Knowledge Managemenent in the IPv6 Migration Process

Krzysztof Nowicki, Mariusz Stankiewicz, Aniela Mrugalska,
Tomasz Mrugalski, and Jozef Wozniak

Gdansk University of Technology
Faculty of Electronics, Telecommunications and Informatics
11/12 Gabriela Narutowicza Street, 80-233 Gdańsk, Poland
{krzysztof.nowicki,mariusz.stankiewicz,aniela.mrugalska
tomasz.mrugalski,jozef.wozniak}@eti.pg.gda.pl
http://www.eti.pg.gda.pl/

Abstract. There are many reasons to deploy IPv6 protocol with IPv4 address space depletion being the most obvious. Unfortunately, migration to IPv6 protocol seems slower than anticipated. To improve pace of the IPv6 deployment, authors of the article developed an application that supports the migration process. Its main purpose is to help less experienced network administrators to facilitate the migration process with a particular target usage being SOHO networks.

Migration to IPv6 is a complex process. For the sake of simplicity, it has been split into several phases. First one is dedicated to automated network discovery that helps to gain initial knowledge about migrated networks and its services. Second step is devoted to building knowledge database about used hardware, operating systems, services and applications. It is being represented as a collection of interdependent questions with possible answers. This knowledge database is then used to provide relevant assistance for the IPv6 migration steps.

To reduce the tree structure management costs and choose only applicable elements that actually exist in migrated network, there are two proposed algorithms to create a list of questions and advices. Developed methods enable quick and easy expansion options available without any knowledge about all existing questions in base and their dependencies.

This paper presents a prototype application that is still under development. It will allow to add elements to IPv6 Migration Guide in an easy way. Used together with already developed IPv6 Migration Guide solution, it will provide significant assistance to network administrators during IPv6 enabling phase.

Key words: IPv6, Separable Algorithm, Dripping Algorithm, IPv6 Migration Guide, IPv6, multi-platform Questions Editor

1 Introduction

Both IPv4 and IPv6 address spaces are governed by Internet Assigned Number Authority (IANA) [1]. It allocates address blocks (an /8 class of IPv4 addresses)

to five Regional Internet Registries (RIRs) that in turn delegate addresses to network operators. On February 3rd, 2011 IANA allocates last of the available IPv4 addresses to RIRs. Although RIRs supplies are expected to last for several months, there are no longer new addresses to replenish them. Each RIR is responsible for certain world region. Depending on the demand, some regions may run out of IPv4 addresses sooner than others. Depending on used prediction algorithm, the estimated date of first RIR will run out of IPv4 addresses is expected to happen on August 14th, 2011. The problem of the depletion of public IPv4 addresses is a very serious threat both to all operators and Internet service providers and individual users. There is a need to migrate to IPv6. This fact is confirmed by all the organizations managing the assignment of IP addresses such as IANA [1] and the RIPE.

There are many initiatives fastering the implementation of IPv6. For instance the services available via IPv6 content providers such as Google [2] and YouTube [3]. Many operators started to offer access via IPv6, such as Comcast (USA) [4], Free.fr (France) [5] or the Polish Jarsat [6]. It also should be mentioned about the forthcoming Word IPv6 Day event, when many content providers will be available include IPv6 in their services. Participation in this event scheduled for July 8, 2011 announced Facebook, Google [2], Yahoo, Akamai, Cisco, gazeta.pl and many others. Implementation of new IPv6 addresses allows a large number of telecommunications equipment allows to be visible on the Internet.

Devices, such as personal computers, small home servers, cameras, alarm systems and telephones, will be able to communicate directly and without interruption, regardless of their location. This method of communication called end-to-end, was lost in IPv4 when networks started to use Network Address Translation (NAT) solutions [7].

IPv6 provides faster transmission of packets resulting in better performance and less load in the network routers. There was also introduced QoS (Quality of Service), more expanded mechanisms for autoconfiguration or the support for IPsec [8] in the same protocol. Without the development of the IPv6 Internet technologies in the longer period of time will be impossible. We should start to implement it now to avoid higher costs, which involves adapting the network to support IPng at the last moment [9].

2   System architecture

Migration of existing network which uses IPv4 to IPv6 is a complicated process [12]. In order to facilitate the transition through the process of changing the protocol system the IPv6 Migration Guide was developed. Guide's main goal is to help administrators to facilitate the implementation with a particular focus on SOHO networks.

There are planned two methods of a software distribution - as an application installed on Windows systems and as a component of an automatic bootable CD. The IPv6 Migration Guide will be one of the fundamental elements of the LiveCD, which being prepared by the Future Internet Engineering [10]. Devel-
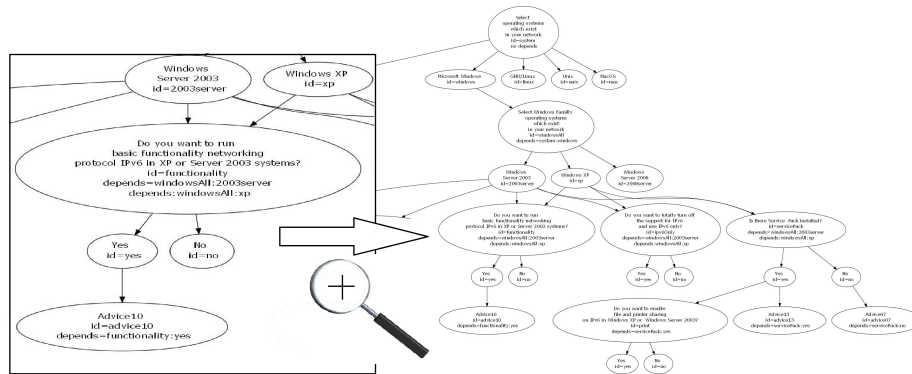
Fig. 1. $< depends >$ parameters can be used to construct dependency tree. The figure shows an example subset a such tree with questions, advices and answers visualized.

oped solution is also in addition to the IPv6 Migration Guide, going to offer a number of validation tools that are used to confirm the correctness of the migration to IPv6[11]. LiveCD will works under the supervision FreeBSD.

Working of IPv6 Migration Guide can be divided into four basic processes. The first step includes the detection and analysis of networks, based on external scripts. In the second step it is possible to verify the information from the first step, and to provide additional answers that can not be achieved through automated network scanning. Particular questions are related to each other to not display the next question, when previous answers indicate that they are not needed in a given situation. The hierarchy of questions with dynamic dependences and well state of information about network is a base element of principal knowledge. Step three provides the necessary knowledge needed to make the real migration process. This is based on data acquired on the network from the previous action. A package of tips can be treated as assurance of knowledge accumulated in previous step. The fourth step provides opportunity to test the network settings in odder to ensure that the required steps have been made by an administrator correctly. This is the last step.

Due to the complexity of the process, the amount of data collected and processed in the second and third steps is very large. It became necessary to create a solution that provides the ability to limit viewing tips matched to the topology, technology and services available in a particular network migrated.

3   Questions and advices constructions

All questions and advices are stored in separated XML tags. One of the main attributes is relation called $< depends >$, which defines the relationship between the elements. Inside questions there are answers embedded, which have a unique name id. Because of the uniqueness of this tag it is possible to determine the

answer to the question which is clearly defined. Example relationships will be presented in Fig.1.

Building relationships was presented in the example below, where an affirmative answer to the first obvious questions about name id WinXP reads "Do you have Windows XP Professional in your network?" brings up another question that reads "Do you want to turn off all support for IPv4 and use only IPv6 in Windows XP Professional?" [13] It was made by developing the function $< depends >$, which for these questions may take the following form WinXP:yes or WinXP:no, where in the first case it could display the question with id=ipv6only while in the second case its elimination.

```
<question type="single" id="WinXP" depends="system:windows">
  <content  lang="en">Do you have Windows XP Professional
      in your network?
  </content>
  <answer id="yes" lang="en">yes</answer>
  <answer id="no" lang="en">no</answer>
  <helpHtml>winVer.htm</helpHtml>
</question>

<question type="single" id="ipv6only" depends="WinXP:yes">
  <content  lang="en">Do you want to turn off all support for IPv4
      and use only IPv6 in Windows XP Professional?
  </content>
  <answer id="yes" lang="en">yes</answer>
  <answer id="no" lang="en">no</answer>
  <helpHtml>ipv6only.html</helpHtml>
</question>
```

A similar relationship exists with the advices that are directly related to the answers. There is also inbuilt an elimination procedure of advices that are unrelated with migrated network. In this way we avoid a situation where the administrator receives advices, which he could not use, because of the lack of understanding of their purpose and meaning.

The following example illustrates the mechanism of actions undertaken in relation to the advice of migration. For the question with id=nat and content "Is NAT used in your network?" [7] there are possible answers yes or no, which will mean selecting the appropriate response with id yes/no. Depending on the answers selected by the user there will be displayed an advice from the file natAnswYes.htm, if the relationship was nat:yes because of an affirmative answer, or advice from the file natAnswNo.htm if the answer was negative.

```
<question type="single" id="nat" depends="system:linux">
  <content  lang="en">Is NAT used in your network?
  </content>
  <answer id="yes" lang="en">yes</answer>
  <answer id="no" lang="en">no</answer>
  <helpHtml>nat.htm</helpHtml>
```

```
</question>

<step lang="en" depends="nat:yes">
   <helpHtml>natAnswYes.htm</helpHtml>
</step>

<step lang="en" depends="nat:no">
   <helpHtml>natAnswNo.htm</helpHtml>
</step>
```

4   Proposed solutions algorithms

To show a simple and transparent structure of the network it was decided to use the form of a tree to illustrate its components, where the next elements in the network have the form of branches. This form is presented in Fig.2.

Branch delves more with details increasing in questions. The application is designed to help a users selecting the tree items that are in his network. This solution is easy to understand by a large group of administrators who can find an analogy to the tree structure which is used in popular operating systems.
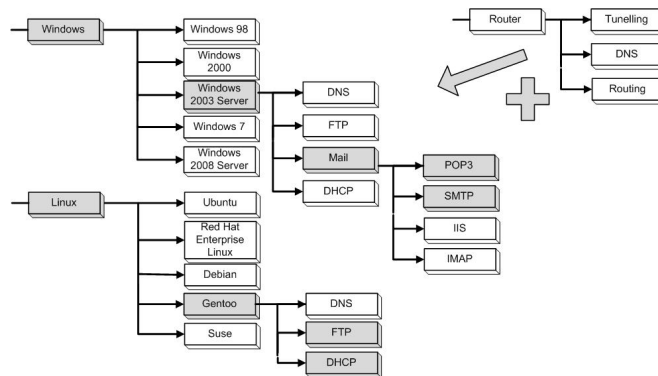
Fig. 2. Structure of the network was decided to use the form of a tree to illustrate its components. This solution is easy to understand by a large crowd of administrators who can find an analogy to the tree structure which is used in popular operating systems.

Additionally this solution provides an easy way to add new questions and advices. There is no need to view trough all the questions. We can simply search key words and display only these branches in which we found our key words. Then the user, who wants to add the new question can easily check place in the display industry where he wants to add it. It is importance to add such possibility when we have a few hundred questions and advices. Otherwise, user who wants to add a new element must read all XML file and search relationships between the asked questions to find the physical location for the advice.

In order to reduce the tree of objects and select the only elements presented in the network that would launch the application there were proposed two algorithms solutions that allow to create a list of questions and advices in order to manage the displayed informations.

The principle of the first one, Dripping Algorithm with a single set, is creating a list of all the questions, where each question is marked active or inactive flag, which would find its counter part in the tag $< question >$ with a flag="active" or flag="inactive". When the second step of IPv6 Migration Guide starts only some of the questions have active flag.

Every time you give answer the application makes a search of tree branch to flags searching that should be changed so that after each of the reply is possible to change the displayed number of questions. It should also consider the case where a question depends on many others. In this case, verification is required for each question separately.

The Separable Algorithm is based on creation of two separate tables which are both for the questions and answers. At first, the reactive elements are displayed to the user while the other contains inactive elements which are at the time omitted. During running the program and loading a list of questions it follows their initial sorting.

Questions which always have to be raised up and do not have a $< depends >$ tag go directly to the first (active) marker. The other questions, which have a marker go to another.

In allocating the appropriate sets of questions are followed by viewing the whole tree response. If we find the relationship back into the branch that matches the current tag is added to the dependence of the previous questions to the currently viewed. Following example explains this concept:

The dependence of the analyzed questions takes the following form:

```
<question type="single" id="functionality" depends="windowsAll:xp">(1)
```

There is dependence from the question:

```
<question type="multiple" id="windowsAll" depends="system:windows">(2)
```

And also from the answer:

```
<answer id="xp" lang="en">
```

In turn this question depends on the another:

```
<question lang="pl" type="multiple" id="system" >(3)
```

And the answer:

```
<answer id="xp" lang="pl">Windows XP</answer>
```

Summarizing, question (1) depends on question (2) directly and indirectly from the question (3). Therefore, depending on this tree looks like this:

```
system:windows -> windowsAll:xp -> functionality
```

Because of this the tag $< depends >$, in a currently browsing answer, will be automatically added a flag from the previous answer. In this case it will be the entire path that describes a particular branch of the dependent response. Added tag $< depends >$ will take the following form:

```
system:windows, windowsAll:xp
```

This solution gives a guarantee that if we answer to all questions in the branch and return to the first question to change our answer, all of the questions depends on our answer in this branch will be transfered to the inactive set of questions. For example if we consider that did not have in our network any of Windows systems, all of the questions depends on this system will be transferred to the inactive set of questions, because all of these questions can not be found dependency system:Windows.

The Separable Algorithm with two sets is also used for management of advices displayed for the user. Its flowchart is presented in Fig.3.

The Separable Algorithm
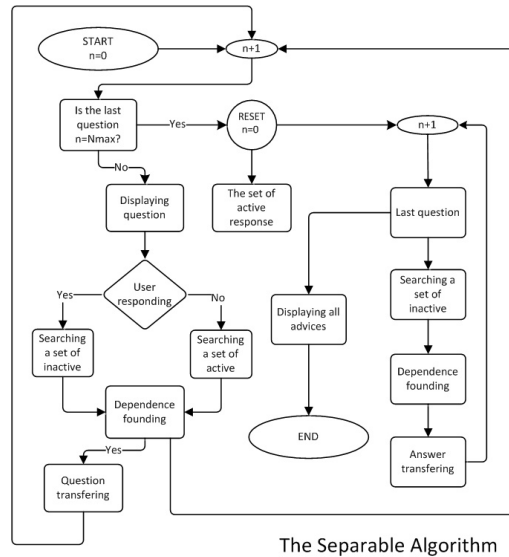
Fig. 3. Flowchart of the Separable Algorithm with two sets used for management of advices displayed for user.

To the end of the process of giving answers there is parallel searching in all sets of advices and transferring relevant advices to the appropriate set.

Presorting of questions starts when the application is being started. Those that have tag $< depends >$ are allocated to set with the inactive questions, and those that do not have a tag are allocated to the set with the active questions.

While starting the system there is an initialized counter N and a determined value of the N_{max} that equals the number of questions in a set of active questions.

Then N is increased by 1 and this algorithm checked to see if $N \leq N_{max}$ and the question with the number N is displayed to the user. After answering the following set of relevant searches. If the relationship is not found in the set, there is return to the the beginning and the increment of N. If the relationship is found, that question shall be transferred to the active set of questions and N_{max} is recalculated to succeed increment N and displayed next question. If $N == N_{max}$ there is progress to the next step.

In first order N is set to 0,then the discussed algorithm moves moved all the answers to inactive set and the increment of N.

There is a created tag $< depends >$ with id from question N and with selected answer to make possible search a set with the inactive answers to find a newly created question. If the tag is found the answer is moving to a new set and we increase N and check if $N \leq N_{max}$. If so, we take a new question with the number N and repeat the operation until the dependencies $N \leq N_{max}$ is true. Algorithm describes the figure below.

Between the two sets there are maintained relationships similar to the Dripping Algorithm. As a result of the answers given by the administrator, questions are moved between the sets. As a result of the above mentioned algorithm, user receives advices which are related only to his network.

5   Research and decision making – process selection

During the initial tests in the laboratories authors was able to demonstrate that the Separable Algorithm because of the two sets of turns out to be more efficient solution in the aspect of the displaying speed. Currently, questions or suggestions transfer between two separate arrays, resulting in faster viewing them in looking for a relationship. There is no need to browse all the available questions and answers, check their flags and dependencies.

We are searching for inactive set for questions dependent on the affirmative answer, or active set in the opposite case. Using the two sets is also reflected in the speed of counting number of questions available to the user at a time. To do this, simply use the simple features built in the standard Python language [15], which was used in described solutions. Python has been used also in IPv6 Migration Guide. Past tests show the correctness of his actions[14].

Optimization activities are planned to increase, in order to improve the efficiency of the algorithm.

6   Development plans – extension of the contributors circle

The development plans of IPv6 Migration Guide predict the further development of process of making questions and advices related to migration which involves taking many relationships between contained therein essence.

With development of the available knowledge base, tracking and making correct definition of a new relationships may become problematic. Technical
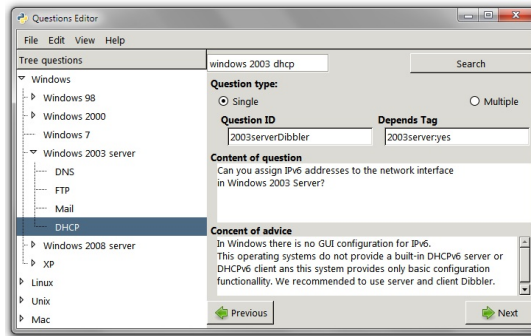
Fig. 4. Main window a prototype of a multi-platform Questions Editor. [16]

University of Gdansk is developing a prototype of a multi-platform Questions Editor presented in Fig.4 with main objective to facilitate the definition of new and manage existing migration questions and advices.

It will be compatible with the solutions used in the IPv6 Migration Guide, and therefore will support the creation of the knowledge base based on XML files, handle the dependency tree of scenarios and many others.

The Questions Editor's idea is to be an intuitive application that Allows to add elements to IPv6 in easy way for every person who did not participate in its creation process.

Because of using the questions structure with analogy to the tree structure used in popular operating systems, when adding an element it is possible to quickly scan all the all the branches that match the newly designed advice, attaches the newly created questions for branch and setting a unique id or checking tag $< depends >$ and many others.

7   Conclusions

Exhaustion of available IPv4 addresses is a serious issue for Internet expansion. This threat is reported by all organizations dedicated to the IP addresses management, such as IANA and RIPE [1]. The only viable long term solution is a migration to next generation of IP. To ease such migration, authors proposed a *IPv6 Migration Guide* system that assists the migration process and helps network administrators to facilitate the IPv6 deployment. Although the primary scope are SOHO networks, there are plans to expand knowledge base to other network types.

Two algorithms were proposed to create a lists of questions and advices to manage required information. Methods presented by authors enable quick and easy options expansion available without knowledge about all existing questions in base and their dependencies.

Described applications are designed to help user in migration to IPv6. To make this process easier representation structure of the network was presented

in analogy to the file system used in popular operating systems. Further development plans are focused on optimization Separable Algorithm and creating specialized advices to allow use IPv6 Migration Guide in a wider aspect than originally planned.

References

1. IANA's IPv4 Address Report "The Official Website of IANA's IPv4 Address Report", `http://ipv4.potaroo.netl`, retrieved on Feb. 12, 2011
2. Access Google services over IPv6 "The Official Website of Google over IPv6", `http://www.google.com/intl/en/ipv6/l`, retrieved on Jan. 2011
3. C. Marsan "YouTube support of IPv6 seen in dramatic traffic spike", `http://www.networkworld.com/news/2010/020110-youtube-ipv6.html`, Network World, retrieved on Feb. 11, 2011
4. Comcast "The Official Website of Comcast IPv6", `http://www.comcast6.net/`, retrieved on Dec. 2010
5. French ISP Free.fr IPv6 "Free ISP on Wikipedia", `http://en.wikipedia.org/wiki/Free_%28ISP%29#Internet_accessl`, retrieved on Oct.2010
6. Jarsat IPv6 "The Official Website of Jarsat", `http://ipv6.jarsat.pl/l`, retrieved on Jan. 2011
7. K. Egevang "The IP Network Address Translator (NAT)", RFC 1631, IETF, May 1994
8. S. Kent, K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, IETF, Dec.2005
9. Niall Richard Murphy, David Malone "IPv6 Network Administration", O'Reilly Media, 2005
10. Future Internet Engineering "The Official Website of Future Internet Engineering", `http://iip.net.pl`, retrieved on Jan. 2011
11. S. Deering, R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, IETF, Dec.1998
12. T. Mrugalski, "Metoda szybkiego wdrazania IPv6 przy wykorzystaniu DHCPv6", Techonologie Informacyjne conference, Gdansk, 2004
13. Microsoft TechNet, "Microsoft Internet Protocol Version 6 (IPv6)", `http://technet.microsoft.com/en-us/network/bb530961.aspx`, retrieved Oct. 2009
14. L. Gajek, M. Kaluszka, "Wnioskowanie statystyczne", Wydawnictwa Naukowo-Techniczne, Warsaw,2000
15. Mark Lutz "Programming Python. Powerful Object-Oriented Programming", O'Reilly Media, December 2010
16. H. Schulzrinne, B. Volz "Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiation Protocol (SIP) Servers", RFC3319, IETF, July 2003

Acknowledgment