Network Working Group
Request for Comments: 2322
Category: Informational

K. van den Hout
HvU/HIP-networkteam
A. Koopal
UUnet NL/HIP-networkteam
R. van Mook
University of Twente/HIP-networkteam

1 April 1998

Management of IP numbers by peg-dhcp

Status of this Memo

Copyright Notice

Introduction

This RFC describes a protocol to dynamically hand out ip-numbers on field networks and small events that don't necessarily have a clear organisational body.

It can also provide some fixed additional fields global for all clients like netmask and even autoproxyconfigs. It does not depend on a particular ip-stack.

History of the protocol.

The practice of using pegs for assigning IP-numbers was first used at the HIP event (http://www.hip97.nl/). HIP stands for Hacking In Progress, a large three-day event where more then a thousand hackers from all over the world gathered. This event needed to have a TCP/IP lan with an Internet connection. Visitors and participants of the HIP could bring along computers and hook them up to the HIP network.

During preparations for the HIP event we ran into the problem of how to assign IP-numbers on such a large scale as was predicted for the event without running into troubles like assigning duplicate numbers or skipping numbers. Due to the variety of expected computers with associated IP stacks a software solution like a Unix DHCP server would probably not function for all cases and create unexpected technical problems.

So a way of centrally administrating IP-numbers and giving them out to people to use on their computers had to be devised. After some discussion, the idea came up of using wooden clothes-pegs. Using pegs has the following advantages in respect to other methods:

- cheap
- a peg is a 'token' and represents one IP-number, therefore making the status of the IP-number (allocated or not allocated) visible.
- a peg can be clipped to a network cable giving a very clear view of where a given IP-number is in use.

Credits for the original idea of using wooden pegs go to Daniel Ockeloen.

The server.

The server can have many appearances. At HIP it was a large tent situated at the central field where all the activities were. It can also be a small table in the corner of a terminalroom.

The server can hand out two parts to the client, the peg and a paper with additional fields fixed for the site the server is running for. We will describe both here.

The peg.

On the peg the IP-number is mentioned. The text on the peg can be described according to the following BNF:

Total ::== IP | Net

IP ::== num.num.num.num | num.num | num

Net ::== num.num.num/mask | num.num/mask | num/mask

num ::== {1..255}

mask ::== {8..31}

The Net-method of writing larger nets is an optional part of the protocol, it doesn't have to be implemented. If it is implemented, it requires more administration at the server (see below).

The short versions of the IP-number with only 1 or 2 chunks are meant for large servers where writing the whole number on the peg is just boring and time-consuming. It requires the prefix to be mentioned on the additional field paper, but that can be produced in more

convenient ways. It is not recommended to work with more prefixes. It is better to write more numbers on the peg and use a smaller prefix.

If the network to be numbered is rather large and some kind of subnetting has to be implemented it is possible to give the pegs from the different subnets different colors. This has proven to be a very convenient way at HIP.

The additional vendorfield paper.

This part is meant for information that is fixed for the whole site. It can either be implemented as small printed notes handed out with the peg or as a large paper billboard hung at a convenient place where everybody can read it.

The information can be described with the following BNF:

Network ::== num.num.num.num

Netmask ::== num.num.num.num | num

Gateway ::== num.num.num.num | num.num | num

Proxy ::== num.num.num.num:port | num.num:port | num:port

Paper ::== Network Netmask Gateway Proxy | Network Netmask Gateway

num ::== {0..255}

port ::== {1..65535}

The paper and the peg are of course one part, if two numbers are used on the peg, two numbers are used on the paper.

Because it is fixed information, it can be produced with means of mass-production (printing, copying).

The IP-repository

Due to the nature of the peg, the repository can be quite simple. Just a clothes-line with all the pegs that are ready to be handed out attached to it. If you work with different subnets, it is convenient to group the pegs for the different subnets (colors).

At large networks where it is not really known how many IP-numbers are needed, a first set of pegs can be made in advance, and the administration of produced pegs kept on paper so it is known for which numbers pegs have already been made. If use is made of the

net-extension on the pegs, numbers given out that way can be administrated this way too.

Issuing IP-numbers.

The pegs and the IP-numbers are issued at the server to the client. Normally the client has to visit the server personally. Depending on how secure and controlled you want the process, the client has to ask for a peg to a responsible person, or he or she can just get a peg from store himself.

If someone could apply for a networkrange, and he net-extension isn't used, coat-hangers can be prepared with sets of pegs attached to them.

The vendorfields paper doesn't have to be issued with every peg, it is only needed when wanted.

Reclaiming and reusing IP-numbers.

It is not easy to implement a TTL in this protocol. One obvious TTL is the duration of the event after which the IP-numbers are not valid anymore.

However, if a client decides that it doesn't need an IP-number anymore it can bring the peg back to the server.

The server should at that point decide what to do, if desired, it can bring the peg back into the pool (attach it to the clothes-line again).

If the server is not manned (the client has to help themselves), the only thing possible is that the client just places the peg back into the pool.

The client side.

The optimum location for the peg is clipped to the network cable near the NIC of the device needing an IP-number allocated. This ensures a clear visual connection between the device and the IP-number allocated and makes it an easy task to see which IP-number is allocated.

Transfer of the IP information from the peg and the additional vendorfield paper note to the settings in the IP stack is done by human transfer. A person reads the information from the peg and from the additional information and enters this in the configuration of the used IP stack. This transfer is not completely free of

corruption of the information or loss of the information contained on the peg.

A certain amount of knowledge of the logic of IP settings is also assumed on the part of the person transferring the information.

Other information on the vendorfield paper note has to be transferred to the settings within specific application programs.

Use with other protocols

This protocol could be combined with avian carriers as described in RFC 1149 to hand out IP-numbers remote.

At the first avian carrier, the peg is clipped to the leg of the carrier after rolling the additional vendorfield paper around it.

The remote site can take the peg on arrival of the avian carrier and use the information on it.

This part of the protocol is still experimental and requires some additional research on topics like the weight of the peg and loss of the peg/whole carrier.

Security Considerations

Some remarks about security can be made.

Pegs are small devices and can be lost. At that time, the IP-number which was lost can't be used anymore because someone else can find the peg and use the information stored on it. But, once the peg is attached to a network cable, the chance to loose the peg is minimized.

All the information on both the peg and on the additional 'fixed' fields on the paper record are plain text and readable for everyone. Private information should not be exchanged through this protocol.

On the client side all sorts of clients exist and cooperate freely. Due to the human factor of the clients transferring information from peg to IP stack, the information can be misinterpreted, which could cause network troubles. In the field test at HIP this became perfectly clear when someone mixed up the numbers and used the address from the default router as his IP-number, rendering the network useless for a period of time.

Authors' Addresses

Koos van den Hout
Hogeschool van Utrecht / Expertisecentrum Cetis
P.O. box 85029
3508 AA Utrecht
The Netherlands

Phone: +31-30-2586287
Fax: +31-30-2586292
EMail: koos@cetis.hvu.nl

Andre Koopal
UUnet Netherlands
P.O. box 12954
1100 AZ AMSTERDAM
The Netherlands

Phone: +31-20-4952727
Fax: +31-20-4952737
EMail: andre@NL.net

Remco van Mook
Van Mook Consulting
Calslaan 10-31
7522 MA Enschede
The Netherlands

Phone: +31-53-4895267
EMail: remco@sateh.com

Full Copyright Statement