RSVconf: Node Autoconfiguration for MANETs

Robert Bredy, Tatsuaki Osafune, Massimiliano Lenardi
Hitachi Europe, Sophia Antipolis Laboratory
1503 Route des Dolines, F-06560 Valbonne, France
Email: bobbredy@hotmail.com, tatsuaki.osafune@hitachi-eu.com, massimiliano.lenardi@hitachi-eu.com

***Abstract*— A Mobile Ad-hoc NETwork (MANET) is a self-configuring network of mobile nodes connected by wireless links, which can form an arbitrary topology.**
We have worked on a new protocol, RSVconf, to ensure the IP address self-configuration of MANETs, with a special focus on the ITS (Intelligent Transportation System) context which is indeed the most mobile scenario.
The RSVconf protocol manages the creation, merger and re-merger (merger after part of the network temporarily separated) of networks at IP layer. It is stateful, distributed and routing independent. The simulation results show its capability to react fast and correctly to the rapid topology changes of mobile networks without a waste of bandwidth.
In this paper we present the current status of our research, open aspects and future directions.

I. Background

Autoconfiguration is the procedure used by MANET nodes to get automatically a unique IP address. The protocol must be able to create a new network, assign addresses to newcomer nodes, and treat network mergers and partitions. Our target is the ITS context where the number of nodes is limited (less than 1000 nodes), the mobility speed is high, the mobility is defined by the other factors like driver behavior and traffic rules.

The wired network address assignment mechanisms (manual configuration or DHCP) are not suitable for wireless highly mobile networks. New specific protocols must be conceived and optimized to achieve a fast convergence to support the mobility.

One of the most interesting protocols presented is MANETconf [1] which proposes a reliable Duplicate Address Detection (DAD) requiring an answer from all nodes. A new arriving node (the requester) looks for a configured neighbor (the initiator) in order to obtain its configuration information. The initiator broadcasts an address for the requester on the MANET. All nodes have to answer to the request to avoid using the address of a node that has been temporarily disconnected from the MANET. If a node does not answer after, it is considered as having left the MANET and its address becomes available. This implies that each node of the MANET keeps a list of all assigned addresses in his MANET. Mergers and partitions are managed with Network Identifiers (NID).

Another interesting protocol is DACP [2] where every addresses have a lifetime. NID are used. The AA maintains the state information of the network. There are a Primary AA (PAA) and a backup AA (BAA). Newcomer selects a candidate IP address and then broadcasts an Address Request (AREQ). If a node has the same address or the PAA has the address in its list, they reply with an Address Reply (AREP). AREQ are sent for a number of times if no AREP is received. When the node has finished the DAD and receives advertisement from PAA it will send a Registration Request to the PAA. PAA updates its database (DB) and sends a Registration Reply. Nodes have to perform a periodical registration to renew address lifetime. There are also other protocols like Pacman [5], IPv6saa [6], NOA-OLSR [7], Prophet [8], Boleng [9], Zeroconf [10], Buddy [11] which target the autoconfiguration in other contexts.

The solution adopted here as starting point is EPDI (EPidemic Dissemination of Information) [3]. EPDI uses the concept of the epidemic dissemination of information: single nodes form an abstract global entity which has global network knowledge. Each node has a partial view of the network; to have a global sight all local databases must be collected and consolidated. The protocol easily supports the network creation and the newcomers joining. Mergers and remergers are supported only for few nodes (about 25 nodes).

These protocols are not suitable for the targeted scenarios. MANETconf requires too much messages, traffic to maintain the statefulness, avoiding the scalability and sometimes detects wrong partitions. DACP, like other centralized approaches, is fragile because too much importance is given to few nodes. Furthermore mergers are slower because the leader has to manage all the procedure. EPDI manages mergers and remerges only for few nodes and needs to pick information from the routing protocol.

This paper is organized as follows: initially we present RSVconf, its features and the details of the algorithm; then the simulations done and the corresponding results are described. Finally we conclude with a discussion about the possible improvements.

II. RSVconf

RSVconf is an autoconfiguration protocol for MANETs realized to support the high mobility of scenarios like those in the ITS context (Intelligent Transportation System). The nature of this environment requires affording fast network mergers and remergers and easy adaptation. RSVconf can easily create a network and reach the stability after a merger.

A. Features

The main characteristics of RSVconf are:

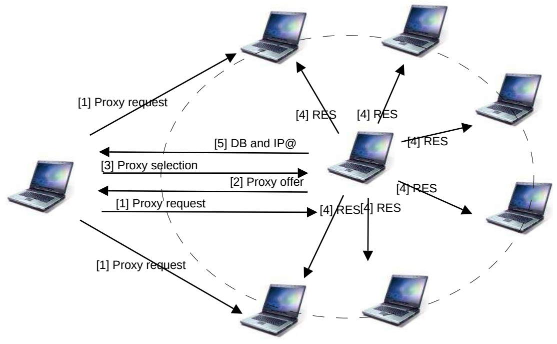

Fig. 1. Picture of the proxy, reservation and configuration phase

- Statefulness: each node has a global view of the network. Any node could manage a merger. Statefulness allow to react fast to topology changes.
- Merger: due to the statefulness, merger of two partitions is quickly solved. Because of NIDs and periodic messages, the joining of different partitions is easily detected such as the address duplication.
- Remerger: RSVconf allows network previously partitioned to merge together without changing all addresses. In case of a remerger of many networks, they are considered two by two.
- Fast reaction: due to flexible parameter settings, events can be quickly detected, for example mergers could be detected instantaneously.
- Number of nodes: performances are good with MANETs formed by 200 nodes or less, there are not packet fragmentations. Above 200 nodes the packet size is higher than the MTU (Maximum Transmission Unit) limit, thus packets must be fragmented reducing the performances.
- Scenario: ITS context.
- Independence of routing protocol: the algorithm essentially uses broadcast messages which need only one-hop routing protocols. Only in the first phase of the algorithm messages are sent in unicast to a proxy directly reachable, but a rooting protocol is not necessary because we do not need to send messages trough many nodes: the destination is in the one-hop range.

B. RSVconf Algorithm

The protocol is divided in four phases which correspond to each moment of a node life:

- Proxy selection: an existing configured node is selected as configurator.
- Reservation: the IP address assigned to a joining node is broadcasted to be reserved in the network.
- Configuration: an IP address is assigned to a node.
- Merger: union of two networks.

1) Proxy selection: Initially the joining node chooses a random temporary address in a specific range. It broadcasts a PREQ (Proxy REQuest) to contact neighbors in order to have assigned a proxy. If there is not any POFF (Proxy OFFer) from a proxy, it will assign an IP address itself and it will initialize a network (generating a NID). If many proxies reply, the newcomer chooses only one of them. The selection is done sending a PACK (Proxy ACKnowledgment) to the proxy whose message is firstly arrived. Proxies which do not receive a PACK delete the pending request.

2) Reservation: The proxy looks for a free IP address in its IPDB (IP DataBase). If one is available it will broadcast a RSV (ReSerVation message). The RSV reaches all nodes in the network, so they can add the address in their local databases maintaining the information homogeneity. At each RSV received, nodes check if the address is not already present in the IPDB, and if not, they register the IP in the IPDB and the RSV ID, which is an identifier in each RSV packets, in the packet database. If a conflict of the IP address is detected a REP (REsPonse) message will be broadcasted.

If two proxies choose the same address, any node in the network can detect the conflict when it receives the two RSVs with the same address that has to be assigned but with two different proxies' addresses. One of the two proxies has to be informed that there is a conflict and that it has to change its selected address. The ”reservation” message can be sent in the network anyway because the address will be assigned despite all, but different proxies should not send two RSVs with the same requested address.

3) Configuration: In the address assignment message newcomers find their selected IP address, the random value associated (this random value is used to differentiate two nodes with the same address, considering that the probability that two nodes generate the same random number is very low) and the IP database of the proxy. If the value of the address is NULL, it means that there is not any available address and it has to restart the configuration procedure after a timeout. By sending the database the statefulness is assured also for the newcomer.

The newly configured node has to set up its periodical activities to keep the obtained IP address. It sets the IP renewal time and starts the merger detection procedure. Each address has a lifetime and before the lifetime expired, the node has to send a RSV to keep its address. Additionally, other periodical messages are sent to detect the merger of networks as is stated in the next section.

4) Merger: To characterize each network, a NID (Network ID) should be used. It is a random number chosen by the network initiator. To detect merger of networks, nodes periodically send a broadcast message: DM (Detect Merger), limited to one hop, containing the NID and a hash computed on the list of IP addresses and their associated random value present in the local DB. The retransmission period is set between 1s and 2s. When a node receives a DM with a different NID or a different hash it starts the merger procedure.

Only the nodes at the edge (only two nodes) have to communicate and exchange their databases in order to find duplicated addresses. To avoid multiple merger processed at the same time, a waittime is introduces before starting the merger. If there are more than 3 networks to be merged, only the merger of two networks are processed. If Node A is the one that sends the DM and node B is the one that detects the merger, Node B sends its database through the MERHI (MERger HI) message, which contains its IP database, to the Node A, which computes the new network database and the new NID if it is not a remerger. Node A sends then a reservation message which contains the new database to the whole network. Each node refreshes its database. The node whose address appears in the new database has to check the random value. If the random value corresponds to its own random value, it simply neglect it, because it means that there is only one node with this address. Otherwise it changes its address.

Moreover each node, in order to avoid multiple successive mergers, when receiving the RSV after a merger, freezes itself for a short period, disabling renewal time and merger detection beaconings to avoid additional traffic. By doing so, the current procedure should terminate correctly.

III. Simulations

All simulations have been done using the NS-2 network simulator. The scenario tries to reproduce IVC (Inter Vehicle Communications), where driver behaviors, constraints on mobility, and high speeds give unique characteristics [4]. In order to evaluate the protocol performance, some criteria like the number of messages or the number of bytes received and the time needed to react to events were considered.

The utilization of the bandwidth is difficult to count in the case of broadcast messages like almost all the messages of RSVconf. Moreover nodes are sparse in more than 2km, the radio range is 250m, and many clusters exist. The bandwidth should be computed separately for each zone, but they are not independent from each other. The number of messages and their composition is the best evaluation criteria to analyze the medium usage. Moreover the number of bytes sent by the RSVconf in the whole network was counted. Finally the average number of bytes concerning the protocol received by each node is calculated.

Furthermore the capability to react fast to an event is important because the ITS context is characterized by rapid topology changes, frequent fragmentation of the networks. If the node is alone, without neighbors it has to create the network, thus it takes more time because it has to wait a timeout. Instead if the network was already formed, the node takes shorter time to be configured. Even in the case of a merger the reaction will be fast, it means that it is detected immediately, if the timeout to send the DM (Detection Merger) message is short (1s).

There are three initial networks: one (A) formed by 10 cars simulates a crash, the second (B) formed by 40 cars which arrive periodically from the left and finally (C) the remaining cars [number of nodes - 50] which arrive from the right. Nodes in network A do not move, and the first car is positioned at 500m on the first lane with the intervals of 5m. Each Node in network B starts to move each 2 seconds. They are placed at 0m and they go to 4000m on the second lane at a speed of 90 km/h. The network C is created at 1500 m on the first lane, each node joins every 3s. They move immediately until they meet the last car of the network A.

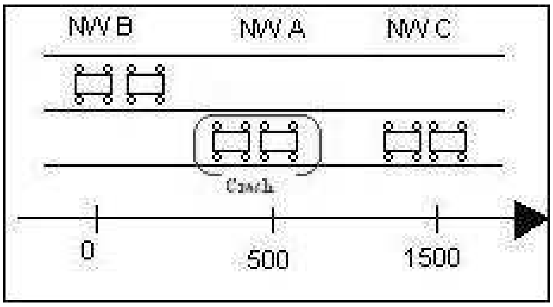

Fig. 2. ITS context scenario

The simulations were done from 140 to 200 nodes and the simulation time was 600s. Other simulations were done from 60 nodes to 200 nodes to analyse the scalability. Fig.2 shows the ITS scenario used in simulations

At the beginning of simulations networks are well separated, so it may happen that at the same time we have nodes that need first-time autoconfiguration and nodes already configured that are merging through their networks.

This scenario, in which the bandwidth usage is not so important because the number of merging nodes is not high, simulates a real context when nodes arrive during an ongoing merger.

This scenario allows also testing the remerger functionality which successfully works.

A. Network creation

Fig.3 shows the time needed for IP autoconfiguration for 200 vehicular nodes. The average time to configure a node is 3.4 seconds for 50 simulations. This value is influenced by the network initiator nodes which have to wait longer timeouts (10s) like the first nodes (nodes around the ID 0, 10 and 50). The median is 1.8s; the median is more representative than the normal mean because is less influenced by outliers. Furthermore, 75% of the nodes takes less than 1.8s to be configured.

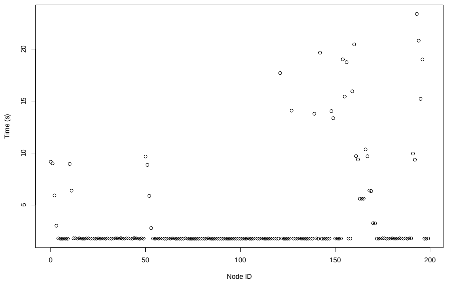

Fig. 3. Configuration time

Nodes which take more than 2s to be configured can not reach their neighbors; the packets concerning the proxy phase are lost. In this case these nodes create a true partition and later they merge. Normally they can contact other configured nodes and they wait additional 10s before to restart the configuration procedure.

B. Packets sent

The main traffic load is generated when there is a merger and the databases are sent. In this scenario mergers normally happen when networks have few nodes and databases are not big. At the end of the simulation, the nodes enter in a stabile phase with only the DM messages being sent.

C. Packets received

The maximum number of bytes received in 1s is 57511 Bytes, 460 kb/s in the case of merger. On average the number of bytes received each second by a node is 1363 Bytes, 11 kb/s.

D. Scalability

Fig. 4 shows the increment of number of messages received per second versus the network load. The number of bytes sent during a simulation increases when adding other nodes (in our cases 20 nodes each time). The traffic load is proportional to the number of nodes, it increases without a limit. In effect every node sends periodically a DM message and participates to the broadcast flooding incrementing the traffic.

Each node receives more packets when the network size is larger, but we noticed that the number of DM received does not augment when the number of nodes is higher than 140 because the number of one hop neighbors does not change, since the radio range is always the same.

The increment is principally due to the size of the packets which increases when the number of nodes becomes higher because they contain information about all nodes. From the graph we can see that the variance of the number of messages received in average each second by a node is very small, confirming that the packet size is the main component to take in account for scalability. Fig. 5 shows us the trend of the configuration time when the number of nodes increments. Although the average value is affected by the number of outlier nodes, as is already stated, the graph shows the trend that the average configuration time becomes longer according to the number of nodes in the network.

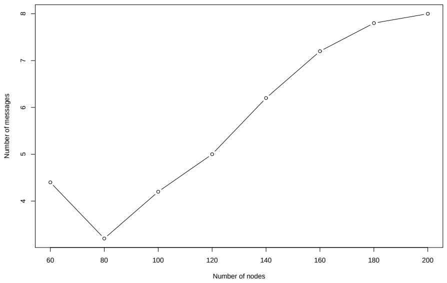

Fig. 4. Messages received per second vs. number of nodes

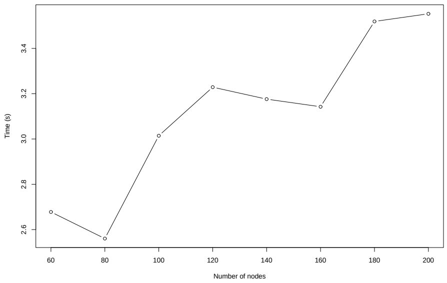

Fig. 5. Configuration time vs. number of nodes

IV. Conclusions

A new protocol for MANET node IP autoconfiguration, RSVconf was designed and proposed.

The protocol was conceived to work in an Intelligent Transport Systems (ITS) context, thus it ideally requires networks with a modest number of nodes and it supports high mobility and frequent network partitions and mergers in contrast with DACP, MANETconf and EPDI. The protocol is distributed, stateful, independent from the routing protocol.

Simulations prove the correctness by the absence of address duplications while network creations, newcomers' configurations and mergers are done successfully. Mergers are always done in less than 1 second and the nodes with a duplicated address are reconfigured within the delay initially set. RSVconf slightly reduces its performances when the number of nodes increases; this because the packet sizes are proportional to the number of nodes (so when there are more than 200 nodes the packets should be fragmented). The bandwidth used is low; the maximum peak noticed at the reception was 460 kb/s in only one simulation. Finally the total independence from any routing protocol is confirmed.

The future work should improve the choice of which of the involved nodes has to change the address among the duplicated addresses in order to reduce the communications failures and deal with newcomer configuration in a middle of a merger. Furthermore, RSVconf should be simulated in scenarios when using the packet fragmentation in order to analyze further scalability and related performance.

References

[1] S. Nesargi, R. Prakash, *MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network* - IEEE INFOCOM, 2002 - ieeexplore.ieee.org

[2] Sun, Belding-Royer, *Dynamic Address Configuration in Mobile Ad hoc Networks (DACP or HCQA)*

[3] Osafune, Yamamoto, *Analysis of an Epidemic Dissemination Protocol for Ad Hoc Networks*

[4] JJ. Blum, A. Eskandarian, LJ. Hoffman, *Challenges of Intervehicle ad Hoc Networks* - IEEE Transactions on Intelligent Transportation Systems, 2004 - ieeexplore.ieee.org

[5] K. Weniger, *PACMAN: Passive Autoconfiguration for Mobile Ad hoc Networks*- IEEE Journal on Selected Areas in Communication, 2005 - ieeexplore.org

[6] K. Weniger, M. Zitterbart, *Ipv6 Autoconfiguration in Large Scale Mobile Ad-Hoc Networks* - Proceedings of European Wireless, 2002 - ing.unipi.it

[7] Mase, Adjih, *No Overhead Autoconfiguration for OLSR (draft-mase-MANET-autoconf-noaolsr-00)*

[8] H. Zhou, L.M. Ni, M.W. Mutka, *Prophet Address Allocation for Large Scale MANETs* - IEEE INFOCOM, 2003 - ieeexplore.ieee.org

[9] J. Boleng, *Efficient Network Layer Addressing for Mobile Ad Hoc Networks (Boleng)* - ICWN - toilers.mines.edu

[10] M. Gunes, J. Reibel, *An IP Address Configuration Algorithm for Zeroconf. Mobile Multi-hop Ad-Hoc Networks (CAC)* - On Broadband Wireless Ad-Hoc Networks and Services, 2002 - projectmesa.org

[11] M. Mohsin, R. Prakash, *IP Address assignment in a mobile ad hoc network (Buddy)* - MILCOM, 2002 - ieeexplore.ieee.org