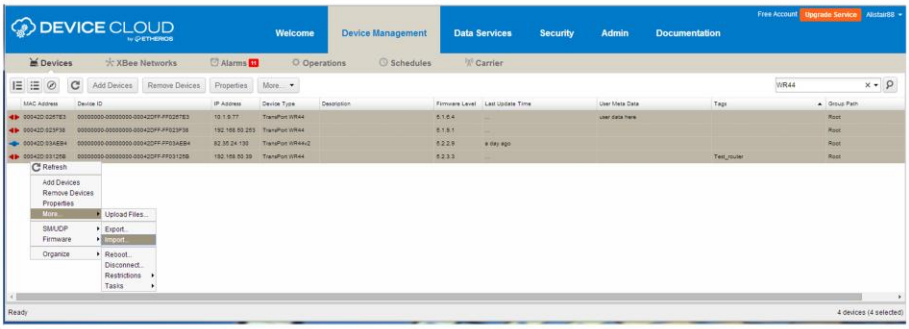
- Search Box
- Flat view
- Order by Column (e.g. by tags)

To multi select the routers you wish to roll out the configuration change to.

Right click → More → Import

Then upload the edited XML file.

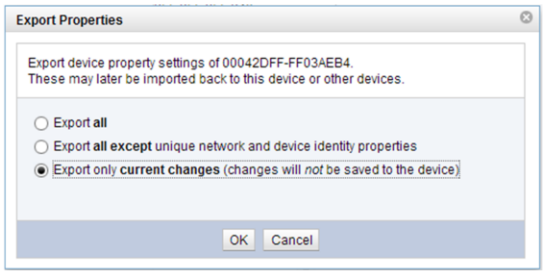
The alternative is to use a Scheduled Operation (covered later) to "import" the file to routers in selected groups or matching certain tags.



Demo some of the user interface features, such as flat view vs groups. Order by columns and multiple selections.

Use the UI to generate an XML RCI file with just the changes required.

- Alternative to manually editing an XML file
- Load the properties screen for a router
- Make the configuration changes
- Click Export → Export Only Current Changes
- Resulting XML file contains ONLY the changes made in the user Interface.
- Resulting XML file can be pushed to other sites by right click on multi selection in Device List view or a Scheduled Operation.
- Demo

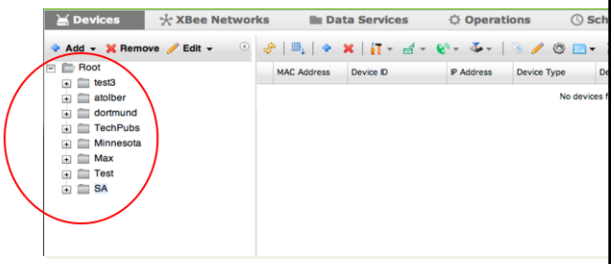


Tags

- Non-hierarchal, any router can have any tag associated with it.
- Used to organise and apply operations to specific sets of routers.
- Demo of editing a tag

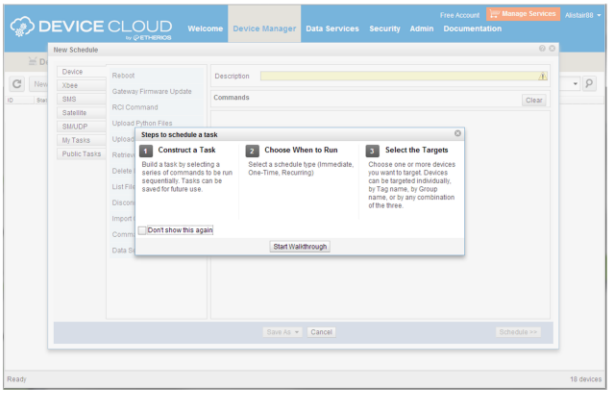
Groups

- Groups organise sites into hierarchical folder structure
- Add, Remove, Edit
- Unlimited number of devices can be assigned to a group
- Used for organisation and to apply operations to multiple groups.
- Demo – create group and move a device into that group



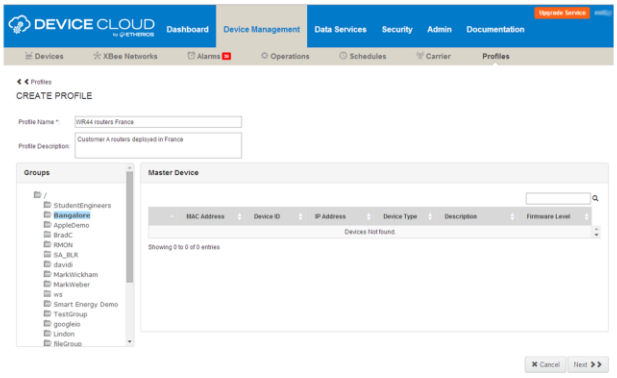
Scheduled Operations

- Very Powerful
- Wizard-Driven
- Build up a task list from multiple commands.
- Assign a task to a single device, a single group or multiple groups or tags.
- Schedule immediate, one-time or recurring tasks
- Review log of completed tasks
- Tasks can be configured to execute the next time a device connects if offline.



Profile Manager

- First version coming September 2014
- Allows a “profile” to be assigned to group, tag or device
- Profile is
  - configuration template
  - firmware version
  - site specific parameter rules
  - file system files
- If a device does not match the profile the options are:
  - Bring into compliance automatically
  - Generate an Alarm
- Needed for
  - Automatically commissioning new units firmware/config including site specific settings
  - Reporting on compliance with configuration profile (e.g. for PCI DSS purposes)



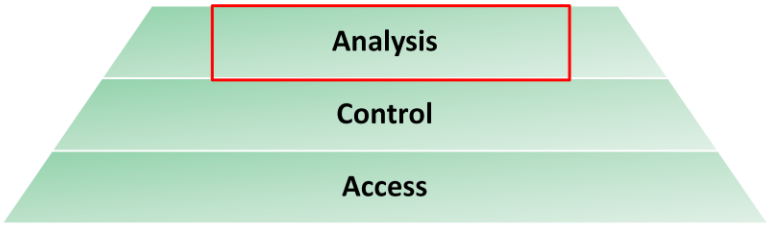
Note that there is a configurable scan interval for checking that devices match the profile.

Scripting

- All of the previously discussed features available via the UI can be controlled programmatically via RESTFul web service interface.
- This allows simple scripts to have very powerful management capabilities.
- This allows third party systems to be integrated with Device Cloud.
- Built in API Explorer can be used to build script templates in Python, Java, Ruby, Perl and C#
- Built in API Explorer can be uses to perform tasks not supported in the User Interface (e.g. send a message to a Python program running on a TransPort router or group of routers)
- Check out Documentation → Resources → Device Cloud Programming Guide
- Check out Documentation → Resources → API explorer

Analysis

- Definition: Review of past performance
- Eventlog and/or capture file retrieval
- Retrieval of statistics (query_stat API call)
- Device Health → September 2014



Health Monitor Data

- Some data is easily available without generating extra traffic:
  - Signal Quality
  - RSSI (Signal Strength)
  - Uptime
- To really understand network performance though, testing is required.
  - Monitor pings or UDP data
  - Collect stats on the performance (latency, packet loss)

Health Monitor Cellular Data Element Summary

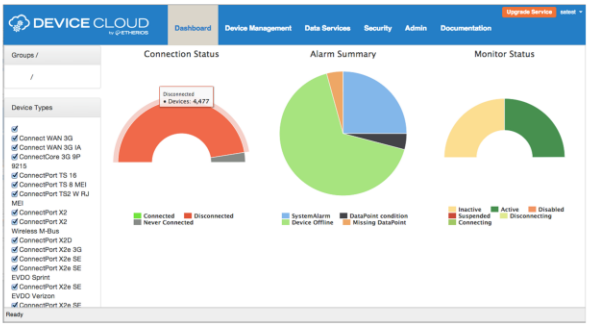
Rolled Up Data (default period of 1 hour)	
Number of bytes received during the sample period	metrics/mobile/<n>/net/<n>/<tech>/rxbytes
Number of bytes transmitted during the sample period	metrics/mobile/<n>/net/<n>/<tech>/txbytes
Minimum latency for monitored ICMP/UDP packets during the sample period	metrics/mobile/<n>/net/<n>/<tech>/latency/min
Average latency for monitored packets during the sample period	metrics/mobile/<n>/net/<n>/<tech>/latency/avg
Maximum latency for monitored packets during the sample period	metrics/mobile/<n>/net/<n>/<tech>/latency/max
Number of times a reply was received for monitored packets	metrics/mobile/<n>/net/<n>/<tech>/transactions/count
Number of times a reply was NOT received for monitored packets	metrics/mobile/<n>/net/<n>/<tech>/drop/count
Number of times link was deactivated due to firewall detecting no response	metrics/mobile/<n>/net/<n>/<tech>/oos/count
Number of seconds the cellular interface is up per hour	metrics/mobile/<n>/net/<n>/<tech>/uptime

Other mobile data includes
• Signal Quality measure
• RSSI
• Network
• Cell ID
• Location are code
• Temperature of cell modem
• ICCID
• IMSI

Device Health – High Level Summary

- Active Monitoring and Alerting on Device Health
- Account-level dashboard illustrating roll-ups for installed inventory
- Visualize roll-ups for entire inventory, specific groups or individual devices

- API's already designed
  - General Network Metrics
  - Mobile Network Metrics
  - Ethernet Network Metrics
  - Wi-Fi Network Metrics
  - System Metrics
  - Device GPS Streams

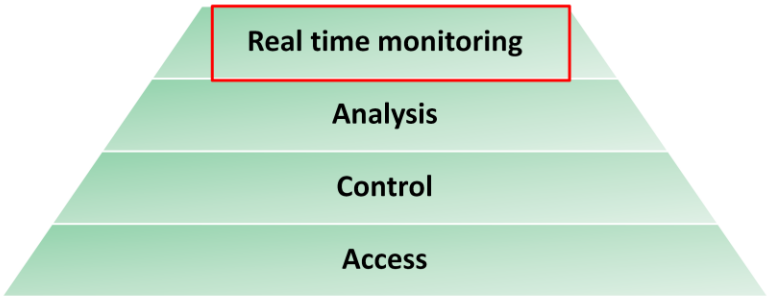


Device Health – Detailed View per Site

- Latency & Packet Loss
- Signal strength and quality
- Data volume
- Down time
- Out of service count

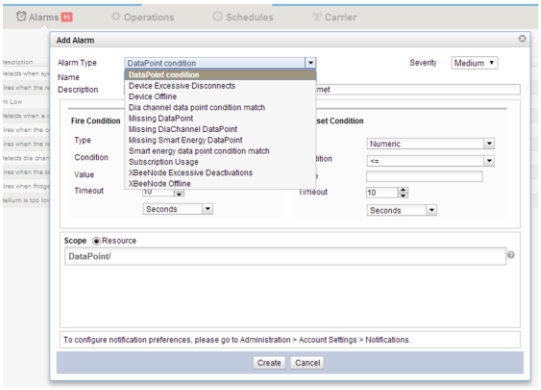
Real Time Monitoring

- Alarms
- Monitor API
- Map



Alarms

- A number of built in alarms exist:
  - Device Excessive Disconnects
  - Device off-line
  - Data point condition match
  - Missing data point
- Alarm notification options:
  - Email
  - Pull/push to 3rd party application (monitor API)
  - Device Management User Interface
- Demonstration of alarm creation



The data point condition match allows alarming based on just about any conceivable metric. This should be used in conjunction with a Python program running on the router to upload the metric in question to the DataStreams service of Device Cloud.

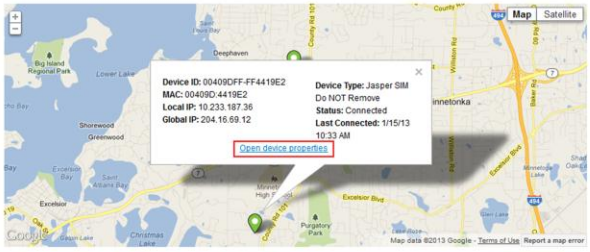
Covered in another module is the Digi ESP IDE (Integrated Development Environment). This can be used to create and run a DIA Python project on a Digi TransPort router. This makes it easy to push any kind of custom metric to a Device Cloud data stream.

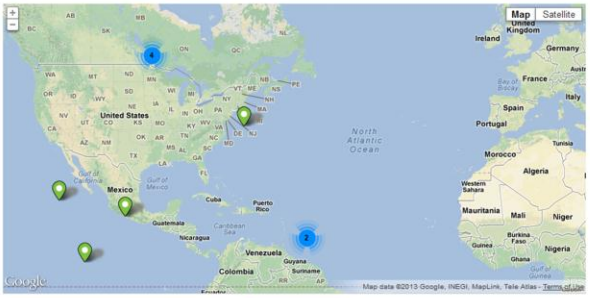
Monitor API

- The monitor API is a publish and subscribe mechanism that allows a third party application to receive data from Device Cloud. In other words push notification.
- The monitor API can be used to monitor various topics in Device Cloud including:
  - Alarms
  - Device Core (connections, disconnections, device provisioning
  - DataPoint
  - DataStream
  - FileData

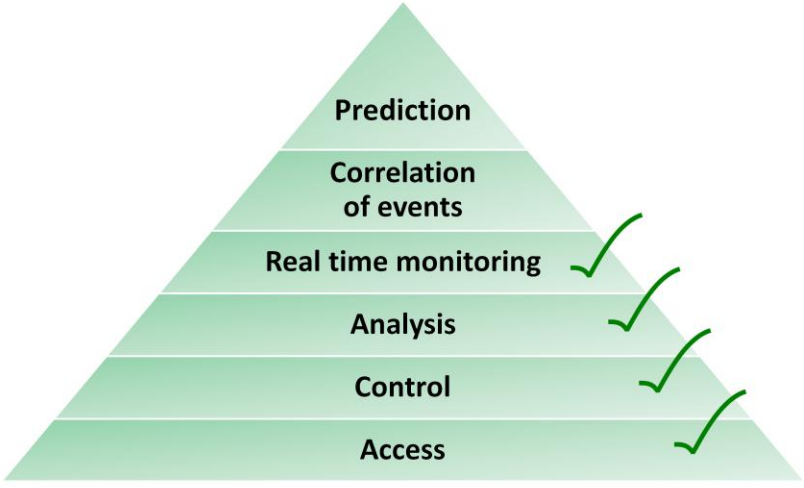
Map

- Clicking the compass button under Devices will show or hide the built in map.
- The map displays a limited amount of real time information such as:
  - Connection Status
  - Location
- It's possible to click on an item in the map to access the properties (configuration screen) of the router.
- If multiple devices are found in the same location and the zoom level is not high enough – a blue device count icon appears.





Summary



Hands on practical session

- Connect a TransPort router to DC and provision to account
- Download the eventlog.txt file.
- Schedule an operation to
  - upload an XML file to change an item of main configuration
  - upload a new firewall script fw.txt
  - issue the CLI command "fw" to make the new firewall active.
- Send a command to flash LEDs
  - If time permits repeat via a Python program created from API explorer.