MANET-AUTOCONF
Internet-Draft
Expires: April 20, 2006

C. Adjih
INRIA Rocquencourt, France
Pr. Mase
Information and Communication
Network Lab., Niigata University

Oct 17, 2005

Conflict Detection in MANET Autoconf
draft-adjih-manet-autoconf-detect-00

Status of this Memo

Copyright Notice

Abstract

Several wireless ad-hoc routing protocols have been and are being developped for MANET. However, autoconfiguration of MANET networks is still an unsettled area, and several methods have been proposed to perform such a task. One of the mecanisms that may be required for address autoconfiguration, is the detection of address conflicts. This is specially true for one of scenarios for MANET autoconf, the

case of the merge of MANET networks.

This document specifies a general protocol for the detecting address conflicts in a MANET network, and hence addresses a subset of the requirements of a full MANET address autoconfiguration solution. It is specified as an independent protocol from the MANET routing protocol, and the address configuration method, and may be used with any of them. It aims at conceptual simplicity: essentially, a tree of the nodes is built, from which all the information from all the existing nodes is known. Conflicts are detected by the node at root of the tree, or as inconsistent information on the root of the tree.

Table of Contents

1. Introduction 3
2. Problem statement 5
3. Terminology 6
4. Protocol Principles 7
5. Protocol Overview 8
5.1. Root Election 9
5.2. Parent Selection, Parent and Subtree Advertizement 9
5.3. Address Conflict Detection and Notification by the Root . 10
5.4. Root Address Conflict Detection and Notification by one Node 11
6. Detailed Specifications 12
6.1. Message format 12
7. Optimizations 15
8. IANA Considerations 16
9. Security Considerations 17
10. Acknowledgements 18
11. References 19
11.1. Normative References 19
11.2. Informative References 19
Authors' Addresses 21
Intellectual Property and Copyright Statements 22

1. Introduction

A mobile ad hoc network is a collection of nodes, which collaborate to each other without depending on centralized control for enabling wireless communication among nodes. When two nodes are within direct transmission range, they communicate directly (one hop wireless communication) ; and otherwise they communicate using other nodes as intermediary nodes (multihop wireless communication), where the intermediary nodes act as routers for forwarding IP datagrams. Accordingly, routing is a key problem for mobile ad hoc networks and many routing protocols have been proposed.

In IETF, in the MANET working group has specified the characteristics of such networks and the design guidelines for MANET routing protocols in [2]. In MANET, two proactive routing protocols, OLSR [4] and TBRPF [5], and two reactive routing protocols, AODV [3] and DSR [6] are or will progress to experimental RFC status; in addition, current work is focusing on defining standard track routing protocols with DYMO and OLSRv2.

In any case, these routing protocols assume that each node has been assigned an unique IP address on each of its network interfaces. IP address autoconfiguration is therefore an important practical issue: a problem statement of address autoconfiguration in MANET may be found in [8]. In the recent past, many autoconfiguration methods for various types of MANET networks have been proposed, and this prompted the proposal of a MANET-autoconf working group in the IETF.

Many conventional methods are organized independently from routing protocols so that they can be used for any MANET regardless of the routing protocols. Some other methods are intended to work jointly with the routing protocols to improve efficiency of IP address autoconfiguration and address conflict detection. For example, information about IP addresses in use can be collected with support of the routing protocol and can be used in selecting a new free addresses for a node seeking address allocation. A recent analysis of some MANET autoconfiguration techniques may be found in the survey [9].

In the current charter of the proposed MANET-Autoconf working group, the support of the merge of two or more ad-hoc networks is explictly included. More precisely, one of the specified goals is to "develop a mechanism to promote configured address uniqueness in the situation where different ad hoc networks merge". This document proposes a protocol for this situation. It aims being independent from the routing protocol and also being independent from the a MANET autoconf allocation algorithm. It is understood, of course, this mechanism and its implementation might be directly integrated with another

proposal, or with the functionning of the routing protocol.

The protocol itself is designed to detect address conflicts: one node collects the addresses of the other nodes in the network, and is able to detect whether or not, an address conflict exists. When an address conflict is detected, the nodes which are in conflict are notified. An overview of the protocol is given in section Section 5.

This document is structured as follows: the Section 2 highlights the problem statement for detection of address conflicts in merges. The section Section 4 the principles and the reasons behind those principles. The Section 6 gives some details of the specification of the protocol. Finally, the Section 7 gives an hint of some optimization and some similar protocols which have introduced some optimizations.

2. Problem statement

As mentionned previously, several autoconfiguration proposals exist as Internet Drafts specifying differents methods for MANET autoconfiguration (such as those reviewed [9]). However, in several of them, by choice, a specific scenario is not addressed: address conflict detection in the case where two or more ad-hoc networks join together.

Handling this scenario is the goal of this conflict detection method. A problem statement for network merges and partitions, was well presented in the draft "Ad hoc network autoconfiguration: terminology and problem statement" [8], and is quoted here:

> "- Dealing with network merges and partitions
>
> By the nature of MANET, two or more ad hoc networks which are initially isolated, can merge together or a single ad hoc network can get partitioned into two or more separate networks, at any moment in time. While network partitioning may not cause any severe problem in the MANET's operation, it may be needed that network partitioning is detected so that the resources (e.g. limited number of addressed) can be re-used among the nodes. Network merger introduces challenges to maintain the address uniqueness. Normally, once an address is allocated to a node, it continues using it and at the same time defending its own addresses from being allocated to any other node.
>
> However, since initially isolated ad hoc networks allocates addresses independent with each other, there remains some probability of more than one node using same address once two/more independent ad hoc networks merge."

Hence address conflicts may occur as a result of the case of the merge of two or more ad-hoc networks (previously out of range with each other), as there may have been no preexisting coordination for their autoconfiguration.

In this document, the "promotion of address uniqueness" is performed by a protocol which is running in parallel with the routing protocol, and address autoconfiguration method, and checks, constantly, whether or not an address conflict exists (checking address uniqueness). In the case where an address conflict exists and is detected, it is assumed that the address autoconfiguration method will allocate another address, and such method is not the subject of this document (enforcing address uniqueness in case of conflict).

3. Terminology

Terminology:

Address Conflict: When two nodes in the same MANET network share the same address or candidate address, the situation is described as an "Address Conflict". The nodes involved are "conflicting nodes" and their shared address is called "conflicting address".

Address Conflict Detection (ACD): Address conflict detection is the action of detecting address conflict, the situation where some nodes are going to use or u sing the same address in the same MANET network.

Duplicate Address Dectection (DAD): In this document, we avoid using the term "Duplicate address detection" (DAD), as it already has a predefined meaning in the IETF, and substitute the previous term of "Address conflict detection".

Conflict Detection Tree (CDT): The tree which is build in order to check the address conflicts is termed a "Conflict Detection Tree" (CDR).

Conflict Detection Tree Root (CDTR): In the protocol, one node is at the root of the CDT: this node is naturally called the "Conflict Detection Tree Root" (CDTR). This node has the responsibility to perform address check.

4. Protocol Principles

This section describes the principles of the protocol, and their rationale.

For considering address conflict detection, one could start from the following observation: the detection of the existing of a given address conflict in a network requires that at least one node, somewhere, is being aware of that the address is used twice by different nodes.

This observation can be turned in a solution: we point at one unique designated node in the network which will get assigned the task of checking that no address conflicts exist in the network.

This is a choice, and this solution is centralized. Hence note that several alternative solutions exist; for instance, [13] chooses a full distributed approach where all nodes get to know all the addresses and identifiers of other nodes in the network.

In any case, in order to detect conflicts, the designated node should possess two things: the addresses of all the nodes in the network, and then a means to check address duplication.

Then all information about addresses should flow to the designated node. A simple way to organize this flow of addresses to the designated node, is to use a tree (rooted on this node). Additionally, when a tree is built, a means exists for detecting address conflict: if for instance each node in the network attaches itself only once in the tree, an address conflict will be detected by the designated node when an address is present twice in the tree.

5. Protocol Overview

The rationale for the principles of protocol were highlighted in the previous Section 4. In this section, an overview of the functioning of the protocol is precised.

The protocol operates by exchange of protocol control messages ("Conflict Detection Tree message", CDT messages). The protocol messages are assumed to transmitted in a similar way to MANET routing or autoconfiguration protocol messages (messages in UDP packets). Hence they are broadcast to their neighbors (within the radio range), and this allows neighbor detection.

As described previously, the basis of the protocol is to build a tree of the nodes in the MANET: this tree may be based on the actually topology (depending of the range of control messages) of the network, as it is built by discovering neighbor nodes through the reception and diffusion of control messages.

The first step in the construction of the tree, is done by having each node advertising the choice of its root node, and receiving from neighbor nodes their choice: the root node is thus elected. Then, at the same time as one node transmits its choice of root node, it also transmits its choice of parent node: one neighbor node is chosen as parent, based, for instance, on the distance of one neighbor to the selected root (shortest can be preferable). Lastly, at the same time that the node transmits the previous information, it also transmits the list of its children: the nodes that had selected itself as parent. Moreover, it not only transmits the children, but also their children (its grand-children), as they have advertized their own CDT control messages: in fact, and to be complete, the node ends up advertising all the subtrees of its children.

With this mode of operation, the node that was elected as the root (the CDRT), will obtain the tree of all the nodes in the networks. It will then be able to detect conflict, by using passive "Duplicate Address Detection" techniques, such as those proposed in [10], in [11] and in [12]: a node should be present only once in the tree. The delicate point is that the topology might evolve: this is handled by maintaining and transmitting a Parent Selection Sequence Number (see Section 5.3). Additionally, address conflicts for the root(s) of a tree should be handled specifically.

Overall, the protocol integrates four distinct functions, detailed in the following section:

- Root election: it is a mechanism which allows nodes to choose one node as root (Section 5.1).

- Parent selection, and parent & subtree advertisement (Section 5.2)
- Address conflict detection and notification by the root (Section 5.3)
- Root Address conflict detection and notification by one node (Section 5.4)

Note that the overhead of transmitting an entire tree, may be large; however if the tree does not change, optimizations may be applied (see Section 7).

5.1. Root Election

The Root election is a common root election algorithm: each node advertizes a set of information which indicates whether or not it should have priority for being chosen as root compared to another node (for instance the IP address may be chosen, and some comparison defined on them). This information is the "Election Priority" in the message format Section 6.1, and will be precisely defined in future versions of this document.

Periodically, each node generates CDT messages. Each node detects from the control messages of its neighbors which root (CDTR) they have chosen, and the election priority of that root.

If a node detects that it has higher election priority than all nodes advertised by its neighbors, it automatically, elects itself as CDTR. Similarily, upon receiving a CDT message from a neighbor, if a node had chosen no root, or if its previous root had lower priority, the node will update its choice of CDTR.

Note that the election priority can be arbitrarily chosen, for instance it can be chosen to be the sequence of bytes of the IP address of the node. In the case, where the autoconfiguration method for address assignement is actually not fully decentralized, the priority may be given to nodes which have a part in the address assignement of other nodes (for instance, a MANET DHCP server).

5.2. Parent Selection, Parent and Subtree Advertizement

At the same time that a node chooses a root, it starts a procedure to select, amongs its neighbors, one node which will be its parent to reach the root of the tree.

The CDT protocol messages propagate the information of the distance of each node, and the path to the route. Hence, the selecting node, attempts choose one neighbor with some small distance to the root of

the tree, and all cases, avoids loops.

Once a parent is selected, the node will advertise, in its CDT protocol messages, its choice of a parent. This will allow the parent of the node to detect that new child node. Each node keeps a sequence number for its parent selection, which is incremented at each new choice. This Parent Selection Sequence Number is also advertized.

At the same time, the node will advertise all the neighbors that had chosen itself as parent in the tree. More, it will actually advertises the neighbors with their children nodes, their grand-children and so on, in fact, all the subtree of the nodes. The subtrees include the proper sequence numbers.

Because links might be unidirectionnal, a selected parent node have an asymmetrical link to the node, i.e. packets might go from the selected parent node to the node, but not in the opposite direction. To handle this situation, the node that has selected a parent will check whether or not its selected parent advertizes itself as child. If it is not the case, after some delay, the node selects a new parent.

5.3. Address Conflict Detection and Notification by the Root

Because the root has the list of all the nodes in the network, it is able to check whether an address conflict exists: if an address is present more than twice in the tree, an address conflict is detected. This task is performed each time it receives a CDT protocol message.

In a MANET environment, the topology may be changing quickly. Hence, it is expected that nodes may change their parents frequently: then they could temporily appear to be in two or more places of the tree. The address conflict detection may then lead to false alarms. False alarms at the root are not a problem in this protocol, except for the control packet overhead generated, because the receiver nodes are performing the final check. Still, to dramatically limit false alarms, the Parent Selection Sequence Number (PSSN) (and its history) is used to check if the address redundancy may be due node mobility. Note that in order to limit forced parent selection changes, triggered by topology changes, in addition, in the protocol, the CDT need not to be a shortest path tree.

Upon detection of an address conflict, the root will inform the proper conflicting nodes (through a message forwarded by the intermediate nodes in the tree) that such a conflict has been detected. This message includes the other advertized parent of the conflicting node, and the associated Parent Selection Sequence

Number.

From this address conflict advertizement message, and from the history of its recent parent selection, the target node is able to detect with complete certainty if the conflict is actually a false alarm, or most of the time, is an actual conflict. In this case, the node will take proper action.

5.4. Root Address Conflict Detection and Notification by one Node

In case of a conflict of addresses of the CDTR, two (or more) trees may be created in parallel. Another known mechanism is introduced: using an identifier, consisting of mostly random bits, to check if a conflict exists, with a sufficiently low probability. Such an identifier is included in the header of CDT protocol messages, and hence each node in the tree is performing this verification, upon reception of each message.

Upon detection of a conflict, a node will send a conflict detection notification message to the root of the tree. This message is forwarded to the root of the tree.

6. Detailed Specifications

6.1. Message format

The CDT protocol exchanges messages in UDP packet. The format chosen for such messages is illustrated on figure Figure 1, more precisely, CDT protocol advertizement messages. The message type should be "CDT_ADVERT".

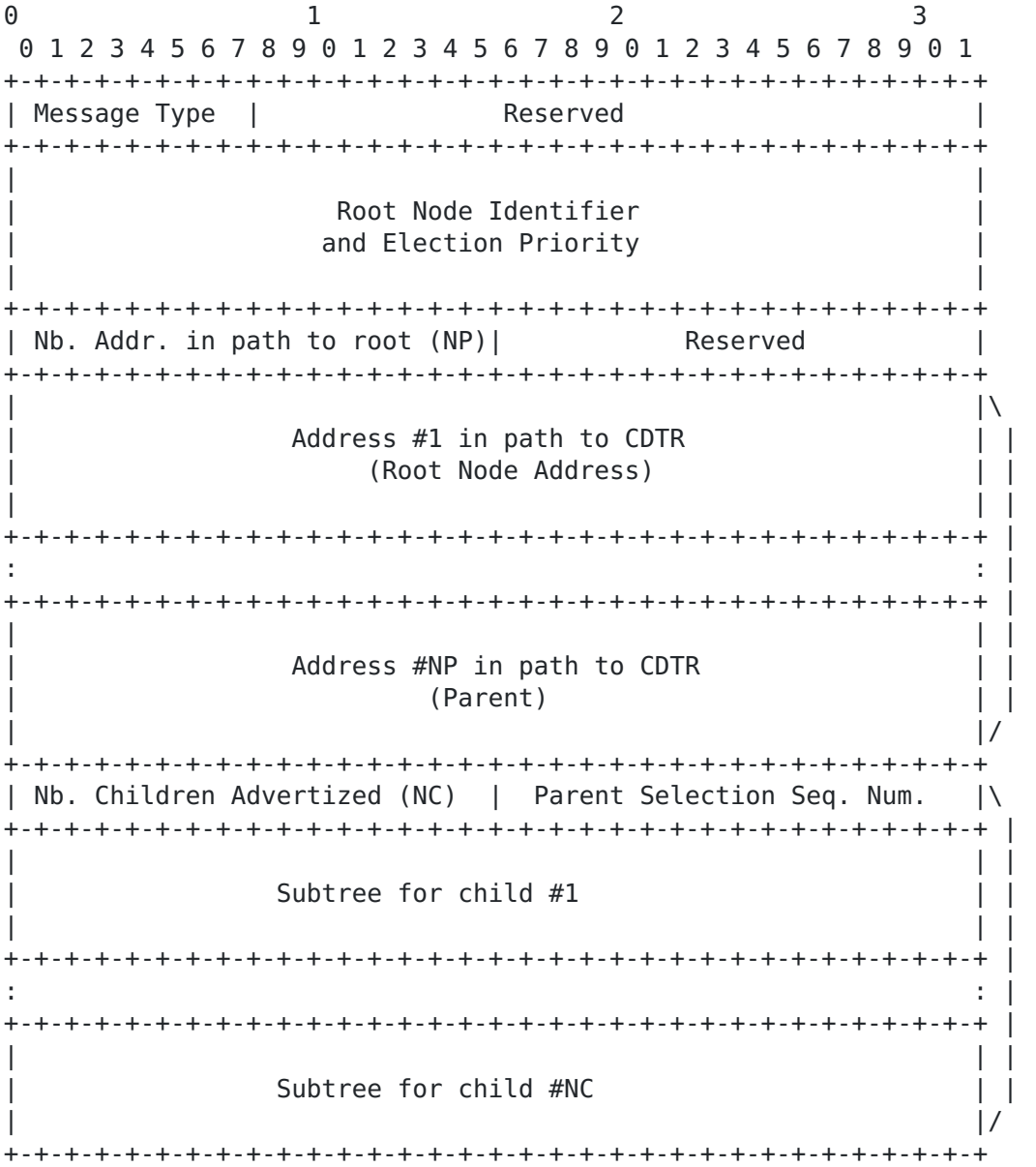


Figure 1

Such messages include the subtrees of child nodes which have a structure which is described on figure Figure 2. The subtree of a child node is given as a list of subtree for grand-child, which have exactly the same format, recursively. Because packet size is limited, a node may transmit the different subtrees in different messages, hence, part of the subtrees may be transmitted.

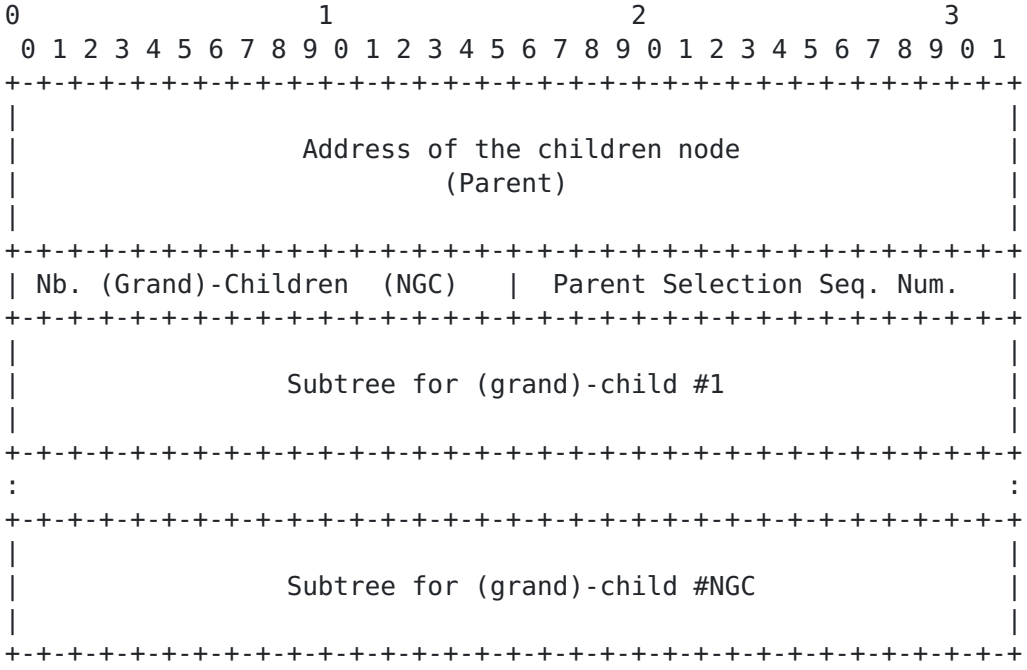
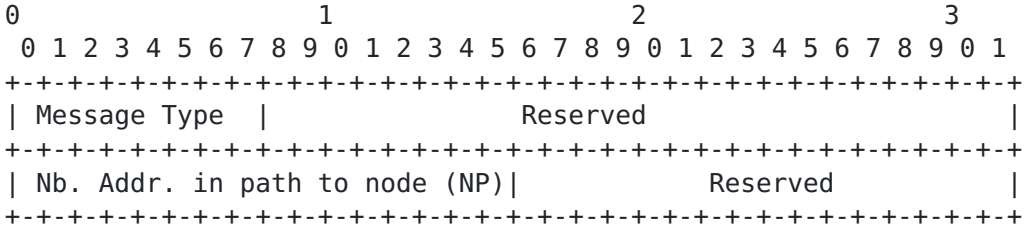


Figure 2

Additionally, the protocol transmits address conflict notification messages, with message type "CDT_CONFLICT". Their format is represented on Figure 3. The same format is used both when the CDTR advertizes a conflict to a node, and when a node advertise a root-address conflict. In both case the path is contained in the message. In the first case, the "Root Node Identifier" and the "Conflicting Root Node Identifier" are set to identical values and all the address in conflict are given. In the second case, only the "Node Identifier" fields are set, and the field "Nb. Nodes in Conflict" is set to 0 (and no conflicting parent addresses are advertized).



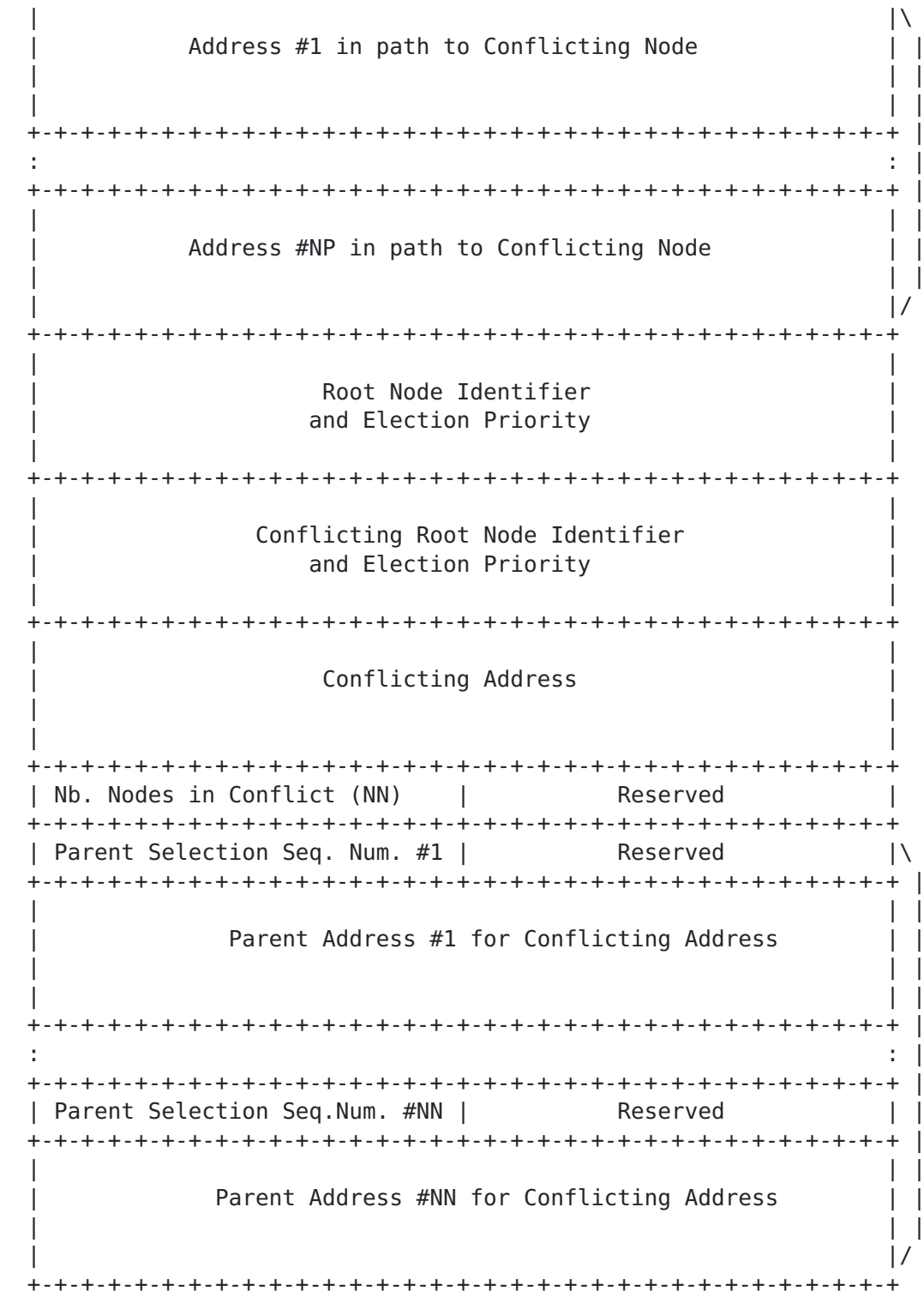

Address #1 in path to Conflicting Node
Address #NP in path to Conflicting Node
Root Node Identifier
and Election Priority
Conflicting Root Node Identifier
and Election Priority
Conflicting Address
Nb. Nodes in Conflict (NN)
Reserved
Parent Selection Seq. Num. #1
Reserved
Parent Address #1 for Conflicting Address
Parent Selection Seq.Num. #NN
Reserved
Parent Address #NN for Conflicting Address

Figure 3

7. Optimizations

Many optimizations may be applied to the CDT protocol, including the following:

- because the tree is not supposed to change often, it is easy not to send the whole subtree each time indeed, but only differential updates.

- it is possible to make a more resilent protocol by having different trees at the same time, or by having not really a tree: a node may chose different parents. This requires some changes to the simple mechanism above but can be made.

- in case of mobility, some repair-strategy might be used which do not require re-building a whole tree, but only re-using known subtrees.

In this version of the document, it was chosen to concentrate on main mechanisms and postpone the introduction and application of such optimizations. Moreover, many protocols define similar mechanisms or protocols, and their design may be reused: for instance the STAR routing protocol maintains exactly a tree for routing at the source; TBRPF also maintains trees. In one proposal by Jelger and al. for Manet Gateway Autoconf, a protocol is also used to maintain a tree to the gateways. Other extensions to MANET routing protocols exist with also uses tree in some way, for instance Multicast extensions for OLSR. In general, many other tree maintance protocols exists in the routing protocol literature and in the MANET autoconfiguration litterature. Hence, in the future, it is intended to carefully review such protocols before introducing an optimization mechanism.

In practice also, many mechanisms detailed in this document may be provided by an underlying routing protocol implementation or an underlying MANET autoconfiguration methods, in which case, the implementation becomes simpler and the protocol overhead is decreased.

8. IANA Considerations

This document has no actions for IANA

9. Security Considerations

This document presents only framework for conflict detection in MANET autoconfiguration for stand-alone MANETs and does not specify any special security measure.

10. Acknowledgements

This work was partly funded by the Strategic Information and Communications R&D Promotion Programme (SCOPE), Ministry of Internal Affairs and Communications, Japan.

11. References

11.1. Normative References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

11.2. Informative References

[2] Corson, M. and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations", RFC 2501, January 1999.

[3] Perkins, C., Belding-Royer, E., and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", RFC 3561, July 2003.

[4] Clausen, T. and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)", RFC 3626, October 2003.

[5] Ogier, R., Templin, F., and M. Lewis, "Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)", RFC 3684, February 2004.

[6] Johnson, D., "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)", draft-ietf-manet-dsr-10 (work in progress), July 2004.

[7] Ruffino, S., Stupar, P., and T. Clausen, "Autoconfiguration in a MANET: connectivity scenarios and technical issues", draft-ruffino-manet-autoconf-scenarios-00 (work in progress), October 2004.

[8] Singh, S., "Ad hoc network autoconfiguration: terminology and problem statement", draft-singh-autoconf-adp-01 (work in progress), October 2005.

[9] Bernardos, C. and M. Calderon, "Survey of IP address autoconfiguration mechanisms for MANETs", draft-bernardos-manet-autoconf-survey-00 (work in progress), July 2005.

[10] Weniger, K., "Passive Duplicate Address Detection in Mobile Ad hoc Networks", Proceedings IEEE Wireless Communications and Networking Conference (WCNC) 2003, New Orleans, USA, March 2003.

[11] Mase, K. and C. Adjih, "No Overhead Autoconfiguration OLSR", draft-mase-manet-autoconf-noaolsr-00 (work in progress),

May 2005.

[12] Clausen, T. and E. Baccelli, "Simple MANET Address Autoconfiguration", draft-clausen-manet-address-autoconf-00 (work in progress), February 2005.

[13] Laouiti, A., "Address autoconfiguration in Optimized Link State Routing Protocol", draft-laouiti-manet-olsr-address-autoconf-01 (work in progress), July 2005.

Authors' Addresses

Cedric Adjih
INRIA Rocquencourt, France
Project HIPERCOM
Domaine de Voluceau -Rocquencourt
BP 105
Le Chesnay 78153 cedex
France

Phone: +33 1 3963 5215
Email: cedric.adjih@inria.fr

Pr. Kenichi Mase
Information and Communication Network Lab.,Niigata University
Niigata University
Niigata 950-2181,
Japan

Phone: +81 25 262 7446
Email: mase@ie.niigata-u.ac.jp
URI: http://www.net.ie.niigata-u.ac.jp/

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.