Home Network Administration Protocol (HNAP) Whitepaper

Revised January 2009

Abstract
This whitepaper provides information about the Home Network Administration Protocol (HNAP), which enables network equipment manufacturers to allow programmatic configuration and management by remote entities. By implementing HNAP, manufacturers can better identify their devices and provide improved, more cost effective support.

Contents

Introduction

The Home Network Administration Protocol (HNAP) is an HTTP-Simple Object Access Protocol (SOAP)-based protocol that can be implemented inside of network devices to allow advanced programmatic configuration and management by remote entities. By implementing HNAP, network equipment manufacturers can better identify their devices and provide improved, more cost effective support. Using HNAP, manufacturers can display custom, device-specific information and tasks in applications that use HNAP in device discovery and management, such as Cisco's Network Magic.

HNAP was designed to be a simple, light weight protocol that is easy to implement inside of small cost-constrained hardware such as network routers, cameras and other small devices. Because the protocol is based on existing HTTP-SOAP standards, it is very flexible and easily extensible.

Benefits of HNAP

There are three high level benefits to vendors for implementing HNAP in a network device:

Accurate Topology Discovery

By implementing HNAP, a network device can accurately describe itself to applications that support HNAP and show detailed information about the device. A device can choose its type and subtype, define its iconic representation, and also expose properties such as model name, firmware version, and support contact information. This type of information is useful to consumers when managing their networks, and to support personnel when troubleshooting a problem.

Custom Task Extensibility

By implementing HNAP, a network device can display custom tasks for the device. For example, when a device with HNAP support is selected in an application, tasks related to that device can be displayed. Custom tasks allow device vendors to differentiate their devices by exposing to consumers support for their advanced or exclusive features.

Programmable API

Perhaps the largest and most comprehensive benefit of implementing HNAP on a network device is its ability to be silently managed by other management products. The full programmable API suite allows devices' network connections to be remotely managed and administered. For example, for a device such as a router, the programmable API can provide:

- Automatic port mapping management
- Network security with MAC filtering, WEP, and WPA
- Router Setup, provisioning WAN and LAN interfaces, including advanced settings such as PPOE username/password etc.
- Add new devices to the network
- Bandwidth monitoring
- DHCP Server Management and Static Reservations.

How HNAP works

Protocol Design

The HNAP protocol is based on the well known WC3 HTTP-SOAP standard defined in www.w3.org/TR/2003/REC-soap12-part1-20030624.

There are two distinct roles with any HNAP interaction – an HNAP server and an HNAP client. HNAP servers are typically implemented inside of networking devices to be managed. HNAP clients are usually software applications residing on PCs or other devices that can interact with an HNAP server in order to manage it, and ultimately, the device.

A typical client server interaction begins when a client has discovered an HNAP server on a network. It issues an HNAP discovery command in order to determine the capabilities of the device. A client then proceeds to make one or more HNAP requests to the server, which performs the desired action and returns the response.

For example, if an HNAP client needs to determine what firmware version a particular device is running, it might issue a GetFirmwareSettings request as shown below:

```
POST /HNAP1/ HTTP/1.1
Host: 192.168.0.1
Authorization: Basic YWMEHZY+
Content-Type: text/xml; charset=utf-8
SOAPAction: "http://purenetworks.com/HNAP1/GetFirmwareSettings"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope>
<soap:Body>
<GetFirmwareSettings xmlns="http://purenetworks.com/HNAP1/" />
</soap:Body>
</soap:Envelope>
```

The server would then respond with a GetFirmwareSettings response:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: 300

<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope>
<soap:Body>
<GetFirmwareSetttingsResponse xmlns="http://purenetworks.com/HNAP1/">
<GetFirmwareSetttingsResult>OK</ GetFirmwareSetttingsResult>
<VendorName>Linksys</VendorName>
<ModelName>WRT150N</ModelName>
<ModelRevision>2.00</ModelRevision>
<FirmwareVersion>1.21</FirmwareVersion>
<FirmwareDate>2005-05-31T17:23:18</FirmwareDate>
<UpdateMethods>
<string>HNAP_UPLOAD</string>
<string>TFTP_UPLOAD</string>
</UpdateMethods>
</GetFirmwareSetttingsResponse>
</soap:Body>
</soap:Envelope>
```

Note: In these examples, XML namespace information has been removed for clarity and ease of understanding.

By parsing the response, the client is able to determine the exact firmware build and model information for the device.

With the exception of the discovery command, all requests to the HNAP server are performed as an HTTP POST to http://{device-ip}/HNAP1/

The HTTP headers contain a SOAPAction field that defines the particular method being invoked. The payload of the HTTP body is an XML data block describing the parameters for the method invocation encoded in SOAP. Note also that an Authorization header is included to authenticate the caller.

HNAP Location Discovery

HNAP devices can locate other HNAP enabled devices on a network by leveraging the existing SSDP protocol. HNAP outlines a set of best practices for discovering HNAP capable devices over SSDP.

HNAP Capability Discovery

When an HNAP client needs to communicate with a device on the network, it will issue an HNAP discovery command to determine if the device in question supports HNAP, and if so, what specific capabilities that device has.

This HNAP discovery command is a simple HTTP GET request against a well known directory on the Web server. It is assumed that the Web service is running on port 80. If an alternative port is preferred, the response to the default GET request on port 80 may redirect to an alternative location and port for all subsequent POST requests.

An HNAP client must perform an HTTP GET against the following URL:
`http://{device-ip}/HNAP1/`

A valid response to this request informs an HNAP client that the HNAP protocol is supported by this device. The device can then further interrogate the response of GetDeviceSettings to determine which SOAP methods are implemented by this device.

Flexible Implementations

One of the key features of HNAP is its flexible implementations. A device may choose on a method-by-method basis, choosing which HNAP methods make sense for it to support. A device publishes which methods it supports by returning a list of SOAPActions returned by the GetDeviceSettings method. At a minimum, all devices must implement GetDeviceSettings. Some vendors may choose to implement only this method for accurate device discovery.

In general, it is highly recommended that devices support as complete of an implementation as possible; this allows the devices to be competitive with other devices from a management capability standpoint.

Protocol Security

HNAP leverages the same simple security mechanism available in most consumer network devices today – BASIC authentication, defined in RFC 2617, located at http://www.ietf.org/rfc/rfc2617.txt. It is recommended that an HNAP client attempt to use default credentials to negotiate with the device. If these default credentials are invalid (for example, the user has manually set a password different from the factory default), then the client should prompt the user for credentials so that it may manage the device.

An HNAP implementation may choose to require all HNAP session traffic to run over SSL to provide enhanced transport level security. To do this, simply redirect the initial HTTP-GET request to a secure port for all subsequent HTTP-POST requests.

Development Cost and Firmware Size Considerations

Cisco provides reference implementations of the HNAP protocol complete with sample source code. These implementations are based on well known open source routers, such as the Linksys by Cisco WRT54G and the NETGEAR® KWGR614.

Using these references as a starting point, most ODMs can very quickly port the source code to their platform in a few weeks. In most cases, ODMs prefer to re-use their HTTP server and XML parser implementations that already exist in their platforms.

Typically, a comprehensive implementation of HNAP for a Router device will add about 65k of additional size to the compiled and compressed firmware footprint

Testing an HNAP Implementation

After implementing HNAP in a device, you can test it with the Cisco validation tool, TestDevice, developed specifically to test and troubleshoot SOAP- and XML-based HNAP files. Testing involves completing a series of validation tests upon a device. This ensures that the device is HNAP compliant and the correct information displays for the device with no errors.
Deploying HNAP with network devices involves a close working relationship with Cisco's partner developers and testers. Your Cisco representative can help with HNAP deployment, development,

HNAP Capabilities for Specific Devices

All Devices

- Device type
- Vendor information
- Firmware information and updates
- Model name and description
- Serial number, time zone, and locale information
- Device rebooting
- Connection speed statistics

Wireless Devices

- Wi-Fi connection configuration
- Security settings configuration
- Support for devices with multiple radios
- Detailed capability discovery of channels and region-specific information

Routers

- WLAN, WAN, and LAN configuration
- Port forwarding
- MAC Address filtering
- Access point vs. router mode
- Connected client information

Internet Gateway Devices

- Configuration of DSL modem settings
- Configuration of MOCA modem settings

NAS Devices

- Share enumeration
- Share creation and management
- Physical and logical volume management
- Drive formatting and RAID configurations

Network Cameras

- Manage picture settings: white balance, sharpness, brightness, contract, saturation, hue etc.
- Programmatically pan, tilt, zoom and focus motorized cameras
- Take still snapshots
- Stream video
- Set video streaming settings
- Manage RTP settings

For More Information

To learn more about HNAP, email: cdcp-programmgr@cisco.com

