Automatic Configuration of IP Networks and Routers

Kenji Hori, Kiyohito Yoshihara, and Hiroki Horiuchi

KDDI R&D Laboratories Inc.
2-1-15 Ohara Kamifukuoka-Shi Saitama
356-8502, Japan
TEL: +81 49 278 7651  FAX: +81 49 278 7510
{hori, yosshy, hr-horiuchi}@kddilabs.jp

Abstract
The ability to configure IP networks automatically would be advantageous in particular to the case where we cannot expect the user to be highly skilled like a network operator, typically in a home or SOHO (Small Office Home Office) environment. It would be labor-saving if we could use network devices such as wireless LAN access points for public access or residential routers without manual configuration, since they are being used in various places, and do not need detailed configuration in ordinary operation. Deployment of such auto-configurable networks and devices would be cost-effective in terms of efficient network operation and management. Some studies have investigated automatic configuration of IP networks and routers. As one of the most promising protocols, the IETF (Internet Engineering Task Force) Zerouter group has just begun the development and standardization of ARCP (Automatic Router Configuration Protocol). ARCP is based on a centralized architecture, in which a server called an ARCP server detects addition or deletion of routers, and adapts router settings to network changes, in conjunction with an ordinary DHCP (Dynamic Host Configuration Protocol) server. The current status of ARCP; however, is too immature to implement, and requires additional elaboration before deployment in an operational network. In this paper, in order for ARCP to work on an operational network, we first present three requirements for automatic IP network configuration protocols, through a comparative study with other existing proposals. After identifying ARCP as a potential candidate, we address four problems of the current immature ARCP, and propose a new extended version of ARCP, providing solutions to each problem. Some works in progress are described, including the development of a simulator for evaluating the proposed solutions, and preliminary simulation results are presented.

Keywords
automatic configuration, ARCP, DHCP, Zerouter

Contact Person
Kenji Hori(hori@kddilabs.jp)
Introduction

- Techniques for automatic configuration of IP networks and routers are desirable
  - The presence of a highly skilled user like a network operator cannot be assumed.
  - The deployment of automatically configurable network would be cost-effective in terms of efficient network operation and management.

- Requirements for automatic configuration are identified, comparing existing proposals

- ARCP is promising; however, the current version has problems to be solved:
  - An address management solution is not provided explicitly.
  - A DHCP server cannot provide an appropriate address of the default gateway.
  - Operations for discovering addition/deletion of routers (“Peer Discovery”) and for adapting router settings to network changes (“Renumbering”) can cause unstable repetitive changes of address.
  - “Renumbering” operations can evaporate the addresses of default gateways.

- In this paper:
  - We identify requirements for an automatic configuration protocol.
  - We compare existing proposals with our requirements, showing that ARCP is the closest to satisfy our requirements.
  - We address the above problems in detail, and propose solutions.
  - We describe works in progress and show preliminary simulation results.

Introduction

Techniques for automatic configuration of IP networks and routers are desirable because we cannot necessarily expect the user to be highly skilled like a network operator in a home or SOHO environment and the deployment of an automatic configurable network would be cost-effective in terms of efficient network operation and management. Our research goal is to develop a practical automatic configuration method for IP networks and routers in such environments.

In order to develop a method for automatic configuration, it is first necessary to identify requirements for the automatic configuration under the circumstance described above. There are some study efforts for such automatic configuration as [1, 2, 3, 4, 5, 6, 7, 8]. As one of the most promising proposals, the IETF Zerouter[9] group has just begun to develop and standardize ARCP (Automatic Router Configuration Protocol). ARCP is based on a centralized architecture, in which a server called the ARCP server manages addition and delete of routers, new connections and disconnections, and adapts router settings to network changes, with an ordinary DHCP (Dynamic Host Configuration Protocol) server. However, the current status of ARCP is too immature to implement, and the following problems should be solved before deployment in an operational network:

Problem (1): An address management solution is not provided explicitly.
Problem (2): A DHCP server cannot provide an appropriate address of a default gateway to routers, because such an address is unknown until the routers are configured.
Problem (3): An operation called “Peer Discovery” for detecting addition and deletion of routers and an operation called “Renumbering” for adapting router settings to network changes can cause unstable repetitive addresses changes, due to each operation working independently.
Problem (4): The “Renumbering” operation can evaporate the addresses of default gateways of routers, because the routers do not know if it is used by other routers as the default gateway.

In this paper, we first present requirements for the automatic configuration, and compare existing proposals with the requirements. After identifying ARCP as a potential candidate, we briefly explaining ARCP. Then, we address the above problems in detail arising from the current incompleteness of ARCP, and propose solutions to these problems. Some works in progress to evaluate the proposed solutions by simulation and preliminary results are shown. Finally, we conclude with future work.
Requirements and comparison of existing proposals

### Requirements

In the following discussion, we assume home or SOHO networks, in which a highly skilled user cannot be assumed. Automatic configuration of IP networks and routers means management by external operators with management systems for helping a less-skilled user use his/her network device such as access router.

In order for such networks to be configured automatically and to gain Internet access, an automatic configuration protocol should provide:

- **R1**: a method to configure and modify router settings by the external management system typically in ISP (Internet Service Provider) instead of unskilled users, because at least one address prefix (prefix hereafter) is needed. Settings of routing protocol such as OSPF (Open Shortest Path First) may also be needed.

- **R2**: a method for automatic decision and assignment of addresses and prefixes, under arbitrary network topology. It should also follow changes in topology. This is required because the unskilled users may not care about the topology of their network, and might suddenly connect or disconnect routers providing Ethernet LAN and an access point for wireless LAN facilities, for example.

- **R3**: support for both IPv4 and IPv6. This is required because IPv6 brings larger address space than IPv4, and we may only need to assign a prefix by SAA (Stateless Address Auto configuration). On the other hand, there still exist some applications unable to work with IPv6, and an automatic configuration protocol should count such applications as well.

### Comparison of existing proposals

Table 1 shows a brief comparison of existing proposals.

#### Table 1. Comparison of existing proposals

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Automatic Router Configuration Protocol (ARCP)</strong> [1][2]</td>
<td>Configuration settings are managed by external management systems.</td>
<td>Prefixes and addresses are assigned under any topology. Automatic decision of an appropriate prefix length is not provided.</td>
<td>Supports IPv4 and IPv6.</td>
</tr>
<tr>
<td>Multilink-subnets [1]</td>
<td>Not provided. There is no need to be managed, if they use the link-local prefix.</td>
<td>Assignment of prefixes and addresses is not required, because it assumes a single subnet. Current support is only star topology.</td>
<td>Supports IPv6 only.</td>
</tr>
<tr>
<td>Prefix Delegation (PD) methods</td>
<td>Provided, but limited. They provide methods for prefix delegation only.</td>
<td>[6][7][8] do not provide automatic decision of prefixes and address length for a subnet. [6] can assign prefixes, and may work under any topology if used with DHCP relay agents. [7] and [8] work only under a single subnet. Automatic decision of an appropriate prefix length is not provided.</td>
<td>[6][7][8] supports only IPv6.</td>
</tr>
<tr>
<td>Autonomous Prefix and Address Assignment (APAA) methods</td>
<td>Not provided.</td>
<td>Prefixes and addresses are assigned under any topology. A presetted prefix length is applied to all subnets.</td>
<td>[4] supports IPv4 and IPv6. [6][7][8] supports only IPv6.</td>
</tr>
</tbody>
</table>

The above comparisons show that the ARCP is the closest proposal to satisfy the requirements. In the next two pages, we present an overview of ARCP.
Overview of ARCP (1)

Figure 1 shows basic operation of ARCP. The operation is divided into four phases: an ARCP capable router (1) first obtains host setting from a DHCP server, (2) next obtains router settings from an ARCP server, (3) acts as a DHCP relay agent for other router configuration, and (4) sends a dummy BOOTREQUEST message for detection of addition and deletion of other routers. The operation is described in more detail below.

1. Obtaining basic IP settings as a host from DHCP server
   In Fig.1 from (1) to (4), the ARCP router 1 behaves as a DHCP client to obtain an address and its associated information for the interface 1 (IF1 hereafter) from the DHCP server. Different from normal DHCP messages, they contain a newly defined DHCP option for ARCP, called “ARCP server IP address” carrying addresses of ARCP servers.

2. Obtaining complete settings as a router from the ARCP server
   From (5) to (6), ARCP router 1 opens a TCP connection to the ARCP server. First, an ARCPREG message, containing information such as hardware addresses of all interfaces, is sent to register itself with the router hardware database in the ARCP server for management of the configuration status of the routers. In response to the ARCPREG message, the ARCP server replies with an ARCP CFG message, containing information such as the address of IF2 and initial IGP (Interior Gateway Protocol) settings. Then, ARCP router 1 sets the address to IF2, and starts an IGP process such as a single area OSPF. This completes the configuration of ARCP router 1, and the router becomes operational.

3. Acting as a DHCP relay agent for other routers
   From (8) to (15), an already configured router, ARCP router 1, performs as an ordinary DHCP relay agent, for the configuration of downstream routers. By using the DHCP relay agent, downstream ARCP routers can indirectly communicate with the DHCP server. When a DHCPDISCOVER message is sent from ARCP router 2, this is relayed to the DHCP server by way of ARCP router 1. It specifies the address of IF2 of ARCP router 1 in the giaddr field of the DHCPDISCOVER message (the interface where the DHCPDISCOVER message was received), so that the DHCP server can distinguish which subnet this message comes from.

4. Sending “dummy BOOTREQUEST” messages
   Periodically, configured ARCP routers broadcast “dummy BOOTREQUEST” messages from each interface ((7), (16)). The address of the sending interface is specified in their ciaddr (Client’s IP address field), so that the ARCP server can discover other interfaces on the same link in the subsequent operation described in Fig 2 on the next page. Dummy BOOTREQUEST messages are called “dummy” because the sending router will never accept any configuration offers in reply to such messages - the replies must be ignored.
Overview of ARCP (2)

There is a case where the interfaces on the same link are assigned inconsistent addresses on a different subnet, because the ARCP server assigns addresses regardless of connections between routers. For instance, as shown in Fig.2 (A), IF2 of ARCP router 1 and IF1 of ARCP router 2 may be assigned addresses on a different subnet, even though they are on the same link. In order to avoid these discrepancies, “Peer discovery” and “Renumbering” operations detect the discrepancy and renumber correctly (Fig.2 (B)). Their details are described below.

Peer discovery

Once an ARCP router reaches the status of Fig.1 (7) or (18), it sends dummy BOOTREQUEST messages periodically from each interface (such as ARCP router 2 in Fig.2 (A)) specifying the address of the sending interface in the ciaddr field (Fig.2 (1)). If an ARCP router received a dummy BOOTREQUEST, it forwards it to the DHCP and ARCP servers specifying the address of the receiving interface in the giaddr (Relay agent IP address) field (Fig.2 (2)). The ARCP server checks the values of the ciaddr and giaddr fields of the incoming dummy BOOTREQUEST message. By comparing the network portion of the values of ciaddr and giaddr fields, it can detect inconsistency of two interfaces on the same link (Fig.2 (C)). If they differ from each other, the interface sending the dummy BOOTREQUEST and the interface relaying it are assigned addresses on the different subnets in spite of on the same link. If not, these interfaces are assigned consistent addresses.

Renumbering

If such inconsistency is detected, the ARCP server sends an ARCP CFG message containing a request for renumbering one of the interfaces, to one of the ARCP routers (such as ARCP router 1 or ARCP router 2), which has an interface assigned an inconsistent address (Fig.2 (3)). By this request, the ARCP server changes the address assigned to IF2 of ARCP router 1 or IF1 of ARCP router 2. So, the addresses assigned can be altered by this operation, and used only for temporal usage. Even if there are two or more inconsistent interfaces, the ARCP server renumbers one interface at a time, and repeats it until all the inconsistency are resolved.

As we described, there is a study on the basic operation of ARCP. However, ARCP is in the first stage of the standardization, and current proposal is insufficient to use in real operation. We indicate the problems that need to be solved toward real operation, and propose solutions to them in the following four pages.
Problem (1): Address management is not explicitly provided.
Solution (1): All addresses are managed originally by a DHCP server.

There are two possible solutions for address management as shown below:

S1. An address space is divided into two ranges: one for a DHCP server and another for an ARCP server.

S2. All addresses are managed originally by a DHCP server, and an ARCP server obtains addresses from the DHCP server acting as a proxy DHCP client, when necessary.

S2 provides more efficient address utilization than S1, as shown below:

We suppose subnets \(1.1.1.0/29\), \(1.1.2.0/29\) (Fig.3 (A)) and a network as shown in Fig.3 and Fig.4, where Link (B) (both in Fig.3 and Fig.4) has six interfaces on the same link.

In case of S1(Fig.3), address space is divided into two ranges: one for the DHCP server (Fig.3 (C)) and another for the ARCP server (Fig.3 (D)), and in order to accommodate all interfaces on the link (B) into a single subnet, at least six addresses should be pooled in both the ARCP and DHCP servers, summing up twelve addresses in total, since we cannot know how many addresses in a subnet used by DHCP or ARCP. For example, Fig.3 (E) shows the situation where the ARCP server needs the 4th address in \(1.1.1.0/29\) for the renumbering. However, this address is only available by DHCP. On the other hand, in case of S2(Fig.4), in order to accommodate all interfaces on the link (B) into one subnet, at least six addresses should be pooled only in the DHCP server (Fig.4 (A)).

Detailed example operation of S2 is shown in Fig.5. The ARCP server acts as a proxy DHCP client(Fig.5 (A)) to obtain an address for IF1 of ARCP router 2 from the DHCP server using giaddr of 1.1.1.1 (Fig.5 (2) to (5)). Next, it sends an ARCPCFG message (Fig.5 (6)) to change the IP address of IF1 of ARCP router2. ARCP router 2 sends a DHCPRELEASE message(Fig.5 (7)) to the DHCP server for releasing unused address of 1.1.2.1 (Fig.5 (B)).
Problem (2): DHCP server cannot provide addresses of default gateways.

Solution (2): ARCP routers replace values of DHCP “router” option field.

Possible solutions for Problem (2) are shown below:

S1. The DHCP server provides dynamically configured settings such as default gateway addresses created from the giaddr value of the DHCPDISCOVER message.

S2. The ARCP routers acting as DHCP relay agents replace the value of DHCP router option field in DHCP messages with the value of the giaddr field that was used by the DHCP relay agent.

We choose S2, because S2 requires no modification to the current DHCP server.

As an example of S1, we suppose such a network as shown in Fig.6. The value of the giaddr field is the address of the interface on which the DHCPDISCOVER message is received first (Fig.6 (1)), so it can be a default gateway of the sender of the DHCPDISCOVER message. If the DHCP server is able to provide a default gateway address configured from the value in the giaddr field of the DHCPDISCOVER message (Fig.6 (C)), then the server provides it to ARCP routers by specifying it in the DHCP “router” option field (Fig.6 (3)).

As an example of S2, Fig.7 (A) shows ARCP router 1 acting as a DHCP relay agent replaces the value of the DHCP “router” option field in DHCP messages with the value of the giaddr field that was used by the DHCP relay agent (Fig.7 (B)).
Problem (3): “Peer Discovery” and “Renumbering” operations can cause repetitive address changes.

Solution (3): The ARCP server presets static priority for each subnet.

Problem and Solution (3)
Problem (3): “Peer Discovery” and “Renumbering” operations can cause repetitive address changes.

As shown in Fig.8, the addresses assigned to ARCP routers can be renumbered repeatedly, when there are three or more interfaces on the same link. Fig.8 (A) shows three interfaces on the same link but each of them has a different prefix. If a dummy BOOTREQUEST message from IF2 of ARCP router 2 (Fig.8 (1)) reaches the ARCP server through IF2 of ARCP router 1 first, it invokes the Renumbering operation of IF2 of ARCP router 2 or IF2 of ARCP router 1. Now, suppose IF2 of ARCP router 2 is renumbered to 1.1.1.2 (Fig.8 (B)). Next, a dummy BOOTREQUEST message from IF2 of ARCP router 1 (Fig.8 (2)) reaches the ARCP server through IF1 of ARCP router 3. Suppose IF2 of ARCP router 1 renumbered to 1.1.3.2 (Fig.8 (C)). Again, a dummy BOOTREQUEST message from IF2 of ARCP router 1 (Fig.8 (3)) reaches the ARCP server through IF2 of ARCP router 2, and if IF2 of ARCP router 1 is renumbered to 1.1.1.1 (Fig.8 (D)), then this link returned to previous address assignment in Fig.8 (B).

Solution (3): The ARCP server presets static priority for each subnet.

This problem can be solved by giving a static priority to each pre-configured subnet, and setting this information to an ARCP server in advance. The ARCP server chooses an interface with the lowest priority for renumbering. The priority should be determined 1) automatically by an algorithm, for example numerical order of subnet prefixes, 2) randomly, or 3) manually if needed. The priority is never changed once in operation.

As an example of this solution, in case of Fig.9 (A), the ARCP server chooses IF2 of ARCP router 2 to be renumbered, since the preset priority (Fig.9 (I)) of {1.1.2.0/24} is lower than that of {1.1.1.0/24}. In the same manner, in case of Fig.9 (B), IF1 of ARCP router 3 is chosen. Thus, with the fewer number of renumberings than the case described in Fig.8, all interfaces of this link are accommodated into a single subnet without repetition (Fig.9 (C)).
Problem (4): “Renumbering” operations can evaporate default gateway address.

Solution (4): ARCP routers suspends actual Renumbering operation until downstream ARCP routers are configured.

Fig.10: Problem of evaporating default gateway address

Fig.11: Solution with prohibition of actual Renumbering operation

Problem and Solution (4)

Problem (4): “Renumbering” operations can evaporate default gateway address.

Depending on the progress of the configuration operation, the Renumbering operation can abort the configuration process for the downstream routers. For example, as shown in Fig.10, ARCP router 2 obtains an address and a default gateway (for IF2 of ARCP router 1) information from the DHCP server (Fig.10 (A)), and then begins to communicate with the ARCP server (Fig.10 (4) to (5)). At the same time, the ARCP server sends an ARCP CFG message to ARCP router 1 for renumbering IF2 (Fig.10 (B)). As a result, ARCP router 2 has lost a default route (Fig.10 (C)) and it cannot communicate with the ARCP server (Fig.10 (D)).

Solution(4): ARCP routers suspends actual Renumbering operation until downstream ARCP routers are configured.

This problem can be solved by suspending the ongoing Renumbering operation typically by changing IP address and suspending/resuming the DHCP relay agent and IGP, until the downstream routers with the renumbering interface are configured. The completion of the configuration of the downstream routers is detected by monitoring DHCPACK and dummy BOOTREQUEST messages they send.

An example of this solution is shown in Fig.11. With this solution, actual Renumbering operations are suspended until the downstream routers begin to send a dummy BOOTREQUEST message. To achieve this, ARCP routers should have Configuration status table of the downstream routers (Fig.11 (A), (B) and (E)). This table is initially empty (Fig.11 (A)). If ARCP router 1 relays a DHCPACK message (Fig.11 (1) and (2)) for IF1 of ARCP router 2, then the entry for this interface is registered as {DHCPACK relayed} (Fig.11 (B)). If ARCP router 1 requests renumbering operation (Fig.11 (C)) is suspended. If ARCP router 2 begins to send a dummy BOOTREQUEST message (Fig.11 (6)), then the entry of the table is updated as {dummy BOOTREQUEST relayed} (Fig.11 (D)). Now actual Renumbering operations are resumed (Fig.11 (E)).
Simulation example of ARCP

We are developing a simulator by OPNET[10] in order to evaluate the proposed solutions. Our work for this simulator is in progress; however, it can simulate basic operation of ARCP shown in Fig. 1 and Fig. 2, and the Static Priority proposal (the solution (3) as shown in Fig. 9) for the problem (3). Implementation of other proposed solutions are underway.

We show an simulation results of current simulator implementation which shows the effectiveness of the Static Priority for the problem (3). The network configuration in the simulation is shown in Fig.13. In this example network, 1 to 10 ARCP routers are connected in parallel. All ARCP routers have two interfaces. One interface of each router is connected to Hub 1 which is in turn, connected to the DHCP and the ARCP servers, and another is connected to Hub 2. The Hub 2 provides a segment called Link 1.

The current simulator implementation has some simulation parameters as shown in below:
- $N$: Number of ARCP routers connected.
- $T_d$: Period of sending the dummy BOOTREQUEST messages.

In this simulation, $T_d$ is fixed to 60 seconds and $N$ is set to between 1 and 10.

Figure 14 illustrates the completion time of accommodation of the Link 1 in cases of 1) the Static Priority is enabled and 2) disabled, and each pair of two bars shows a result for a different value $N$. Because fewer renumberings brings faster accommodation, so this measurement can be index of the effectiveness of the Static Priority which we illustrated in Fig. 9.

When fewer interfaces are on the link ($N <= 6$), results of 1) are almost equal to those of 2) except the case of $N = 4$. On the other hand, if relatively larger number of interfaces are on the link ($N >= 7$), results of 1) are always shorter than those of 2), and it is significant when $N >= 8$.

The effectiveness of the Static Priority may be spoiled by the current implementation. In the configuration of Fig. 13, some dummy BOOTREQUEST messages reach the ARCP server in a very short period, because the current ARCP router model does not wait a random time before the sending of the dummy BOOTREQUEST messages, although an ordinary DHCP client waits a random time before the sending of the DHCP messages to desynchronize. In addition, the current ARCP server model discards the dummy BOOTREQUEST messages which potentially invoke further renumbering to the interface under renumbering. So the Static Priority does not work as we have expected in Fig. 9, and the result can be equal to or worse than those of the case where it is disabled depending on the order of arrival of the messages.
Conclusions

In this paper:
- Requirements for automatic configuration of IP networks and routers are identified first.
- Comparative study on existing proposals is presented next, which shows that ARCP under standardization by IETF is the closest to satisfy requirements.
- We examine a basic operation of ARCP in detail.
- We address four problems and propose solutions.
- In order to evaluate proposed solutions, we are developing a simulator. The work is in progress, and we present preliminary simulation results.

Future work:
- Comparative simulation study of proposed solutions and standardized ARCP
- Development of:
  - Advanced decision method of subnet selection priority in “Renumbering” operation
  - Adaptive and dynamic subnet resizing (variable prefix length) method
  - Extension for operation on IPv6 and IPv6-IPv4 mixed networks.
  - Hybrid method combined with other proposals
- Implementation of proposed solutions and evaluation in operational networks

Conclusions

Conclusion
In this paper, for the purpose of developing an operational automatic configuration protocol for IP networks and routers in a home or SOHO environment, we propose a new extended version of ARCP, and show a simple simulation study results as the first experience with ARCP. ARCP is one of the promising automatic configuration protocols which Zerouter group in IETF has just begun to develop and standardize. We first identify requirements for the automatic configuration of IP networks and routers, based on the assumption that the automatic configuration implies operators with management systems help a less-skilled user use his/her network device in a home or SOHO environment. Comparative study on existing proposals is presented next, which shows that ARCP is the closest to satisfy requirements. After an overview of ARCP, we address four problems resulting from the current immaturity of ARCP, and provide solutions to each problem. Developing a simulator to evaluate the proposed solutions are now underway, and some preliminary results are presented.

Future work
Comparative simulation study of proposed solutions with the standardized ARCP is high-priority. In addition to the study, development of the following methods or extensions will be required:
1. Optimized decision method of subnet selection priority for IP renumbering method, which provides the least number of renumbering operations.
2. Adaptive and dynamic subnet resizing (variable prefix length) method, which eliminates the necessity of manual decision of prefix length, and reducing the unused addresses.
3. Extension for operation on IPv6 and v6-v4 mixed networks.
4. A Hybrid method combined with the other proposals.
Implementation of proposed solutions and evaluation in operational networks should also be preferred.

Acknowledgments
We are indebted to Mr. Tohru Asami, President, Chief Executive Officer of KDDI R&D Laboratories Inc., Dr. Yuichi Matsushima, Executive Vice President, and Dr. Takeshi Mizuike Executive Director for their continuous encouragement into this research.

This work is supported by Ministry of Public Management, Home Affairs, Posts and Telecommunications.
References


