

MINA: A Reflective Middleware for Managing Dynamic Multinetwork Environments

Zhijing Qin*, Luca Iannario[†], Carlo Giannelli[‡], Paolo Bellavista[‡], Grit Denker[§], Nalini Venkatasubramanian*

* University of California, Irvine, USA [†] HP Enterprise Services, Italy [‡] University of Bologna, Italy [§] SRI International, USA

Abstract—The networking landscape of today is characterized by diverse access technologies including cellular, WiFi, Ethernet, MANETs, and ZigBee, and properly managing this heterogeneous networking infrastructure is a key challenge to take full advantage of its many opportunities. In this paper, we propose MINA (Multinetwork INformation Architecture), a reflective (self-observing and adapting) middleware approach to realize and manage dynamic and heterogeneous multi-networks in pervasive environments. A novel aspect of MINA is that it embodies an Observe-Analyze-Adapt (OAA) loop to i) achieve a reasonably accurate, centralized global view of the multi-network through the design of novel techniques for overlay structuring, network state collection and formal methods-based analysis, and ii) take advantage of the global view for adapting multi-network structure by reallocating application flows across networks and proactively planning and deploying additional network resources.

I. INTRODUCTION

Technology advances in sensing, architecture and communication are creating a rich and complex networking scenario characterized by an increasing number of pervasive devices equipped with notable computational hardware, multiple communication interfaces, and diverse sensing capabilities. For instance, today's personal devices (smartphones, tablets) are multifunctional sensing, storage, computation, and communication platforms. Such devices utilize heterogeneous, often intermittent networks at the edge (ZigBee, Bluetooth, PANs, MANETs, IEEE DSRC, mesh) to interface with higher capacity, relatively fixed backbone networks (e.g., wired IP, 3G/4G, WLAN, WiMax, satellite).

Such multi-network scenarios represent a notable advantage for end consumers, service and network providers, and network administrators. However, pushing the communication envelope toward the impromptu exploitation of any and all available networks has its challenges, primarily due to the diverse nature of traffic and the distributed, dynamic nature of the multiple connectivities. In this paper, we design and develop a platform that manages and coordinates dynamic multi-network environments and enables rich applications to appropriately leverage these network capabilities.

Our past experiences dealing with heterogeneous networks [1], [2], [3], [4] have indicated that changes, such as those induced by mobility or newly sensed events, are frequent and can reduce communication reliability and information quality in traditional network architectures. Failures, such as those triggered by a natural disaster, can cause significant loss of connectivities [5] at critical moments. Often, network resources are provisioned in an isolated (for a single network) and mission-oriented manner, with little or no visibility for

the whole network topology or state. Recent efforts (e.g., Cisco Prime Infrastructure [6], HP OpenView [7], and Spider-Cloud [8]) aim to address the challenges of multi-networking and the integrated synergistic management of heterogeneous wireless networks. To date, these efforts have focused on exploiting lower-layer features specific to an access network or network-layer contexts, typically via new Layer 3 protocols [9], [10], [11], [12]. However, these solutions still manage networks in a low level, distributed, and vendor-specific manner, which is error-prone and inefficient.

In this paper, we propose MINA (Multi-network INformation Architecture), a middleware approach that realizes and manages dynamic and heterogeneous multi-networks in pervasive environments. A key aspect of MINA is that it implements an Observe-Analyze-Adapt (OAA) loop to guide the configuration, state management, and coordination of the multi-network, using available knowledge of network status.

II. MINA ARCHITECTURE AND OAA DESIGN PHILOSOPHY

While MINA realizes management functionalities similar to those of current network management platforms, it specifically focuses on addressing issues arising from heterogeneous networks in a pervasive computing environment. For instance, key tasks include performance management, configuration management, fault analysis and recovery, and network operations and security management. MINA's fundamental difference lies in its ability to perform the above tasks while handling the diversity and dynamic nature of the constituent network platforms.

To support scalability, MINA is designed as a **tree-based, hierarchical architecture** (see Fig. 1(a)). The higher levels of the tiered architecture are more stable (i.e., stationary) and resource-rich, aggregating information from other lower, more mobile, and less stable nodes. At the heart of the MINA system (Tier 1) is a logically centralized server (eventually implemented on a multi-node cluster) that collects and analyzes the network state information from each device. The centralized server allows to detect faults and performance degradation either explicitly (e.g., through thresholds on raw data values) or through inference processes further analyzing collected data, while not limiting the scalability of the system aggregating monitored information on lower tiers. Stationary, resource-capable nodes (e.g., those with Ethernet connections like routers, access points, and stationary PCs) are designated as Tier 2 nodes. Tier 3 typically consists of mobile nodes (e.g., smartphones, laptops, and tablets) that connect to Tier 2 nodes, either directly or by intermediate relay nodes, via

multiple kinds of wireless radio networks. The MINA multi-tier architecture exploits the diverse capabilities of the network nodes, more effectively supporting node mobility. In fact, upper tier nodes assume a larger fraction of the computation and communication workload, while less capable (and more dynamic) mobile nodes are relegated to lower levels of the tiered architecture.

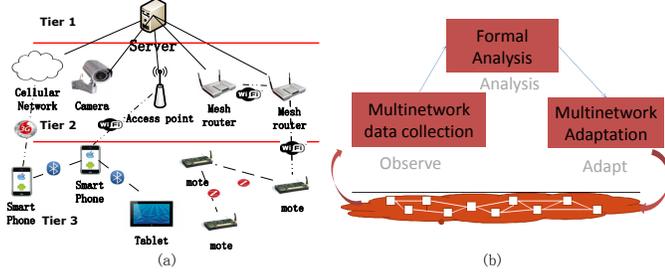


Fig. 1. Tier-based Architecture and OAA Paradigm

To capture and support dynamicity, MINA’s design is based on an **Observe-Analyze-Adapt (OAA) approach** (see Fig. 1(b)), in a manner similar to the OODA paradigm [13]. The novel aspect of MINA is that the autonomic decision-making process occurs at higher software levels enabling global multi-network management and a more synergistic exploitation of multiple communication opportunities for better global performance. A self-observing, introspecting system monitors the dynamically changing multi-network state, analyzes state information streams, and adapts the multi-network usage and configuration to ensure reliable communication functionality for the end applications. Observe, Analyze, and Adapt steps of the OAA approach are implemented through simple modules at the MINA client side (local state capture, local control for adaptations) and more powerful capabilities (global state assimilation, storage, analysis, logical adaptation etc.) at the MINA server.

A. Observe

The main objective of the Observe step is to generate a tree-based overlay network to collect network state information from various components in the multi-network environment.

To this purpose, MINA creates a **tree-based overlay network** following the tier-based approach (Fig. 1(a)) by dynamically evaluating nodes and placing them at the tier best fitting their capabilities. The central server resides at the first tier (the root at Tier 1) while a quantitative metric determines the tier of other node based on hardware, mobility, and role parameters (see additional details in Section II-C). The overall goal is to improve the reliability and efficiency of the whole overlay network. MINA implements *Initialization* and *Maintenance* procedures to create and maintain tree-based overlay networks; the key idea is to exchange messages to establish and maintain parent-child relationships.

The *Initialization* procedure is triggered by the root node to recursively create a new tree-based overlay network (establishing parent-child relations). Note that the first time an overlay network is generated, MINA skips the dynamic evaluation of tier values since it has few or no information to suitably compute them: it only uses the generation number to choose parents among candidates. Once a first overlay

network is established, a specific metric can be adopted to dynamically “migrate” devices among tiers, periodically re-evaluating nodes’ tier value and thus modifying the overlay network (see Section II-C).

The *Maintenance* procedure allows parents and children to verify the validity of parent-child relationships: if a new parent-child relation is found (or a current parent-child relation is lost), associated network management operations are invoked. Note that the most challenging part in overlay construction is how to establish the overlay path from clients to the server quickly without incurring much message overhead. This requires a model that can estimate the current path duration and next path available time, according to mobility model, nodes density, and traffic pattern. More details can be found in [14], [15].

B. Analyze

We propose to use **formal methods** to perform network analysis in a **proactive manner** exploiting information collected during the Observe step. Formal methods are a good match for proactive analysis because they apply abstraction to ensure capture of the system’s dynamic while keeping it abstract enough to allow automated tools to check critical conditions. For the proactive analysis of multi-networks we use the formal method Maude [16] [17], an executable specification language. Maude models system states through user-defined data types and system dynamics through rules. Moreover, not only the Maude interpreter is very efficient at simulating complex systems, but also provides efficient built-in search and model checking capabilities (see <http://maude.cs.uiuc.edu>).

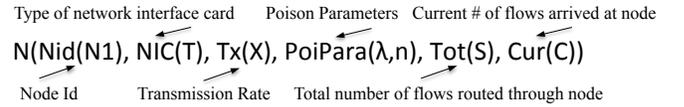


Fig. 2. Node Model.

The Maude representation views the network as a set of nodes and flows. Nodes are modeled as list of attributes and each attribute is represented as a label (value)-pair (see Fig. 2; variables in capital letters are place holders for values). Each flow is a sequence of subflows SF, one subflow for each hop along the flow’s path. Subflows are also modeled as list of attributes (see Fig. 3), including attributes for QoS parameters.

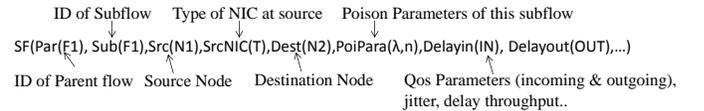


Fig. 3. Flow Model.

There are three main Maude rules that describe propagation of QoS characteristics of flows in the multi-network:

- *flow accumulation*: this rule collects all the subflows at one node. When multiple Poisson flows come into a node, we aggregate the λ and n values for all flows to calculate the overall delay (jitter and throughput) they have suffered, given the service capacity (the capacity/transmission rate of the network interface) of this node. Once the number of the accumulated flows reaches a predefined amount (indicated by the value of attribute *total*), it triggers the *flow processing* rule;

- *flow processing*: once all incoming flows are accumulated, the node has a complete set of values of λ and n . The node then calculates the new incurred delay via several rules as described in [17];
- *flow propagation*: once the outgoing flow is generated by the node via the *flow processing* rule, it will be passed into the next node as an incoming flow. This way, the delay value can be propagated from the source to the destination.

C. Adapt

The MINA multi-network environment is dynamic, characterized by more frequent and abrupt changes than traditional network management scenarios. At the network layer, MINA achieves the dual objectives of limiting power consumption while increasing the overall relay network's reliability by reactively modifying the collection tree, e.g., moving a laptop node running out of battery from upper to lower tiers. At the application layer, MINA avoids the dispatch of heavy application traffic, e.g., multimedia streams gathered from surveillance cameras, to the node. Proactive solutions in case of inter-flow interferences may include i) global rerouting and replanning of multiple flows (given inter-flow impact information) or ii) deployment of new routers by humans at optimal locations to improve current network performance. In the following, we describe how the MINA multi-network is adapted using specific cross-layer and proactive/reactive techniques.

Dynamic reorganization of overlay structure: Recall that overlay construction/maintenance mechanisms in Section II-A use the notion of a generation number to construct and maintain the overlay tree. This approach is necessary in initial stages when adequate network state information has not yet been collected. Once the network has been operational for some time, and additional information about the network becomes available (e.g., as a result of the Observe step), additional metrics can be dynamically adopted to better manage the network and achieve specific goals in relation to the deployment environment. To this purpose, we introduce the *tier value* ($TV=N(H, M, R)$) metric to dynamically assess the layer of nodes in the hierarchical overlay. TV returns real values in the $[0, 1]$ interval (lower values indicate more stable nodes) based on (a) the hardware capabilities of a device (H), (b) anticipated level of node mobility (M), and (c) the expected role of the node in the overlay structure (R). H, M and R can be assigned qualitative (categorical) and quantitative values. Categorical values are currently assigned as follows: $H \in \{Server/Dedicated\ Node, Desktop/laptop, Smartphone, Sensor\}$ and captures the hardware capabilities of a node; $M \in \{Fixed, MobilePlugged, MobileUnplugged\}$ and captures the expected mobility; $R \in \{TransitNode, LeafforTransit, LeafOnly\}$ and indicates the expectation that this node will dispatch packets belonging to other nodes. Note that while the current TV calculation exploits these attributes, it can be easily extended to incorporate more features to fit specific requirements.

In MINA, nodes dynamically compute and maintain up-to-date TVs for themselves based on local knowledge and information received from parents/children. We extend the *Maintenance* procedure in Section II-A adding two new

message types (*ParentUpdate* and *DescendantLeave*) in the overlay protocol to support the capability of spreading updated TV values and changing the parent of nodes.

QoS-Based MultiFlow Rerouting: Application flows may have various QoS needs and network infrastructure may change. Thus, the contention and interference among flows have larger consequence in multi-networks. We thus propose a novel solution to minimize this interference among flows. Traditional routing techniques handle this interference via state exchange: nodes reroute flows based on information received from local neighbors such as hop counts (Dijkstra algorithm), expected transmission time [18], and link position [19]. The route is changed if an improved route is calculated (using local metrics). However, typically used metrics do not account for the effect of the routing action on other flows or other node behaviors.

Here we propose a hybrid technique. The idea is not to design and implement a new routing protocol, but to choose the routes selected by an existing routing protocol as input to MINA. The formal-method based analysis will assess which routes have better globally optimal QoS performance. Assuming we have n flows, each flow will have a best route selected by the default routing protocol, denoted by r_n . Usually these best routes are chosen based on a local metric and might not be globally optimal due to inter-flow interference. To exploit better global QoS performance, we use the same routing protocol to generate the second best route for each flow, denoted by r'_n . There are a total of 2^n route combinations of best and second-best routes for all flows. Some preprocessing algorithms can greatly decrease this combination set, but such optimization is beyond the scope of this paper. For each flow combination, MINA determines overall QoS performance and gives recommendations on route selection for each flow that is globally optimal. Validation of this technique will be shown in Section III.

III. MINA IMPLEMENTATION AND EVALUATION

We have implemented a Java-based prototype of the MINA server and client - the initial implementation uses, on the server side, an Intel Core Duo 2.93 GHz, 4G RAM, Windows 7 with a MySQL 5.1 database to store collected information and Intel Core Duo 1.83GHz, 1G RAM, Ubuntu 10.04 for the client. The MINA client is also being currently ported to Android and MacOS platforms. On the client side, MINA extends the RAMP middleware (for dynamic sharing of resources available via heterogeneous links [4] [20]) to support the MINA tiered architecture and the client-server paradigm (additional information can be found in [15]).

The setting for experiments is an emulated network consisting of 29 nodes and 8 flows with different characteristics (Fig. 4). There are three types of links in this network with data rates of 100Mbps, 10Mbps and 2Mbps. Due to space limitations, we focus on results from the *Analyze* and *Adapt* steps - and assume that the network state information captured in the *Observe* step is already in the database (more details in [15]). In particular, our experiments aim to explore how formal methods based approaches within the OAA loop can help in improving performance via adaptive route planning for multiple flows.

route	flowid	type	TP	Item length	Arrival Rate
1,2,25	1	1	0.8	0.0016	500
1,5,26	2	1	0.8	0.002	400
19,7,3,11	3	1	0.8	0.0016	500
21,3,4,13	4	1	0.48	0.0016	300
1,6,10	5	2	0.48	0.0016	300
24,21,11,17,14	6	2	0.4	0.002	200
23,26,5,2,6,10	7	2	0.4	0.002	200
26,5,7,12,16	8	2	0.24	0.0008	300

Fig. 4. Network Flows

We explore how information from MINA's **Observe** step can help instantiate multipath routing and rapid path switching in multi-networks to enhance performance. To validate the *QoS-Based MultiFlow Rerouting* technique, we determine shortest and second shortest paths for the 8 flows in our emulated networks - this will yield 256 possible route combinations. We designate Combination 0 (00000000) as one that uses the default shortest path route for all flows; combination 5 (00000101) indicates flows 1 and 3 use the second shortest route while all other flows use the default route. Using our formal model described in Fig. 3, the **Analyze** step yields the delay for each flow for all combinations. The analyze step computes the degradation of the end-to-end delay and compares it to the best choice (i.e., the least degradation of this flow over all combinations). We ranked the combinations by their average end-to-end delay degradation over all flows. Our overall aim is to identify the best compromise route combination for all flows compared with their ideal delay performance. For example, combination 32 suffers an average degradation of 0.59%, as compared to the default combination 0, which has average degradation of 2.08% since there are more undetectable inter-flow interferences during local routing decisions. Given this analysis, MINA suggests that nodes involved with flow 6 (combination 32) should reroute the flow using the second shortest path instead of the shortest path picked by the local default routing protocol. The server uses the choices/hints in the analyze step to physically reroute the flows by sending control messages to involved nodes (**Adapt** step).

The interesting aspect here is that the reroute selection process can also be combined with other policies. In the above example, assume flow 2 is a delay-tolerant file sharing flow, while flow 3 is a video stream that requires low delay. In this case, we would like to choose the route combination with 0.0 delay degradation for flow 3 by sacrificing some delay performance on flow 2. The advantages of the formal method-based approach are: i) "globally best" is a user-definable specification, and ii) analyses are much easier to set up, carry out and adapt to user needs than network simulation tools. Network simulators like QualNet are more time-consuming and less adaptive since they usually need to simulate the packet/frame behaviors at all layers; the increased packet numbers require more calculation and time. To assert the best route in the above case, our technique only requires 27ms while QualNet requires 64ms. When we scale the traffic volume tenfold, QualNet's time increases to 81ms while our technique still requires only 27ms.

IV. CONCLUSIONS

In this paper, we have presented main aspects of the MINA multi-network management framework whose central, novel feature is the use of a reflective OAA approach. Through

implementation and simulation-based validation, we have illustrated that effective implementation of the OAA cycle is critical to the management of dynamic multi-networks. Techniques applied at each step of the OAA cycle will need to be different - in our environment, structure maintenance (through overlays) was key to the observation step; the use of formal methods based reasoning was critical to the analysis step and optimization techniques (e.g., flow control) are relevant to the adaptation step.

We are currently in the process of deploying MINA in a large campus-wide pervasive computing testbed (the Irvine Sensorium testbed) and developing more sophisticated analytical tools based on formal methods and other analysis techniques to assist on-the-fly analysis of data gathered in the MINA server database and further optimize achieved performance.

REFERENCES

- [1] "I-sensorium," <http://i-sensorium.ics.uci.edu/>.
- [2] "Safire," <http://cert.ics.uci.edu/SAFIRE/index.html>.
- [3] "responsphere project," <http://www.responsphere.org/>.
- [4] P. Bellavista, A. Corradi, and C. Giannelli, "The real ad-hoc multi-hop peer-to-peer (ramp) middleware: an easy-to-use support for spontaneous networking," in *ISCC10*, 2010.
- [5] C. Davison, D. Massaguer, and et.al, "Practical experiences in enabling and ensuring quality sensing in emergency response applications," in *PERNEM10*, 2010.
- [6] "Cisco prime infrastructure," <http://www.cisco.com/en/US/products/ps12239/index.html/>.
- [7] "Hp openview," <http://www8.hp.com/us/en/software/enterprise-software.html>.
- [8] "spider cloud wireless," <http://www.spidercloud.com/>.
- [9] H. Wu, Y. Liu, Q. Zhang, and Z.-L. Zhang, "Softmac: Layer 2.5 collaborative mac for multimedia support in multihop wireless networks," *Mobile Computing, IEEE Transactions on*, vol. 6, no. 1, pp. 12–25, 2007.
- [10] C. K. Toh, A.-N. Le, and Y.-Z. Cho, "Load balanced routing protocols for ad hoc mobile wireless networks," *Communications Magazine, IEEE*, vol. 47, no. 8, pp. 78–84, august 2009.
- [11] P. Frossard, J. de Martin, and M. Reha Civanlar, "Media streaming with network diversity," *Proceedings of the IEEE*, vol. 96, no. 1, pp. 39–53, jan. 2008.
- [12] H. Luo, R. Ramjee, P. Sinha, L. Li, and S. Lu., "Ucan: a unified cellular and ad-hoc network architecture," in *MOBICOM 2003*, 2003.
- [13] J. Boyd, "A discourse on winning and losing," *Maxwell Air Force Base, AL: Air University Library Document No. MU 43947 (Briefing slides)*, 1987.
- [14] Z. Qin, Y. Zhao, and N. Venkatasubramanian., "Modeling path duration time in dynamic convergecast network," in *IEEE WCNC*, 2013.
- [15] Z. Qin, L. Iannario, C. Giannelli, P. Bellavista, and N. Venkatasubramanian., "Smart communications via a tree-based overlay for multiple and heterogeneous spontaneous networks," in *SaCoNet*, 2013.
- [16] M. Clavel, F. Durán, and et. al, Eds., *All About Maude - A High-Performance Logical Framework, How to Specify, Program and Verify Systems in Rewriting Logic*, ser. Lecture Notes in Computer Science, vol. 4350. Springer, 2007.
- [17] Z. Qin, G. Denker, C. Talcott, and N. Venkatasubramanian, "Achieving resilience of heterogeneous networks through predictive, formal analysis," in *In CPS Week*, ser. HiCoNS '13. Philadelphia, Pennsylvania, USA: ACM, 2013, pp. 85–92.
- [18] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," *Wirel. Netw.*, vol. 11, no. 4, pp. 419–434, Jul. 2005. [Online]. Available: <http://dx.doi.org/10.1007/s11276-005-1766-z>
- [19] G. Jakllari, S. Eidenbenz, N. Hengartner, S. Krishnamurthy, and M. Faloutsos, "Link positions matter: A noncommutative routing metric for wireless mesh networks," *Mobile Computing, IEEE Transactions on*, vol. 11, no. 1, pp. 61–72, 2012.
- [20] P. Bellavista, A. Corradi, and C. Giannelli, "Middleware for differentiated quality in spontaneous networks," in *IEEE Pervasive Computing*, vol. 11, no. 3, 2012, pp. 64–75.