Dynamic Gateway Selection for Cross-Domain Routing with the XLayer Communications Substrate

Marco Carvalho, Carlos Perez and Adrian Granados
Institute for Human and Machine Cognition
40 S. Alcaniz St. Pensacola, FL - 32502
Email: mcarvalho@ihmc.us

***Abstract*—This paper describes a dynamic gateway selection algorithm for cross-domain routing. The proposed cross-layer algorithm is designed on top of the XLayer communications substrate. In the context of this work, cross-domain routing refers to seamlessly routing across different networks running different routing algorithms without pre-defined gateways or a common underlaying protocol. We first provide a brief introduction of the XLayer communications substrate and the adaptive routing controller. After describing the algorithm and a summary of the proposed architecture, a proof-of-concept implementation is evaluated for a simple scenario developed in NS-2. Our preliminary results are encouraging and show the potential of our proposed strategy for dynamic gateway selection and cross-domain routing.**

I. Introduction

Tactical environments, such as those encountered in military operations or disaster relief missions, are generally characterized by ad hoc computational and network infrastructures that enable the monitoring and sharing of information from sensor fields and other assets to support mission management and coordination. Such combined networks often include heterogeneous networks under different administrative domains (for instance in Military Coalition Operations) and possibly different protocols and communication systems. Usually, there is a pre-planning phase to each mission where a support infrastructure based on gateway nodes across the multiple systems is planned and deployed, enabling the interaction (at different times during the mission) between assets, networks, and sensor fields. This is generally a lengthly and costly process which also fails to support the dynamism and flexibility envisioned for tactical enviroments.

The development of adaptive and cognitive radio systems facilitates the interaction between communication systems, enabling radios to adapt themselves and interface (at the physical layer) with each other. With such capabilities, there is an increasing need for automated cross-network bridging and gateway selection mechanisms to support a run-time adaptation to network conditions.

The XLayer communications substrate [1] provides a series of core capabilities designed to provide and support a modular communication infrastructure that allows applications and decision architecture systems to better adapt to and leverage the characteristics of the dynamic communications environment. In this paper, we introduce the design of a specialized module for the XLayer communications substrate that provides a dynamic gateway selection mechanism for cross-domain routing in military (and civilian) tactical network environments.

In the context of this work, cross-domain routing refers to the capability of routing data in a seamless manner across heterogeneous networks that may have different communication capabilities and routing algorithms. Although the issue of interoperating heterogeneous networks is general, we focus on tactical oriented mobile ad hoc networks. In particular, we are interested in developing an algorithm that will allow two (or more) mobile ad hoc networks to dynamically identify, configure, and enable a set of gateways for a seamless integration of the different routing protocols.

There has been numerous research efforts in automatic gateway selection for mobile ad hoc networks. In [2] the authors propose a position-aware algorithm for the dynamic merging of mobile networks. The proposed algorithm differentiates between transient and permanent merging, and it builds on the concept of localized edge detection [3] to create a geographically-aware algorithm for cross-domain routing and packet filtering (to minimize unnecessary transmission across networks). While a very interesting approach, the proposed solution requires position and mobility information to be shared between nodes and makes an underlying assumption of a well behaved and uniform propagation model to support routing within the network intersection areas. In our case, the proposed algorithm for dynamic gateway selection enables one or more nodes within network boundaries to automatically bridge routing packets from different routing protocols without the need to share position and mobility information between nodes.

After a brief review of the XLayer architecture, we will focus on the design and implementation of the Adaptive Routing Controller (ARC). We will introduce and discuss the proposed algorithm for dynamic gateway selection and illustrate its application over a simple mobile network scenario implemented in NS-2.

II. The XLayer Communications Substrate

The XLayer communications substrate was first proposed as part of the U.S. Army Research Laboratory's Collaborative Technology Alliance (CTA) [4] [5]. The original purpose of the research was to design better monitoring and control

9781-4244-3941-6/09$25.00 ©2009 IEEE

interfaces between applications and the underlying communications infrastructure in tactical and mobile ad hoc network environments.

The effort was later extended in collaboration with the U.S. Air Force Research Laboratory to provide support for tactical information management systems in airborne network environments, where new services and capabilities were developed and integrated as *Controllers* to the proposed infrastructure.

The architecture of the XLayer communications substrate was designed to enable a two-way interface (for both monitoring and control) between applications and the underlying communications infrastructure. The goal was to provide the necessary visibility to the underlaying communications environment to allow applications to better monitor and adapt to changes and constraints, while at the same time to specify application-level requirements to be enforced and maintained (to the best possible level) by the underlaying communications infrastructure.

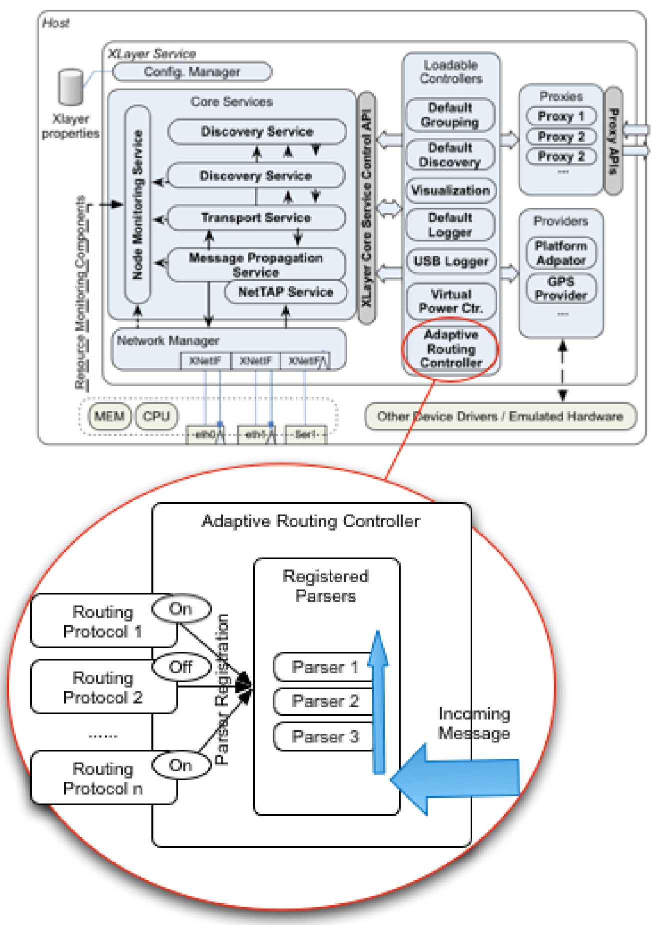

Fig. 1. The XLayer Adative Routing Controller

The XLayer communications substrate creates an abstraction of multiple interfaces to create a common view of the communications infrastructure to higher level applications and middleware. Communication interfaces may include local network interfaces as well as attached tactical radios (such as EPLRS, PSC-5x and others) and even virtual interfaces such as remote tunnels and others. This capability is provided by the *Network Manager Component* and can be fully extended to support other legacy radios or systems. The *Network Manager* plays a critical role in cross-domain routing since it aggregates multiple networks and links (including non-IP links) in a common interface.

The XLayer architecture provides the facilities for on-demand loading of fast-response control algorithms that can be parameterized by applications. One of the difficulties generally associated with cross-layer design is the coordination between layers operating at very different time-scales. Controllers may be implemented to operate a different scales with direct parameterization from the applications. This approach helps XLayer to mitigate the issues associated with lagged adaptation between neighbors layers [6]. In some of our previous research efforts we have developed topology control algorithms manipulating transmission power in coordination with data flows (and message context) using fast response controllers [7], similar approaches have also been used by Gewali [8] and others. For this work, we rely on the same architecture to create an adaptive routing controller for cross-domain routing.

Figure 1 highlights the adaptive routing component (ARC) instantiated as a controller in the cross-layer architecture. ARC is responsible for bridging across heterogeneous networks by dynamically selecting gateway nodes. While active, gateways nodes will simultaneously run multiple routing protocols and handle the translation between each network, acting as a border gateway.

III. The Adaptive Routing Controller

As illustrated in Figure 1, ARC supports the registration of multiple routing protocols that may be activated or de-activated on demand. At startup time, a configuration file at each node lists the routing protocols that should be loaded and bootstrapped by the cross-layer. The bootstrapping process involves the registration of the protocol with the adaptive routing controller.

Upon registration, each protocol provides a simplified message parser and a control interface that enables the ARC to activate or de-active the protocol on demand. Furthermore, the protocol must be able to expose its routing tables, which allows the ARC to translate, as necessary, routes from one protocol into another. Such translation happens only on nodes acting as gateways (i.e. where multiple protocols are active).

A. The Gateway Selection Algorithm

The gateway selection algorithm is activated every time a routing packet is received. In this context, a gateway is a node that is simultaneously running more than one routing algorithm and bridging routes (and topology) from one algorithm to the other. Essentially, a node independently chooses to become a gateway if it receives a routing message associated with one of its inactive protocols, and if there is currently less than two 1-hop neighbors that are already gateways.

In its simplest form, a routing message received by ARC is checked against all registered parsers for a match [1]. If a match occurs for an active routing protocol, the message is normally handled. Conversely, a match to an inactive protocol may trigger its activation.

Input: A routing packet for protocol A: RP_A
Data: $Timestamps_A()$ – A hash table indexed by sender for holding timestamps of routing packets for protocol A
Data: $Timestamps_B()$ – A hash table indexed by sender for holding timestamps of routing packets for protocol B
Data: V_t – Validity time for routing packets
Data: T – Current time

$Timestamps_A$(getSender(RP_A)) $\leftarrow T$;
if $\neg$isUsingProtocol(A) **then**
  if $T - Timestamps_B$(getSender(RP_A)) $> V_t$ **then**
    $c \leftarrow 0$;
    foreach *node* **do**
      if $T - Timestamps_A(node) < V_t$ **then**
        if $T - Timestamps_B(node) < V_t$ **then**
          $c \leftarrow c + 1$;
        end
      end
    end
    if $c < 2$ **then**
      startProtocol(A);
    end
  end
end
if isUsingProtocol(A) **then**
  deliverPacketToProtocol(A, RP_A);
  if isUsingProtocol(B) **then**
    add1HopNeighborsToProtocol(B, getNeighbors (RP_A));
  end
end

Algorithm 1: Pseudocode of the Gateway Selection Algorithm

The criteria for protocol activation may be based on two independent strategies:

- A fixed number of gateways
- An adaptive number of gateways based on network density of the (routing) information received at each node.

The adaptive strategy may leverage from the local topology information provided by the cross-layer component. It may also be based on the routing information already available to each node. For the purposes of this work, we will focus on a fixed strategy based on a constant (maximum) number of gateways within a pre-defined number of hops from each other.

Based on the heuristics described in Algorithm 1, a node decides to become a gateway by activating an additional routing algorithm. At that point, the ARC component at the node is responsible for sharing routing information between the different protocols to enable cross-domain routing. In order to accomplish the sharing of routing information, the ARC must be able to query the routing table (or the topology, in some cases) from each routing protocol. Essentially, each gateway node must tell each routing protocol about the nodes seen with the other protocol, so routes can be established.

The process of sharing routing across protocols is critical for the performance of the systems and may be implemented in different ways. One approach, for instance, is to treat a gateway as a single border router and advertise a single route from each network into another, always based on the gateway as the entry point. This approach allows for the creation of global routes across all nodes in both networks and require some sort of IP deconfliction or address allocation algorithm such as those proposed in [9], [10], and [11].

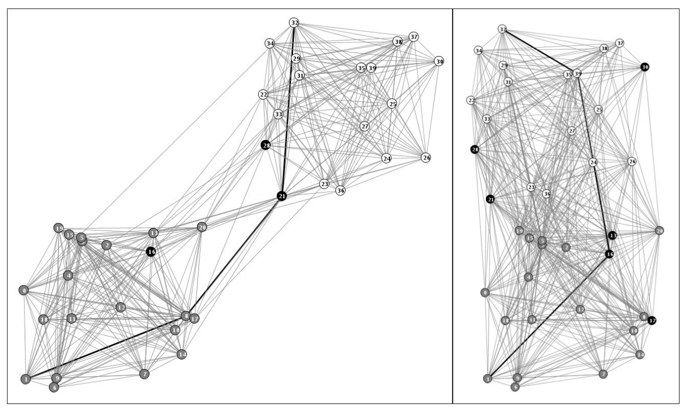

Fig. 2. The Experimental Scenario

A second strategy for sharing routing information between networks consists on the actual advertisement of routes between different protocols. At the local gateway node, this may be implemented by the ARC by directly mapping the routing tables between the different protocols - assuming that an API exists to allow the manipulation of such tables. A higher level of integration between algorithms may actually rely on having a common routing table shared by all algorithms, although such approach would likely require a complete re-implementation of the algorithms to share such critical information.

B. Proof-of-Concept Implementation

As a proof-of-concept, we have implemented the gateway selection and routing sharing algorithms as part of the XLayer communications substrate. To support the dynamic gateway capability, each routing algorithm was modified to provide the following capabilities: an interface for routing registration so

[1]In practice, and for efficiency purposes, the implementation assumes an active protocol handling all messages. A malformed (or unknown message) triggers the ARC component to identify a possible match and possible activate the gateway

that a simple parser would recognize (or not) a given message as its own type, an API for enabling and disabling the routing algorithm, an API for route advertisement, and an API to retrieve the routing table (with the associated costs of each route).

Upon registration of a second routing protocol, or upon routing table updates, the ARC queries the routing table from one protocol and advertises them into the other. Thus, a gateway node will advertise as 1-hop neighbors for protocol P the list of 1-hop neighbors seen with protocol P plus the nodes seen with the other protocol, which allows for full routes to be established across the different protocols.

The proposed approach will likely create the advertisement of redundant routes from multiple parallel gateways but since each route is advertised with their associated cost, they will be sorted by remote nodes accordingly. The costs of different routes are obtained from each of the protocols based on the topology of the network. From a local perspective, the effect of the proposed strategy is that a node in network A will perceive all nodes in network B as one-hope neighbors from the gateway with different link costs. In a link state protocol, where a localized (or global) network topology may be constructed at each node, the approach will not lead to an accurate topology representation of the remote network but will result in the actual representation of route costs, leading to an overall cross-domain routing path that approximates the theoretical path based on the full topology.

IV. Illustrative Scenario and Simulations

To illustrate the proposed algorithm we designed a simulation in NS-2 where two mobile network clusters running different routing algorithms come into contact for a pre-defined period of time. The proposed scenario can be mapped, for instance, to a military operation where friendly troops are crossing each other and are temporarily enabling services to one another. The underlying assumption is that each network is running a different routing protocol but at least some of the nodes on each network have the remote routing protocol registered (although disabled). The goals is to allow for an autonomous interface between the two networks by allowing their reciprocal detection, identification, and configuration for information exchange through a set of dynamically selected gateways using a fully distributed algorithm.

In the scenario illustrated in Figure 2, each network consists of 20 nodes, all operating in the same frequency range (a single radio system is assumed, to focus the discussion in the gateway selection algorithm)[2]

The top network moves to the left at a constant speed while the bottom network moves to the right. At different points in their trajectories, one or more nodes will come into RF contact with one another and routing messages will be exchanged. As previously described, such message will be captured by the ARC and directly relayed to the active routing algorithm for processing. If the message is unknown, the registered parsers are queried to identify the routing protocol associated with the message - assuming it is registered.

Upon a positive identification of the appropriate routing protocol, Algorithm 1 is invoked to decide if a node should become a gateway or not. The heuristics for route exchange are followed if the node decides to activate a second routing protocol (i.e. if it decides to become a gateway). In the proposed illustrative scenario both networks are running a link-state routing protocol, the bottom network is running HSLS (Hazy-Sighted Link State Routing Protocol) [12] and the top network is running OLSR (Optimized Link State Routing Protocol) [13]. The experiment was simulated using the NS-2 wireless network simulator.

In this example, the strategy for capping the number of gateways was based on a maximum and fixed number of gateways within range from one another. Essentially, in this example, a node will choose not to become a gateway if there is already a gateway for the candidate protocols within a 1-hop neighborhood of the node. Such strategy is reasonable for relatively dense networks.

A. *Analysis of Simulation Results*

Our preliminary analysis of the scenario focused on the packet delivery ratio and the delay induced for the establishment of an end-to-end route between the two networks. A sender node is located in one of the networks and is attempting to send messages to a receiver at the other remote network. Both sender and receiver nodes are chosen in such a way that they are not likely to become gateway themselves (i.e. they are remote to the boundary region).

In our tests we compared the scenario conditions (which we refer to as a *Hybrid* routing protocol since it combines both protocols in different regions) with the case where both networks are running HSLS, and both networks running OLSR. In our case, the variable of interest was the delay for the receiver node to actually receive the first data packet from the sender. Each condition (HSLS, OLSR and Hybrid) was tested in several times under the same conditions (simulation seed) in order to appropriately compare delays and standard deviations.

Figure 3 shows a comparison of the ratio do packets delivered for each protocol. For a single stream of 600 packets between two nodes in the network (as illustrated by the solid line in Figure 2), the packet delivery ratio is measured by the total number of packets reaching the destination node divided by the number of packets transmitted. The low ratio (below 60%) shown in the figure, is due to the fact that during some period of the simulation the networks are completely out of range from each other, and data packets are simply dropped. When within range, the packets received by the target node are counted for consideration, and the comparative performance of each protocol is show in Figure 3.

In our simulations, the hybrid protocol is shown to provide a statistically higher data delivery rate than the other protocols (with 95% confidence). It does, however, present a much

[2]The XLayer substrate, however, does provide other controllers for frequency adaptation that enables network to negotiate spectrum allocation for different applications. In such applications (not shown here to simplify the discussion) a scanner node detects activity in other frequencies and re-negotiates spectrum allocation with other nodes to either join a remote network, or move out of interference range.

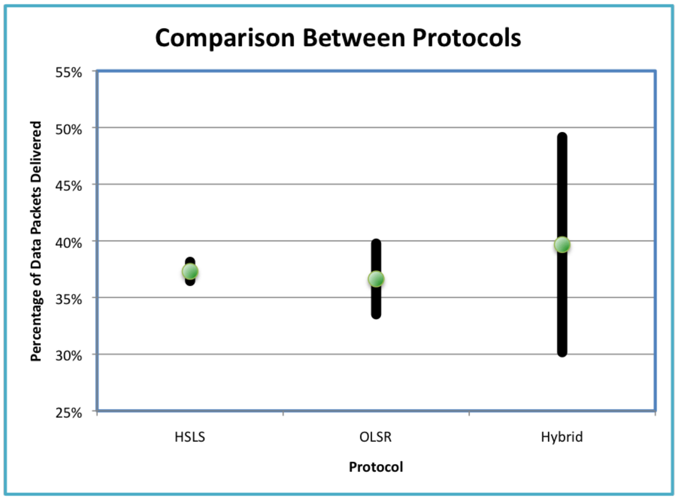

Fig. 3. Packet Delivery Ratio - Protocol Comparison

higher variance between different runs, which we believe is due to the fact that once a bad gateway is selected it remains active at least until it times out. Under some topology conditions, that may greatly affect the overall packet delivery. As part of our future work, we will investigate better evaluation and recover strategies to minimize these effects.

The overhead induced by the protocol is shown in Figure 4. The metric chosen for this comparison was the total number of routing messages (including discovery and recovery) used to maintain the data flow illustrated in Figure 2. As shown in these results, the overhead induced by the HSLS protocol is significantly higher than OLSR's routing overhead. That is, in part, due to the efficient MPR-based flood mechanism used by OLSR. As expected, the overhead of the hybrid protocol lies between the two, which is due to the fact that each side of the network continues to run its own native routing protocols after gateways are established, inducing their natural overheads.

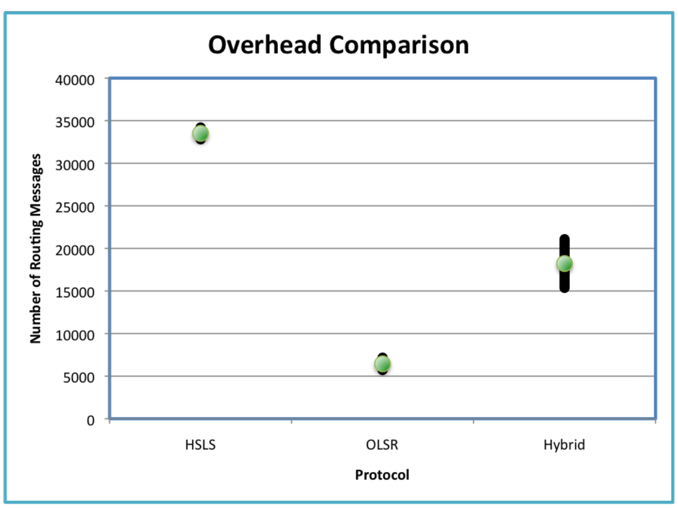

Fig. 4. Protocol Overhead Comparison

The route establishment latency induced by each condition is illustrated in Figure 5. The figure shows the average delay in each case, with its 95% confidence interval. The response time for HSLS was 5.24 (1.41), the delay for OLSR was 8.39 (1.91), and the delay for the Hybrid protocol was 9.94 (1.69), where the number in parenthesis represent the margin of error for each condition with a 95% confidence level. On average, HSLS is a more reactive protocol and responds faster to the new routes than OLSR.

The latency of the Hybrid protocol is bound by OLSR since routing advertisements still rely on that protocol for propagation across the combined network. At a 95% confidence level, there is no significant statistical difference between the delays incurred in OLSR and the delays induced by the Hybrid protocol, which seems to indicate that the delays associated with the detection and configuration phases of the proposed algorithm are not significant in comparison with the delays associated with route propagation and route updates.

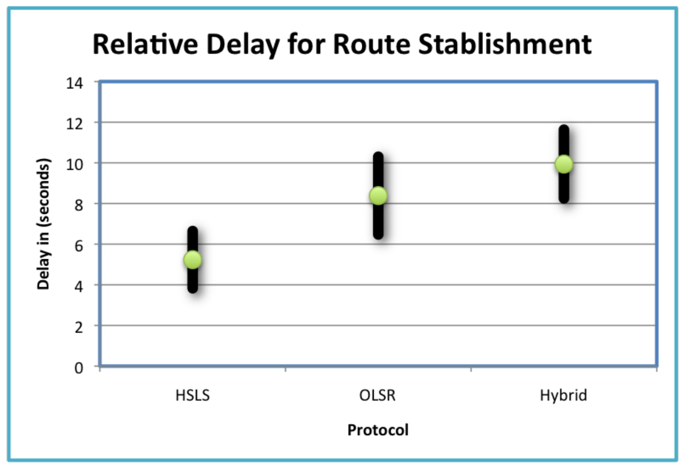

Fig. 5. Experimental Results - Delay in Receiving the First Data Packet (seconds)

Once created, a gateway node will be active while it receives messages from both protocols. If one of the protocols stops sending messages, maybe due to the node moving out of the intersecting region, the remaining routing protocol takes over as the primary, and eventually the only active routing protocol. There is a timeout associated with a gateway node that determines the inactivity period required to deactivate a routing protocol. The overall throughput of the hybrid protocol is affected by the choice of the timeout.

In Figure 6, the overall packet delivery ratio is shown as a function of the timeout factor, which is a function of the HELLO message intervals of both OLSR and HSLS. As illustrated in the figure, the higher throughput is around a factor of 2.5 which corresponds, in our simulation environment, to 2.5 seconds.

V. Limitations and Future Work

The current implementation was tested only for the case where two different routing algorithms are present. While we have no reason to believe that the algorithm would not extend to multiple simultaneous routing protocols, we must recognize

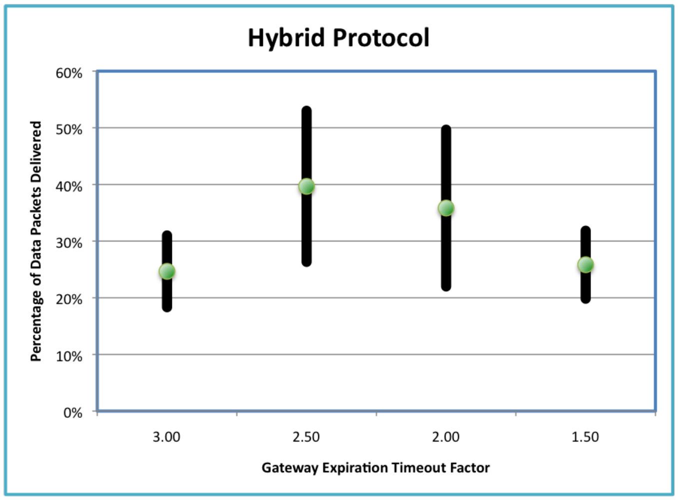

Fig. 6. Effects of the Gateway timeout factor in overall packet delivery

the complexity involved in the exchange of routes between protocols. The approach seems to be practical if combined with other control mechanisms that will limit the explosion of route exchanges between very large networks.

Alternatively, a variation of the proposed approach could be devised to include some horizon-based routing exchange, but that is left as future work. The mobility models used in our illustrative scenario were also simple in the sense that all nodes in each network moved at the same speed and direction so there were not many changes in topology within each cloud. A change in the mobility model would allow us to evaluate the combined effects of route discovery between the different networks and the propagation of route updates across networks. These effects are currently being analyzed and also constitute part of our future work.

VI. Conclusion

In this paper we have described and implemented a proof-of-concept algorithm for dynamic gateway selection for cross-domain routing. The algorithm was developed on top of the XLayer communications substrate, relying on cross-layer monitoring capabilities for traffic detection and route manipulation. While our results are still preliminary, we are encouraged by the fact that cross-domain routing could be achieved without any significant delay overhead and with relatively small changes in the implementation of standard routing protocols.

The cross-domain routing capability is not only relevant to enable communications across different networks (running different protocols) but also to provide the necessary adaptation for optimal utilization of network resources. The cross-layer substrate provides visibility not only to local network topology but also to traffic loads and priorities. An efficient hybrid routing algorithm may leverage that information to better adapt to different network conditions, using different routing strategies for different regions to better utilize and allocate resources.

Acknowledgment

This work was supported by the U.S. Air Force Research Laboratory under Contract Agreement number FA8750-07-018, the U.S. Army Research Laboratory under Cooperative Agreement W911NF-04-2-0013, and the Advanced Decision Architectures Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0009. The authors would like to acknowledge the help and support provided by Marco Arguedas, Massimiliano Marcon and Juan D. Estrada, all members of the Tactical MANET Research Group at the Institute for Human and Machine Cognition.

References

[1] M. Carvalho, A. Granados, W. Naqvi, A. Brothers, J. P. Hanna, and K. Turck, "A cross-layer communications substrate for tactical information management systems," in *Military Communications Conference (MILCOM)*, IEEE, Nov. 2008.

[2] F. Legendre, M. D. De Amorin, and S. Fdida, "Implicit merging of overlapping spontaneous networks," **60**(4), pp. 3050–3054, 2004.

[3] K. Chintalapudi and R. govindan, "Localized edge detection in sensor fields," in *IEEE International Workshop on Sensor Network Protocols and Applications*, May 2003.

[4] M. Carvalho, N. Suri, V. Shurbanov, and L. E., "A Cross-Layer network substrate for the battlefield," in *25th Army Science Conference (ASC)*, 2006.

[5] M. Carvalho, N. Suri, M. Arguedas, M. Rebeschini, and M. Breedy, "A cross-layer communications framework for tactical environments," in *IEEE Military Communications Conference (MILCOM)*, Oct. 2006.

[6] A. Goldsmith and W. S., "Design Challenges for Energy-Constrained Ad hoc Wireless Networks," *IEEE Wireless Communications Maganize* **9**(4), pp. 8–27, 2002.

[7] J. Granados, E. Montoya, and M. Carvalho, "Topology control mechanism in manets," in *Proceedings of Tecnocom 2006*, Nov. 2006.

[8] L. Gewali, K. Mohamad, and M. Tun, "Interference aware dominating set for sensor network," in *ITNG '06: Proceedings of the Third International Conference on Information Technology: New Generations (ITNG'06)*, pp. 268–273, IEEE Computer Society, (Washington, DC, USA), 2006.

[9] M. R. Thoppian, "A distributed protocol for dynamic address assignment in mobile ad hoc networks," *IEEE Transactions on Mobile Computing* **5**(1), pp. 4–19, IEEE Educational Activities Department, (Piscataway, NJ, USA), 2006. Member-Prakash, Ravi.

[10] M. Mohsin and R. Prakash, "IP address assignment in a mobile ad-hoc network," in *IEEE Military Communications Conference (MILCOM)*, October 2002.

[11] H. Zhou and L. M. Ni, "Prophet address allocation for large scale manets," in *IEEE Infocom*, April 2003.

[12] BBN, "Hazy Sighted Link State (HSLS) Routing: A Scalable Link State Algorithm," BBN Technical Memorandum 1301, BBN, August 2001.

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