Simeonov, Plamen L. :

WARAAN: a higher-order adaptive routing algorithm for wireless multimedia in wandering networks

---

Zuerst erschienen in:
Proceedings / 5th International Symposium on Wireless Personal Multimedia Communications, 27-30 Oct. 2002, Honolulu, Hawai. - Piscataway, NJ : IEEE Service Center, 2002, Vol. 3, S. 1385-1389
DOI: 10.1109/WPMC.2002.1088407

WARAAN: A Higher-Order Adaptive Routing Algorithm for Wireless Multimedia in Wandering Networks

Plamen L. Simeonov
Technology University of Ilmenau
Dept. of Telematics
Ilmenau, Germany
plamens@web.de

Abstract

The Wandering Network (WN) [1] is a new type of communications architecture defined by:

1. flexible, multi-modal specialization of network nodes as virtual subnetworks;
2. mobility and virtualization of the net functions in hardware und software;
3. self-organization as multi-feedback-based topology-on-demand.

Network elemens can contain several exchangeable modules capable of executing diverse network functions in parallel. These functions can be invoked, transported to or generated in the nodes upon delivery of mobile code containing programs about the node's behaviour.

An essential characteristic of the WN approach is the inheritant ability to instantly spread out information about architectural changes among the nodes by encoding executable re-constructon (genetic) instructions within the transported active packets – as "network" genes, N-genes.

Keywords

Adaptive Routing, Active Networks, Genetic Encoding, Multimedia Architectures, Ad-hoc Mobility, Autopoiesis.

1. INTRODUCTION

Active networks (AN) has been a subject of intensive empirical investigation for the last decade. Their major design goal has been to accelerate the rapid introduction and deployment of new network protocols and services. A number of different models such as active nodes, active options and capsules have been proposed to implement AN architectures. Most of them investigate to some detail a specific network solution. Implementations have shown that every single network issue such as caching, routing, management, etc. can have a specific *active* network solution. A few survey papers were published trying to provide directions and goals for engineering within the field ([2], [3], [4]).

Recently, an integration and consolidation of the several different AN engineering approaches can be observed. This trend is particularly evident at technology frontiers such as deeply embedded networked systems, autonomous software, configurable computing, adaptive systems, etc. A number of requirements have been collected to activate the network, but there is still no general recipe to address all the problems with only one end-to-end active network. The "killer" network of the future has not been found yet.

Along with the growing scope and number of ad-hoc solutions to active networking, the demand for their systematic categorization, evaluation and integration within a common research framework becomes increasingly evident. In particular, an *evolutionary* approach to active networking requires the development of common models for: a) the encoding of network programs in terms of mobility, safety and efficiency; b) the description and allocation of node resources; c) the built-in primitives and *behavioral* patterns available at each node.

We regard networking as a *synthetic* science. Therefore, the goal of this work is to provide a generic design methodology, referred to as the *Wandering Logic Intelligence (WLI),* for reasoning about autonomous networked systems. Ultimately, the methodology is aiming to deliver a *formal* recursive design model of the discourse domain, further referred to as the *Wandering Network* [1], which has been defined as a superset of the worlds of evolving active networking, reconfigurable computing [5] and adaptive systems [6], viewed from the perspective of biological *autopoietic* systems [7]. The above three research fields were brought together for the following reasons. Firstly, active networking defines the principle and the goal of our research. Secondly, reconfigurable computing brings up the required *detail* and understanding within a context. Thirdly, adaptive systems encompasses the large field of heuristic techniques in AI for the purpose of organizing and optimizing *wandering* media communications. Finally, multimedia communications provide a challenging perspective on applying the WLI approach to the design and verification of autonomous adaptive architectures ([8], [9]).

This paper presents the WARAAN adaptive routing protocol for mobile multimedia in active ad-hoc networks based on the WLI virtual network topology concept. It illustrates the inherent capabilities of the Wandering Network model which can be applied to any kind of network. In an ad-hoc mobile network, the algorithm can be layered along with a other routing algorithms and policies depending on the application context.

0-7803-7442-8/02/$17.00 © 2002 IEEE

2. INVESTIGATION FRAMEWORK

The long term goal of Active Networking is to make the network so simple that we are able to design more complex architectures (adapting, self-configuring, self-deploying, autopoietic, etc.).

An essential characteristic of the WLI approach is the *inheritant ability to instantly spread out information about architectural changes among the mobile nodes, netbots,* of the Wandering Network by encoding *executable re-constructon (genetic) instructions* within the transported shuttles – as *"network" genes, N-genes,* cf. Figure 1.

The Propagation of Architectural Changes
in a Wandering Network

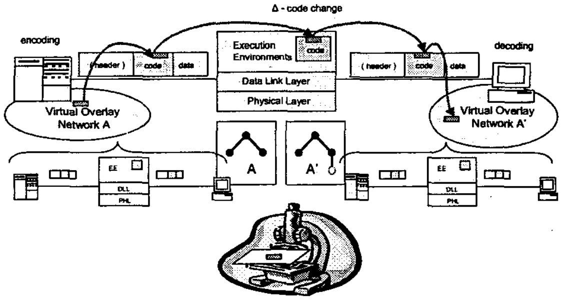


Figure 1: Encoding, transport, change and decoding of architectural information inside the Wandering Network

This is a unique feature which differentiates the Wandering Network from all previous approaches in active and programmable networks. Therefore, we selected the subject of routing in ad-hoc mobile networks for a case study to demonstrate the feasibility of the WLI approach to network evolution. *The goal towards we are striving is the proof of the assumption that AN technology as an integral part of the WLI approach delivers an appropriate methodology for automating the process of route adaptation, and hence of propagating topology changes within a dynamically changeable network infrastructure.* For this reason, network monitoring is distributed between the nodes of the network.

Routing issues in ad-hoc mobile networking are a difficult challenge for protocol designers, since rapid reconstruction of routes is crucial in the presence of topology changes. The primary concerns in ad-hoc mobile networks are bandwidth limitations and unpredictable topology changes. In such an environment, it is important to minimize disruptions caused by the changing topology for critical application such as voice and video. Furthermore, agreeing on which algorithm is the "best" may even be more challenging.

By using an active network approach we can (a) delay this decision until run-time, and (b) hopefully dissolve it by letting different routing algorithms run in parallel.

3. THE WLI ROUTING ALGORITHM

This section discusses the application of the WLI approach to adaptive routing in ad-hoc mobile networks.

WARAAN, the WLI Adaptive Routing Algorithm for Active Ad-hoc Networks is based on the notion of an *attributed non-terminal* which was defined by Vogt, Swierstra, and Kuiper as a part of their milestone work on Higher-order Attributed Grammars (HAG), [10]. In the past 30 years, attributed grammars have proved to be appropriate means for structured modelling in many language-based application areas such as pattern recognition [11], graphics systems design [12], electronics and logic circuit programming [13], as well as neural networking [14]. Because the goal of all routing algorithms is "to discover the *sink*[1] *trees* for all routers" and because of the simple linear encoding of parsing trees generated by the production rules of a context-free grammar, we preferred to use the HAG model in our WLI routing scenario with some modifications, instead of generating a more complex network model based on a general formal approach such as the graph grammars, [15]. Moreover, the universality of higher-order tree transducers was recently motivated again by Noll and Vogler [16] for a series of applications.

We assume that the routing state of the Wandering Network is completely described by the set of reachability trees (r-trees) T_R of the individual netbots participating the network. A reachability tree is the replacement for a routing table in WLI.

Definition: A *reachability tree* (r-tree), T_R, is a dynamic directed tree structure allocated in the operating communication environment of a WLI netbot and responsible for the base routing in a Wandering Network. The root of the r-tree is always the host netbot. The leaves and the intermediate nodes of the tree are the corresponding netbots from which the host netbot can be reached. Each netbot is responsible for:

1. maintaining its r-tree by collecting information from the shuttles traversing that netbot;
2. forwarding shuttles to other destinations; and
3. reporting changes in the own r-tree structure, such as establishing new connections or cancelling old ones, to its neighbours.

A modified higher-order attributed grammar (HAG) represents our model for the netbot's reachability tree.

---

[1] To emphasise the netbot-related nature of the routing path generation in WLI, we decided to use the term "reachability tree" in this work.

In our model each node, except the root, represents an attributable virtual non-terminal. This means that at every single moment the r-tree can be expanded or collapsed at such a virtual non-terminal. They are synthesized attributes of this non-terminal represent the computed potential links to other netbots.

Let us now go back to the construction of reachability trees in WLI and assume two single netbots, A and B, freely traversing the two dimensional space. There is no connection established between them. Thus, each netbot contains only one single element in its reachability tree: itself, the root. Then, at some point of time both netbots approach each other within their access range. One of them, say A, initiates a connection protocol with the other netbot. The opposite side replies positively and the connection is established. Next, each netbot constructs a new branch of its reachability tree ending at the new neighbour.

At the next moment, a new netbot C approaches A and requests a connection. Upon a positive reply the connection is established and the local r-trees at each node are extended by the new branch. However, this time both B and C are unaware of the fact that they may contact each other by letting A to route the shuttles between them. Therefore, netbot A is required to inform each one of its neighbours about the existence of other members[2] in the network. This is achieved by encoding and encapsulating the correspondingly "missing branch" information as executable *r-genes* (reachability genes) into the *r-shuttles* (routing shuttles) which A transmits to its neighbours. Instead of a destination address and a TTL-counter (time-to-live), each r-shuttle is carrying an encoded tree branch called a *q-tree* (quest tree) it is required to traverse until being discarded at the end nodes. The communication environment can manipulate both the q-genes and the r-genes of an r-shuttle in order to update their information, cf. Figure 2. In case that netbot C also has some neighbors it can route to, it is required to send this information via r-shuttles towards A, which in turn takes care to distribute it along the remaining branches of its r-tree.

Generating a new branch of the r-tree on a netbot and dispatching r-shuttles to inform the neighbours about the change can be performed simultaneously. Besides, the same procedure is performed simultaneously on both sides of the newly established connection. These are two important advantages of the distributed WLI routing algorithm. The computing overhead for encoding and decoding the r-trees is minimal because of the event related character of the reachability tree updates.

We claim that by using the r-tree and q-tree types of encoding in the shuttles, the updated routing information is sent effectively to all affected nodes of the network.

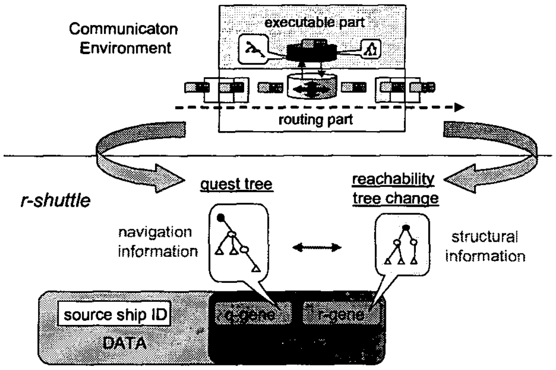


Figure 2: A communication environment manipulating the genetic structure of r-shuttles

Figure 3 illustrates the first two steps of the WLI routing algorithm, the *projection* phase:

1. Connect (X, Y) & Build (T'_X,T'_Y), and
2. Inform (X, Y, T_X ,T_Y).

The notion is taken for the general case of two netbots X and Y and their reachability trees T_X and T_Y . The prime sign upon T means the next state or the change of the reachability tree. The coloured circles denote *acting* nodes with the red one being the new netbot joining the network. The oval legends display the (parts of the) r-tree contents represented in the particular elements with the colour ones being active in the particular step of the algorithm. The shuttles on the figures are assumed to contain q-genes.

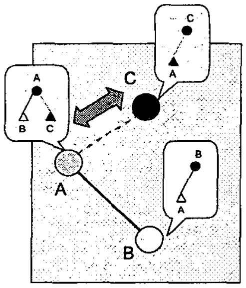
We call the second phase of the WLI algorithm the *capturing phase*. It starts with the evaluation of the incoming shuttles and the expansion of the r-trees at the referred non-terminals by the "missed branches" encoded in the r-genes.

As soon as the r-shuttles arrive at their destinations, they are guided to the corresponding communication environment responsible for the link they come from. The CE then unpacks the "missing branch" information encoded in the *r-genes*, which are part of the executable code carried by the r-shuttle, and verifies it with the structure of its reachability tree.

If the delivered information is redundant, i.e. the r-tree has been already constituted that way that the r-gene information represents a sub-branch of the netbot's r-tree (perhaps by a previously delivered shuttle from some other source), it is discarded.

---

[2] We postulated in WLI that *fairness* and *cooperation* are a must. All kinds of hiding and manipulating information for any reason are not subject of this work.



i) Connect (A,C) & Build (T'_A, T'_B)

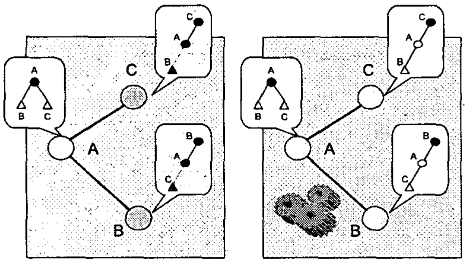
ii) Inform (B,C)

Figure 3: Projection: building and transporting r-trees

In case that the "new branch" information is a *really* new one, the CE takes care for expanding the r-tree at that virtual non-terminal which is assigned to be a root in the sub-tree encoded in the corresponding *r-gene*, cf. Figure 4, step iii.

Finally, the reachability trees of all netbots are verified against each other (cf. Figure 4, step iv) by broadcasting periodically r-shuttles containing the entire r-tree to the neighbors which analyze the incoming information on their side with the local tree structure and send back their feedback to the originating node. If no feedback is registered on a connection after some period has elapsed, the associated link is considered for failed and the change is reflected in the local r-tree and reported to the neighbors. If the requested netbot is only an intermediate station on the path of the shuttle, the responsible CE updates the netbot's reachability tree by the r-shuttle's information and forwards it to the next hop on the shuttle's path. If there are any new structural changes on the shuttle's route ahead known by the CE, the shuttle's q-tree is updated.

The same four steps – *Connect*, *Inform*, *Expand* and *Verify* –, are taking place every time a new netbot joins the fleet. This is because of the distributed and parallel nature of the WLI algorithm which complexity is estimated to be **O (4 + m)**, where "m" is the maximum number of hops throughout all r-trees in all nodes participating the network.



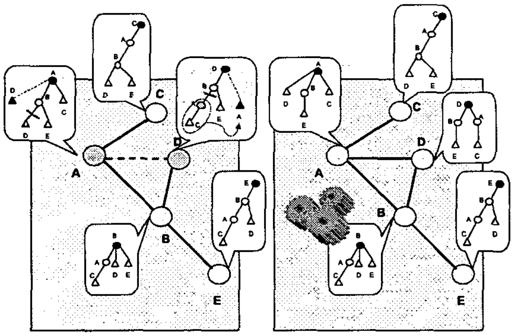
iii) Expand (T_C , T_B)

iv) Verify (T_A , T_B , T_C)

Figure 4: Capturing: expanding and verifying r-trees

In case that a new, direct link is established between two netbots which already communicate[3] through other nodes, the "shortcut" is passed through as a new branch in the r-trees of the both netbots as shown on Figure 5. Then, in the same step of the algorithm, each r-tree is depth-searched again to eliminate the dummy links and relocate the remaining branches on a shortcut path. For instance, in our case the link (B,D) is cancelled in the r-tree of netbot A, since there exists a shorter path from D to A. Analogously, the link (B, A) is cut through in the reachability tree of netbot D because D and A are now communicating directly, and not via B.

However, since A leads to C on that path and there is no other way for C to reach D, except through A, so the (A,C) branch is relocated, i.e. expanded, at the newly generated A. The complexity of this part of the WLI algorithm is **O(4).**



i) Connect (A, D) & Build (T'_A , T'_D)

ii) Verify(T'_A ,T'_B ,T'_C ,T'_D ,T'_E)

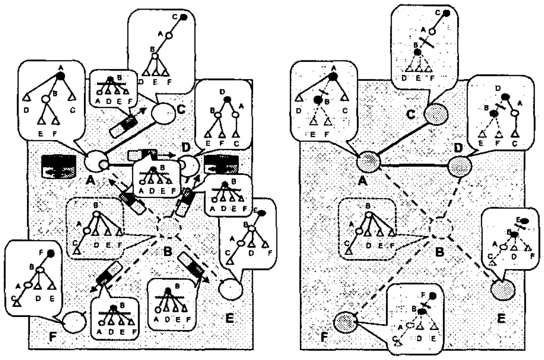
Figure 5: Introducing a Short-Cut

At the end, let us consider the propagation of the r-tree changes when a netbot leaves the fleet for some reason (failure, movement, etc.) as shown on Figure 6. Firstly, the netbot can leave the network gracefully by informing its neighbours for his intention. Secondly, even if the netbot is going to leave the network spontaneously, this case can be reduced to the graceful one.

Each netbot can maintain an *alarm shuttle* (a-shuttle) containing an *a*-gene with the first level of the netbot's reachability tree which includes the direct neighbours as leaves. The a-shuttle has a unique identifier that can be recognized by any netbot in the network. When the netbot intends to leave the network, it fires replicas of the alarm shuttle in all directions as a last action before going to inform its neighbours about this event. The a-shuttle is updated as soon as the first level netbot connectivity changes. This function is maintained in parallel with the rest of the netbot's activities and does not require a specific schedule. For instance, it can be performed each time a new link is established or an old one is cancelled.

---

[3] i.e. both netbots are already present as virtual non-terminals in each others' r-trees



i) Disconnect (B,X) & Inform (A, C, D, E, F)

ii) Collapse (X, B)

Figure 6: Projection of and capturing the exclusion of an intermediate node

As soon as the a-shuttle reaches a netbot, it's a-gene is unpacked and the local r-tree is updated with the new information. If the same shuttle comes later, e.g. from another line, its content is simply discarded and the shuttle is forwarded to the outgoing lines. Alarm shuttles may implement a TTL data field such as in the common data packets to spare the q-gene and though to limit their circulation in the network.

The complexity of this part of the algorithm is **O(3+m)**, where "m" is the maximum number of hops throughout all r-trees in all nodes participating the network. This value includes the firing of the shuttles (1), the collapse of the r-trees in each netbot (2), and the verification of the r-trees (3), which may take several steps depending on the newly emerged topology of the network, but though regarded by us as a single linear step.

4. SUMMARY AND CONCLUSIONS

This paper presents the WARAAN higher-order adaptive routing protocol for mobile multimedia in active ad-hoc networks based on the WLI virtual network topology concept. We are satisfied with the results which delivered four DIN A4 pages of bug-free TLA$^+$ code, [17], with Lamport's TLC model checker, [18], running on top of SUN's JRE-SE-1.3.1 under Microsoft's Windows 2000 within a man-month. This work is for further study.

The major assumption in applying the WLI model for this application domain is: *the more a node knows* about its neighbors and environment, *the better it can serve* the ad hoc mobile community. For instance, the delivery rate of the two-hop variant of the GEDIR[4] algorithm [19] can be improved significantly if each mobile ad hoc node is aware of its 2nd-hop neighbors, i.e. the neighbors of its neighbors. The WLI methodological framework can be extended to monitor spontaneous changes in network topology and node behaviour

5. REFERENCES

[1] P. L. Simeonov, "The *Viator* Approach: About Four Principles of Autopoietic Growth On the Way to Hyperactive Network Architectures", *Proc. of 16th IEEE International Symposium on Parallel & Distributed Processing (IPDPS'2002)*, April 15-19, 2002, Fort Lauderdale, Florida, USA, http://www.prakinf.tu-ilmenau.de/IPI/FGT/.

[2] D. Tennenhouse et al., "A Survey of Active Network Research", *IEEE Comm. Mag.*, Vol. 35, No. 1, Jan. 1997, pp. 80-86.

[3] K. L. Calvert, S. Bhattacharjee, E. Zegura, J. Sterbenz, "Directions in Active Networks," *IEEE Communications Mag.*, Oct. 1998, pp. 72-78.

[4] A. Campbell, H. De Meer, M. Kounavis, K. Miki, J. Vicente, D. Villela, "A Survey of Programmable Networks". http://www.columbia.comet.edu.

[5] K. Compton, S. Hauck, "Reconfigurable Computing: A Survey of Systems and Software", Northwestern University, Dept of ECE, *Tech. Report*, 1999.

[6] DARPA Information Technology Office, *"Adaptive Computing Systems Program"*, http://www.darpa.mil/ito/acs.

[7] Humberto R. Maturana, Francisco J. Varela, "Autopoiesis and Cognition", D. Reidel Publishing Co., 1980, ISBN 9027710163.

[8] O. Spaniol, J. Meggers, "Active Network Nodes for Adaptive Multimedia Communication", *Proc. of SMARTNET'99*, Nov. 1999.

[9] P. L. Simeonov, D. Reschke, "Supporting Adaptive Multimedia with the Wandering Network Model", *4th Int. Forum on Multimedia and Image Processing (IFMIP'2002)* at the "*World Automation Congress (WAC'2002)*", June 9-13, 2002, Orlando, Florida, USA, http://wacong.com.

[10] H. H. Vogt, S. D. Swierstra, M. F. Kuiper, "Higher Order Attribute Grammars", Proc. of the ACM SIGPLAN'89 Conference on Programming Language Design and Implementation, ACM SIGPLAN Notices, 24(7), 1989.

[11] W.-H. Tsai, K.-S. Fu, "Attributed Grammar – A Tool for Combining Syntactic and Statistical Approaches in Pattern Recognition", IEEE Trans. on Systems, Man and Cybernetics, Vol. SMC-10, No. 12, Dec. 1980.

[12] L. A. Barford, B. T. V. Zanden, "Attribute Grammars in Constraint-based Graphics Systems", Software – Practice and Experience, Vol. 19(4), pp. 309-328, April, 1989.

[13] F. Holmgren, "A Grammar-based Approach to Design and its Application to Electronics and Logic Programming", Res. Report, Intelligent Systems Laboratory, Swedish Institute of Computer Science (SICS), December, 1997.

[14] S. Hussain, R. A. Browse, "Attribute Grammars for Representations of Neural Networks and Syntactic Constraints of Genetic Programming" Proc. AIVIGI'98, Workshop on Evolutionary Computation, June 17th, 1998, Vancouver, BC, Canada.

[15] H. Göttler, "Graphgrammatiken in der Softwaretechnik: Theorie und Anwendungen", Springer-Verlag, 1988.

[16] T. Noll, H. Vogler, "The Universality of Higher-Order Attributed Tree Transducers", Theory Comput. Systems, Vol. 34, pp. 45-75, Springer-Verlag, New York, 2001.

[17] L. Lamport, "The Temporal Logic of Actions", ACM Toplas, 16, 3, pp. 872-923, May, 1994; also in: http://www.research.digital.com/SRC/personal/lamport/tla/.

[18] L. Lamport, "Specifying Systems", Draft, August 2001, http://www.research.compaq.com/SRC/personal/lamport/tla/book.html.

[19] X. Lin, I. Stojemovic, "Geographic Distance Routing in Ad-Hoc Wireless Networks", Tech. Report, Dept. of Computer Science, SITE, Univ. of Ottawa, Canada, Dec. 1998.

---

[4] GEographic DIstance Routing