Some Requirements for Autonomic Routing in Self-Organizing Networks

Franck Legendre, Marcelo Dias de Amorim, and Serge Fdida

LIP6/CNRS – Université Pierre et Marie Curie
8, rue du Capitaine Scott – 75015 – Paris – France
`{legendre,amorim,sf}@rp.lip6.fr`

Abstract. This paper addresses some requirements of self-organizing networks as well as interoperability problems due to merges and splits phenomena. In a mobile environment, merges and splits characterize the spatial overlap between two self-organized networks. While merge refers to the time when two disjoint networks meet and overlap, split refers to the time of partition. In a dynamic environment, AutoComm (AC) principles bring a new support for interoperability since current protocol heterogeneity is observed at all stack layers from the radio interface to applications. In this paper, we reconsider the formalization of a community and its requirements. We then characterize the split and merge phenomena and their implications. We give some requirements that must fulfill solutions to merging (high context-awareness) in order for AC groups to self-scale. Finally, we propose a merging solution for overlapping wireless self-organized networks using heterogeneous routing protocols.

1 Introduction

Current networks are limited by principles edicted 30 years ago when requirements of mobility did not exist. Since that time, several innovations were proposed to bypass inherent limitations of IP principles (*e.g.*, NAT, Mobile IP, IPv6). Moreover due to the dynamic nature of ad hoc networks, principles driven by end-to-end requirements do no longer apply. New innovative routing paradigms must be designed.

In fact, routing is the basic *service* of a network and any other service resides on this fundamental functionality. Hence, we believe routing requires being adaptable to the diverse environments, new usages, and QoS requirements desired by applications. Two opposite directions respond to this new constraint. The first approach claims that different routing schemes suit different contexts and that *one routing protocol fits all* cannot be envisioned. Currently, this approach has led to an heterogeneous set of routing protocols. The second approach proposes new flexible communication paradigms such as i3 [1] and Network Pointers [2], which are adapted to new constraints brought by mobility. With i3 the act of sending is decoupled from the act of receiving. Addresses are based on a communication identifier either unicast or multicast. Network pointers extend the semantic of addresses from basic identifiers to specific packet handling functions that are not just limited to forwarding. Both approaches bring innovations but we believe interoperability is required until a unique protocol or paradigm prevails if one can be designed. Indeed, diversity is the right approach to fit the multiplicity of contexts. It is all the more true since the design of routing protocols also relies heavily on the underlying layers and radio access technologies. These technologies are only beginning to develop with Ultra Wide Band (UWB) [3], Multiple Input Multiple output (MIMO) [4], and beam forming with smart antennas. The *diversity* of solutions (at the routing layer) hence enables to better suit evolution of usages, requirements of the environment and underlying radio technologies. We consider that diversity will be an invariant of forthcoming routing protocols and that AutoComm (AC) will enable to handle efficiently this situation. Diversity managed by AC principles will enable new protocols and paradigms to fusion or be dropped. This process will be similar to living elements' natural selection that uses sexual reproduction to enhance the overall fitness of their genetic patrimony [5].

For example, first developments in wireless routing were the adaptation of wired protocols (*e.g.*, RIP, OSPF) giving birth to OLSR [6] and reactive protocols (*e.g.*, DSR [7] and AODV [8]) which are direct adaptations of the Address Resolution Protocol (ARP) to multi-hop wireless networks. However, combining these two approaches lead to more efficient and adaptive protocols such as ZRP [9] and SHARP [10]. AC's framework brings a new chance to reinvest routing. We must restart the cycle of defining routing protocols dedicated to specific purposes and conditions. This might lead possibly to combined solutions and enhance the overall benefit.

Nethertheless, this evolution shows limitations of current proposals for ad hoc routing. These relate to two correlated factors. First, the model of the wireless channel was considered similar to the wired Internet and researchers only considered new requirements brought by mobility. Up to now, evolutions in networking where direct application of wired technologies to the wireless world without rethinking the basics. People saw wireless ad hoc networks as wired networks with end-to-end requirements and hence narrowed their vision required to bypass such a limited model. Second, such limitations stem from the fact that the network architecture as it is designed nowadays reflects our incapacity of communicating between people involved in lower layers with people involved in upper layers. IP, by its universal goal of unifier marks the barrier between both sides and is the point of convergence. This leverages the question of what is now the advantage of such a rigid model in a changing environment. Cross-layering brings a first response to bypass limitations of the layering paradigm. AC is to come next.

AC proposes to reinvest a research effort to bypass these limitations by proposing a dynamic framework and an interdisciplinary view of all networking aspects including routing. AC is a new opportunity to avoid rehearsing the same mistakes. We believe, however, that the design of AC-compliant network elements will not respect in the short term AC's design rules and philosophy. We believe that a first step will combine existing solutions while introducing AC components. The final step will be the design of communicating elements fully AC-compliant. New networking functionalities must be designed accordingly to

AC principles but still interoperate with current existing protocols. For example, with the current diversity of ad hoc and wired routing protocols, interoperability has not been much tackled. We think that the goal of AC is limited, among other purposes to enable protocol interoperation. In our case, we focus on routing protocol interoperation. This temporary solution will at term leave place to a single yet completely dynamic framework for routing that will support the emergence of innovative paradigms.

In this paper, we address the issue of one of the main challenging problems in next generation networks, namely routing in *merge and split* environments. We will mainly study the merge phenomenon and its implications, and define requirements of how an AC must react to merges. Routing has mainly been designed to cope with the scaling relative to the number of individuals collaborating in a group and thus cope with expanding networks (*e.g.*, Internet growth, wireless network radio coverage). These individual entities subscribing to a network engender small scale events. On the contrary, large scale events such as splits and merges are more frequent to occur in a dynamic environment. Due to mobility along with expansion, networks are likely to spatially overlap[1] and separate. Networking and especially routing require efficient and appropriate solutions. If we count upon heterogeneity, the dynamic nature of networks leverages particular difficult problems. We will detail a scheme based on AC principles that enables to merge wireless networks using distinct routing protocols to efficiently interoperate. We define it as an evidence that AC principles are pertinent in dynamic environments subject to merge phenomena. We give practical solutions to ensure our requirements are enforced.

The paper is organized as follows. Section 2 introduces a more general definition of merge and split related to the dynamics of communities. We formalize merges by defining their nature and implications. We then define merge and split as a general framework mainly related to interoperability issues. We give a set of requirements for protocols to be merge-compliant. Section 3 focuses on the implications of merging at the routing layer. We give a proof-of-concept of AC principles applied to the merge problem with two networks using distinct routing schemes, (*i.e.*, AODV, DSR, and OLSR) and give a solution overview. We also tackle improvements to the proposed solution in order to scale with the increasing diversity of routing protocols. Finally, Section 4 discusses future research investigation and concludes this paper.

2 A general view of Merge

In the following, we mainly focus on merging of wireless networks and leave splitting for future investigation.

---

[1] Here we narrow merging to a physical overlapping but a more general definition is given in the following.

2.1 What is a Community or Autonomic system

A general definition of a group is a number of network elements that share a common set of stable patterns of interactions. These interactions are essentially driven by social relations (*e.g.*, meeting), involvement in a collaborative activity (*e.g.*, P2P file sharing, work meeting, students' lecture, battling troops, emergency rescue teams) or with similar spatial patterns or simply geographic proximity (*e.g.*, public transport users). These interactions – often correlated – can be represented by dynamic graphs of interactions in space and time; and mobility is only the visible part of these interactions. Note that this does not preclude a network element or AC system to belong to several groups.

2.2 Why communities merge and split

Predicting patterns of interactions is a hard task given their dynamics. Some interactions are predictable while others are not. For example, social interactions are predictable such as regular meetings or workshops. Some mobility patterns are also predictable such as for users of public transports. The dynamics of a group can be classified in two types of events: small scale events such as node arrival, node departure, or node failure and large scale events such as splits and merges.

A general definition of merging is when two or more AC systems interact in order to collaborate whether spatially close or not given that a communication means is possible. The level of collaboration between these groups depends on their purposes' correlation. Two emergency teams following a similar goal (high correlation) will require to merge when meeting while two groups with different purposes, *e.g.*, different WiFi operators spatially overlapping, might collaborate in order to interfere the least given the radio spectrum available or on the contrary offer roaming to their respective users. When merging the level of collaboration is reflected by the distinct groups' policy toward merging. The question is often to merge (high level of collaboration) or not to merge (low level of collaboration)? Depending on the negotiated level of collaboration, merge occurs at different layers (from the physical (PHY) to the application (APP) layer) and different time scales. For example, temporary splits may arise from a broken radio link due to the radio channel degradation or persist when mobility engenders sparse networks where network elements' radio coverage does not intersect.

2.3 Implications of merging

Merges and splits depending on the level of collaboration is a source of conflict at all levels. When AC group merges, there are two great classes of conflicts:

- Heterogeneity of protocols from PHY layer to APP overlays,
- Resource driven conflicts, *i.e.*, resource conflicts occurring at a given layer for homogeneous protocols. For example, the use of the same radio technology often leads to channel interferences. At the IP layer, merging requires to synchronize the addressing space of both networks in order to avoid conflicts.

2.4 Merging Requirements

Fundamental requirements of splits and merges are for AutoComm systems to keep what we call *consistency.* Consistency is the capacity for networks to keep their QoS and service level whatever small or large scale events occur at any time scale. Schemes dealing with merging and splitting must maintain or enhance the level of *consistency* by means of collaboration.

The second requirement is *smoothness*. Merges and splits are large scale events that can have a great impact on network performances. Smoothness is the capacity to cope with the smaller impact on the performances without disturbing the general QoS. In other words, we must design efficient, flexible solutions and what we can characterize as *smooth* split and merge solutions. Depending on the mobility pattern, merges and splits can be transient phenomena or on the contrary lead to stable situations. For example, current proposals tackling the problem of addressing in a merge environment assume an attraction/gravity mobility model where n wireless networks gather at a defined geographic location. This model leads to a permanent state where two networks spatially overlap. In this case, several schemes propose mechanisms to synchronize the address space in a coherent way so that no address conflict occurs. This can be done through flooding or other means. Nevertheless, with random mobility patterns merging leads to transient states where networks only cross by. In this case, re-addressing an entire network can be a sub-optimal solution.

The last requirement is *efficiency.* It requires from schemes to detect merge and split phenomena as quickly as possible and react appropriately.

With AC, going further than just detecting a phenomenon such as merging but by characterizing more deeply its nature (permanent, transient) will enable more scalable solutions to perform. These requirements require context-aware schemes.

2.5 Merge-awareness

Merge-awareness is a kind of context-awareness or selfware. Context-awareness and what we define as merge-awareness enables to gather enough explicit information or if not available, to infer the underlying phenomenon occurring and take appropriate actions. For example, as explained before, mobility reflects one or several interactions a network element or AC is involved in. Characterizing mobility allows inferring the underlying interaction. In [11], the authors infer the will to merge of two wireless networks as shown in Fig. 1 by computing the relative velocity of both networks as a function of time. If this velocity is likely to converge toward zero, networks decide to merge since it reflects a tendency to effectively collaborate and leads to a stable state that will permit to optimize reconfiguration if needed. Other inference schemes carried out at a higher level study social interactions as an input [12], [13].

To enhance context-awareness, again if we borrow concepts from biological cells, AC systems require a memory similar to the immunological memory. The immune system and its memory allow efficient response to subsequent encounters with similar infections. The AC matching piece to infections is basically the environment context. Since situations (hence similar context) are likely to re-occur frequently given Zipf law, an environment context memory is required. As well, given the same reasons, merges and splits are to re-occur frequently. Keeping tracks of past merges and splits and recognizing an AC group that has partitioned in the past may enable an efficient re-merging thanks to a past shared context.

Solutions to cope with splits have been more tackled since splits are more predictable in their nature. Solutions are twofold, either they use a reactive approach *i.e.*, detect the occurrence of splits and react appropriately by replicating what we call the *patrimony* in both splitting networks so that each networks is a duplicate (similar to bio-cell mitosis) or they use a proactive approach by periodically spatially replicating required patrimony in case of future split occurrences [14] [15]. The patrimony refers to the sum of all available services and information that are required by an AC group to still be autonomous in case of splits. The purpose of these schemes is to keep the consistency of both separating networks. There lies a trade-off between both approaches.

2.6 Interoperability

Merge is a very challenging issue in a dynamic environment. All layers are impacted by merging; from PHY to APP layers. Hence, merging has to cope efficiently with *heterogeneity* at all layers. As stated before, the AC paradigm will create autonomic systems using heterogeneous protocols. Similarly to biological systems that are defined by their *fitness* [5] as the ability to fit their local environment, routing must follow the same concepts. What we require from AC is to fit all situations. Routing must consider the group's purpose and the nature of the underlying phenomenon of the group's dynamic at small and large scale. This will require routing interoperability following the edicted requirements. We study such a case in the next section.

3 Proof-of-concept

In this section, we give a proof-of-concept of AC principles applied to merging networks. We give an example that assists self-organization of network elements between AC groups using different routing protocols (routing protocol heterogeneity). We recall that in our vision, we consider AC as a means to federate existing solutions by enabling interoperability. This is a first step before fully compliant AC protocol design. This interoperability is subject to policy rules and requires specific function to sense the environment and bring context-awareness.

As stated before, merging requires being as smooth as possible. Since it is a large scale event, repercussion must be minimized for both merging AC groups.

Most ad hoc routing protocols are well suited for particular situations and hence are rigid. As said before, this evolution comes from the fact that routing protocols are a direct adaptation of wired routing protocols to the wireless world.

Hence, the rigid nature of current ad hoc routing protocols leaves little space for adaptation. One solution we detail in the following is an AC daemon that enables a dynamic interoperability between rigid networks. Nevertheless, it is important to say that adaptive routing protocols such as ZRP are a first attempt to adapt the routing parameters to the sensed underlying environment.

3.1 Model and hypothesis

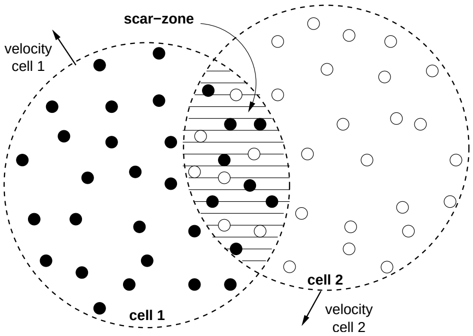


Fig. 1. Network cell model.

In this paper, we define the concept of network cells. A cell, C, is the spatial region spanned by a set of nodes, N_C, willing to participate in some collaborate activity or having similar spatial behaviours. The evolution of a cell is progressive, *i.e.* new arriving nodes acquire the address allocation scheme and routing scheme from an existing cell member. The address assignment may be of any kind (conflict-free, conflict-detection, or best-effort). For example, nodes can acquire an address from their neighbours or randomly generate their addresses and verify their uniqueness using a duplicate address detection (DAD) [16] mechanism. For routing, we assume that due to the diversity of situations, nodes implement a set of existing protocols. However, since equipments have heterogeneous capabilities (*i.e.*, processing, memory), they do not always implement the same set of routing protocols but it is likely that they will implement the most adopted ones. We believe that a restricted list of protocols will be supported by vendor equipments, some respecting standards and others proprietary implementations. When cells C_1 and C_2 are merging, the overlapping region $\mathcal{S}_{C_1,C_2} = C_1 \cap C_2$ is called a *scar-zone* and is delimited by a *scar-zone membrane* (cf. Fig. 1). Nodes located in the scar-zone are called *scar-zone* nodes, S_C, while nodes outside the scar-zone, I_C, are named *interior* nodes. Depending on the respective mobility of the two cells, the scar-zone can evolve between a minimal overlapping where the two cells are interconnected via one radio link to a complete overlapping where the spatial extent of one cell is included in the other.

3.2 Routing merge requirements

We consider the case where cells C_1 and C_2 use heterogeneous routing protocols restricted to OLSR, AODV, and DSR. As described in section 2, we require merging to be as smooth as possible since it is a large scale event. The intuitive solution to the case we are dealing with would be to reconfigure entirely one of the two cells in order to have the same routing protocol. This requires all reconfigured nodes to implement the new routing protocol. Moreover, in a mobile environment successive reconfiguration due to successive merging in a short lap of time will lead to oscillations. Even if both cells use the same routing protocol (*e.g.*, OLSR), synchronizing link-state (LS) tables might be sub-optimal if merging is only transient. Besides, swapping to a different protocol will incur a *patrimony loss* (on-going communications, routing states) and more importantly break all Service Specific Routing (SSR) overlays relying on the previous routing protocol. Recall that the choice of the *physical* routing protocol relies highly on both the underlying environment and the requirements of the applications unless both routing protocols can *cohabit* but this raises scalability issues. We believe our scheme will benefit to situations of transient merging. These occur either when networks just cross by (no stable overlapping, constant relative velocity during merging) or during the transient phase occurring with the attraction mobility pattern (*i.e.*, short period during which the relative velocity of merging cells is non zero). We plead for a transient solution that will take effect until a permanent situation is detected and that will be able to choose the most suited protocol given the context or find the appropriate parameters for adaptive routing protocols.

3.3 Design

We briefly explain our approach here. The purpose is not to explain in detail the mechanisms required for loop-free routing with heterogeneous ad hoc routing protocols but to address the problems arising and give insights toward more efficient and innovative solutions.

In order to achieve interoperability between merging cells, scar-zone nodes must define their neighbourhood environment context *i.e.*, the routing capabilities of their neighbours and the current protocol in use –the *mother* routing protocol. This is done with the Neighbourhood Routing Protocol Discovery Protocol (NRPDP). For example, in Fig. 2 node X sends $\{$`*AODV*,DSR`$\}$ to Y, pointing out the routing protocols it supports and the current routing protocol in use in its cell, indicated by `*Protocol*`. We supposed nodes or both cells to have a common set of routing protocols but are not currently using the same one when merge occurs. Depending on the neighbourhood context, specific interaction must be performed. Scar-zone nodes must either translate routing packets or, if translation is not possible, nodes must execute appropriate neighbourhood interaction. We define a new routing daemon, the Routing Translator Daemon (RTD), that intercepts the I/O of routing control packets (requests, replies, and updates) and given the context information provided by the NRPDP processes these packets accordingly.

3.4 Application: AODV ↔ DSR

We give here an application of our framework for the interoperability case between AODV and DSR. Since AODV and DSR belong to the same family *i.e.*, reactive routing protocols, a translation is only required. On the contrary, if we required interoperability between a reactive protocol and a proactive protocol (*e.g.*, OLSR) an other kind of interaction should have been used.

Consider the scenario shown in Fig. 2, where cell 1, C_1, runs AODV and cell 2, C_2, runs DSR. Consider also two nodes, A and B, with $A \in I_{C_1}$, $B \in I_{C_2}$ (*i.e.*, $\{A, B\} \not\subset \textsf{scar-zone}$). Here, we study how paths can be established in these cells in both ways, from A to B and vice-versa.

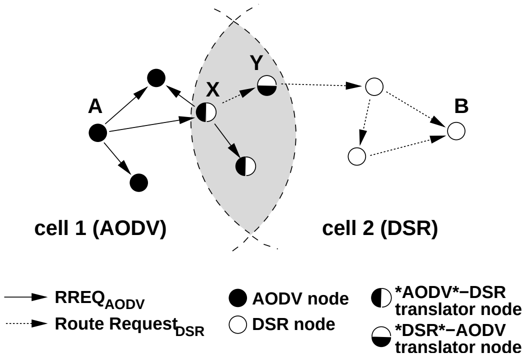


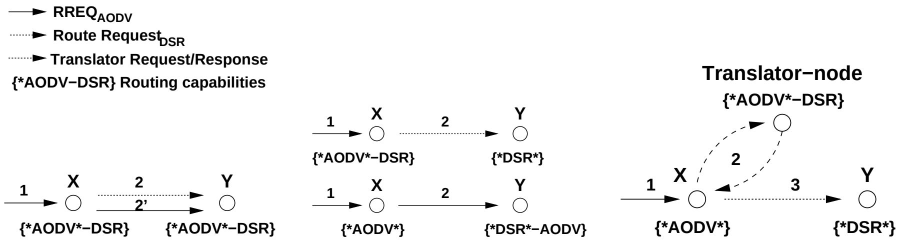
Fig. 2. Translation AODV ↔ DSR.

For the establishment of a path from A to B, $A \to B$, A floods a `RREQ` (Route Request). When a node in the scar-zone receives a request, here X, it translates the AODV `RREQ` into a DSR `Route Request`. This translation is associated with a new entry in a dedicated table maintained by the RTD. This entry will indicate that a translation will be required at the reception of the DSR `Route Reply`. When the DSR `Route Reply` is received, an AODV `RREP` is sent using the reverse route entry established by the initial `RREQ`. In order to avoid loops, the translation requires the use of the same request identifier, the same sequence numbers (d_{seq} and o_{seq}), and the precise translation of the number of hops. Since the sequence numbers of both protocols have different field sizes and to enable recursive translation, we use a `hash` function to associate AODV's `RREQ` sequence numbers with DSR's `Route Request` and add a new header dedicated to our RTD daemon. For the hop count, in the AODV header a hop count field represents the number of hops, while in DSR counting the number of concatenated IDs gives the hop number. Consulting the corresponding entry in the RTD table does this translation. As well, the correct association between routing control packets prevents recursive translations. For example, X receives an AODV `RREQ` from Y in reply to its original DSR `Route Request` sent to Y. By comparing the packet *id* and sequence numbers with the entry in its table, X detects this route request is generated in response to its original request.

Similarly, for the path $B \rightarrow A$, B floods a DSR `Route Request` that is translated into an AODV `RREQ` by scar-zone nodes. These scar-zone nodes will receive `RREP`s that will be translated back to DSR `Route Replies`.

3.5 Extensions to our model

Here, we relax some constraints on our hypothesis in order to obtain a more realistic model. In the previous model, we supposed all nodes to have similar capabilities at the routing layer referred as the routing capabilities or RC set. We supposed nodes to have a set of homogeneous routing protocols but now we consider that some nodes have smaller capabilities or different capabilities than others.



(a) (b) (c)

Fig. 3. Specific-context translations.

Figure 3 lists all the possibilities for the previous case with a wider hypothesis on nodes' capabilities in terms of supported routing protocols. The different cases that occur will influence the reaction of our RTD given the neighbourhood information communicated by the NRPDP as the following:

- The first case, shown in Fig. 3(a) reflects the hypothesis of previous subsections where $RC_X \sim RC_Y$. Here, node X can either forward the AODV `RREQ` as it is or translate it into a DSR `Route Request` as long as the node that makes the translation updates its RTD table, *i.e.*, $RC_X \sim RC_Y \rightarrow X_{translation} \vee Y_{translation}$.
- The second case, shown in Fig. 3(b) is when $RC_X \subset RC_Y \vee RC_Y \subset RC_X$. Here, the translation is done by the most capable node, *i.e.*, $RC_Y \subset RC_X \rightarrow X_{translation}$ otherwise $Y_{translation}$.
- The last case, shown in Fig. 3(c) is the most complex to deal with. Here $RC_X \cap RC_Y = \{\}$. In this case, network entities require to find a node with more capabilities able to work out the translations. We call these nodes, *translator nodes*. We need these nodes to organize in an SSR overlay. Whenever a translation is required, one of these nodes works out the translation on behalf of the *incapable* node. How the overlay of most capable nodes is

structured depends directly on the underlying routing protocol. In reactive protocols, nodes must use expanding ring search to find a translator node while in proactive approach where the topology is known, special entries can be added in the LS update packets or maintained by a new daemon dedicated to maintain the translator overlay. As shown in Fig.3(c), node X requests a translator-node which has a greater routing capability and thus is able of performing the translation on X's behalf, *i.e.*, $RC_X \cap RC_Y = \{\} \rightarrow Find_{translator}$ where $Find_{translator} \wedge Reactive\ routing\ protocol \rightarrow Expanding\ ring\ search$ or $Find_{translator} \wedge proactive\ routing\ protocol \rightarrow Request_{SSR-translator-overlay}$.

Note that we have limited the cardinal of the capabilities set to 2, $|RC| = 2$. But, other more complex possibilities could enhance our approach. The last case requires extending our scheme with a SSR overlay dedicated to routing translation.

4 Conclusion

In this paper, we have addressed one of the future challenges networks will be faced to. We characterized splits and merges as large scale events that occur at different time scales with their causes and implications. We reviewed existing solutions and proposed yet simple but promising solutions for ad hoc routing interoperability. We have shown how AC can fully respond to challenges of merge and split in heterogeneous environment. We can draw several conclusions. First, if reactive/programmable approaches were used there would be no need of such a scheme. As we expressed before, reactive approaches will surely reappear for radio access technologies (PHY-MAC layer) with SDR [17] which enables a radio interface to be reconfigured by software. This will enable to swap from Bluetooth to IEEE 802.11 or ETSI HiperLAN, and to 3G and 4G. Nevertheless, we must take into account that reconfiguration can lead to sub-optimal solutions and initiate oscillations if splits and merges are to occur frequently. Second, our scheme will not scale with an increasing diversity of ad hoc routing protocols. With our designed scheme, $n(n-1)/2$ general translations rules are required and as much specific translation rules (cf. sub cases of extensions) with n as the cardinal of existing routing protocols. We believe our proposition will help in entering a new phase for routing. Our proposition has two opposite goals. On the one hand, respond to the urgent need of interoperability and in the other hand, our proposition is aimed at showing that interoperability is not always feasible and that innovations are still required in order to not reproduce errors of the past (*e.g.*, ITU's Interworking units, OSI's internetworking (IDRP)). Since the tendency is toward protocol heterogeneity, we believe AC principles must enable protocol interoperability easily with more adaptable protocols. We are currently carrying out simulations using NS-2 in order to validate the suitability of our scheme under various mobility models.

References

1. Stoica, I., Adkins, D., Zhuang, S., Shenker, S., Surana, S.: Internet Indirection Infrastructure (i3). In: Proc. of the ACM SIGCOMM Conference (SIGCOMM'02), Pittsburg, PA, USA (2002) 73–88
2. Tschudin, C., Gold, R.: Network pointers. In: First Workshop on Hot Topics in Networks (HotNets-I), Princeton, NJ, USA (2002)
3. EURASIP: Special issue on UWB - State of the Art. In: Journal on Applied Signal Processing. EURASIP (2004)
4. Shafi, M., Gesbert, D., shan Shiu, S.D., Tranter, W.: Guest editorial: MIMO systems and applications. In: JSAC. Volume 21-3. IEEE (2003)
5. MacKay, D.: Information Theory, Inference, and Learning Algorithms. Cambridge University Press (2003)
6. Clausen, T., Jacquet, P.: Optimized Link State Routing Protocol OLSR. RFC (2003) IETF RFC-3626.
7. Johnson, D., Maltz, D.: Dynamic Source Routing in Ad Hoc Wireless Networks. In: Mobile Computing. Volume 353. Kluwer Academic Publishers (1996)
8. Perkins, C., Royer, E.: Ad-hoc On-Demand Distance Vector Routing. In: Proc. of MILCOM (MILCOM'97). (1997)
9. Haas, Z.: A New Routing Protocol for the Reconfigurable Wireless Networks. In: Proc. of the IEEE International Conference on Universal Personal Communications, San Diego, CA, USA (1997)
10. Ramasubramanian, V., Haas, Z., Sirer, E.: SHARP: A Hybrid Adaptive Routing Protocol in Mobile Ad Hoc Networks. In: Proc. of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'03), Annapolis, MD, USA (2003)
11. Legendre, F., Amorim, M., Fdida, S.: Implicit Merging of Overlapping Spontaneous Networks. In: Proc. of the IEEE Vehicular Technology Conference 2004-Fall (VTC'04), Los Angeles, CA, USA (2004)
12. Wang, B., Bodily, J., Gupta, S.: Supporting Persistent Social Groups in Ubiquitous Computing Environments Using Context-Aware Ephemeral Group Service. In: Proc. of 2nd IEEE International Conference on Pervasive Computing and Communications (PerCom'04), Orlando, FL, USA (2004)
13. Kurvinen, E., Oulasvirta, A.: Towards Socially Aware Pervasive Computing: A Turntaking Approach. In: Proc. of 2nd IEEE International Conference on Pervasive Computing and Communications (PerCom'04), Orlando, FL, USA (2004)
14. Wang, K., Li, B.: Efficient and Guaranteed Service Coverage in Partitionable Mobile Ad Hoc Networks. In: Proc. of the 21st IEEE Infocom (Infocom'02), New-York, USA (2002)
15. Hanna, K., Levine, B., Manmatha, R.: Mobile Distributed Information Retrieval for Highly-Partitioned Networks. In: Proc. of the 11th IEEE International Conference on Network Protocols (ICNP'03), Atlanta, Georgia, USA (2003)
16. Nesargi, S., Prakash, R.: MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network. In: Proc. of the 21st IEEE Infocom (Infocom'02), New York, USA (2002)
17. JSAC-IEEE: Special issue on Software Radios. In: Mobile Computing. Volume 17-4. IEEE (1999)