Smart Communications via a Tree-based Overlay over Multiple and Heterogeneous (TOMH) Spontaneous Networks

Zhijing Qin*, Luca Iannario†, Carlo Giannelli‡, Paolo Bellavista‡, Nalini Venkatasubramanian*
* Donald Bren School of Information and Computer Science
University of California, Irvine, USA
{zhijingq,nalini}@ics.uci.com
† HP Enterprise Services, Italy
luca.iannario@hp.com
‡ Dipartimento di Informatica - Scienza e Ingegneria (DISI)
University of Bologna, Italy
{carlo.giannelli, paolo.bellavista}@unibo.it

***Abstract*—The current networking scenario is characterized by widespread availability of ubiquitous devices with significant processing capabilities, e.g., smartphones, tablets, and laptops. In addition, the simultaneous availability of multiple connectivity interfaces, e.g., cellular, WiFi, and Ethernet, pushes towards spontaneous networking scenarios where devices create a multi-network environment based on collaborative and best-effort dispatching of packets. Connectivity in such novel scenarios is far less reliable than in traditional networks, e.g., links abruptly dis/appear simply due to node mobility, thus making hard to support quality-sensitive applications. In this paper, we present our framework for Tree-based Overlay over Multiple and Heterogeneous (TOMH) spontaneous networks for easily supporting smart network management features on top of heterogeneous multi-network environments. TOMH creates and maintains a dynamic and mobility-aware tree-based overlay network to integrate different connectivity technologies while enabling a tradeoff between accuracy of the global network view and collection/monitoring overhead. The TOMH overlay construction mechanism has been thoroughly validated and evaluated via simulation studies: the reported experimental results reveal that our proposal significantly outperforms comparable solutions for MANET environments, especially when the size of the targeted multi-network increases.**

***Index Terms*—Heterogeneous multiple networks; spontaneous networking; tree-based overlay; middleware.**

I. Introduction

Current and future networking scenarios are characterized by heterogeneity and ephemerality, encompassing devices and networks with diverse capabilities and in mutual communication range often in an unplanned way. Devices have been evolved in terms of both hardware, e.g., with multiple sensing capabilities and wireless network interfaces, and software, with the possibility to host complex algorithms, services, and applications. In this scenario, we define multi-networking as the dynamic interconnection of multiple and heterogeneous networks stemming from the impromptu and spontaneous collaboration among nodes. Multi-networks are intrinsically decentralized (each network is managed in a completely autonomous way) and can be based both on relatively fixed infrastructures, e.g., wired IP networks, WiFi deployments, and cellular infrastructures, and on intermittent connectivity, e.g., Mobile Ad-hoc NETworks (MANETs), and Bluetooth/Zigbee-based links.

Multi-network scenarios open tremendous new opportunities for users, device architects, and network/mobile service providers, by pushing the communication envelope toward the impromptu exploitation of any and all available networks. The potential advantages are significant from different viewpoints: network administrators would appreciate the possibility to satisfy a larger number of traffic requests with no need of additional equipment deployment; end-users are likely to enjoy the improved quality of experience from seamlessly and reliably accessing distributed services, e.g., Web and video streaming, due to the concurrent exploitation of multiple wireless interfaces, e.g., WiFi, Bluetooth, and UMTS.

Our past experiences dealing with heterogeneous networks in controlled instrumented environments (within the Irvine Sensorium infrastructure at UC Irvine, disaster response drills with local California agencies, and the RAMP spontaneous networking platform at U. of Bologna [1]) have yielded several observations [2]: a) Failures are likely to happen frequently in multi-networks, most of the times without the capability to diagnose the problem on the spot, reconfigure computers, swap/recharge batteries, or change cables. b) Current network deployments (WiFi infrastructures, cellular, etc.) are sensitive to noise. Even limited network noise can cause a significant drop in information quality especially for rich media data; in such cases exploiting alternative ad-hoc and spontaneous communication can be beneficial. c) Changes to the underlying network topology, especially those changes induced by mobility, further reduce reliability. To address these challenges, it is of primary important to efficiently and promptly spread state information among nodes in different networks.

Specifically, the paper proposes the Tree-based Overlay over

Multiple and Heterogeneous (TOMH) framework, whose main purpose is to make easier the smart management of multiple heterogeneous networks. TOMH creates and manages a tree-based overlay to proactively spread management information among nodes in interworking and heterogeneous spontaneous networks; in this way, it can be exploited to efficiently achieve a global view of network conditions, allowing to take more effective network management decisions based on the full knowledge of available nodes and their current connectivity capabilities. Only to mention one simple example, TOMH can observe novel communication links or link failures; remote endpoints can take advantage of this TOMH-based awareness to exploit alternative and more powerful paths as soon as they become available or avoiding to send packets along paths with broken links.

Many research activities have already investigated overlay construction in both wired [3], [4] and wireless networks [5], [6]. On the one hand, unstructured overlays [3], [7], [8] do not impose a rigid relation between the overlay topology and where resources or their indices are stored. In this manner, overlay networks are easier to implement even in dynamic environments, even if at the cost of limited scalability. On the other hand, structured overlay networks [4], [9] impose a structure on the overlay topology by setting routing table entries to fit certain criteria depending on the respective Distributed Hash Tables (DHTs), which bound the looking up complexity as O(log n). However, the TOMH overlay structure requires new features: first of all, packet transmission is always sent to/from the central server, without any pure peer-to-peer connection between any two arbitrary mobile nodes. Moreover, overlay network construction mechanisms should be lightweight to limit resource consumption on mobile devices. Finally, mobile awareness should be taken into account to promptly and appropriately reconfigure the overlay network.

In addition, most proposals investigating on integrating heterogeneous networks have so far focused on the exploitation of low-layer features specific to a single access network [10], [11], [12] or to particular domain/application contexts [13], [14], [15]. While targeting multi-access networks, the majority of research efforts have addressed network handoff [14] and specific combinations of network pairs (cellular/ad-hoc [16], cellular/WiFi [17], Bluetooth/WiFi [18]). We believe that addressing scalable management in heterogeneous and spontaneous multi-network environments at 3 and lower ISO OSI layers is unrealistic (it would require deployment of novel and compliant network equipment) and cannot support the rapid practical deployment needed in pervasive ubiquitous environments [19].

Just to anticipate a few notable aspects of the proposed solution, let us stress that our tree-based overlay network takes advantage of hierarchical information architecture and of an implemented middleware for managing communication in heterogeneous multi-networks. The overlay network allows nodes to make communication decisions locally (or higher up in the hierarchy), using available knowledge of network state and taking into account tolerance parameters (timing, accuracy, reliability). The advantages of the proposed approach and its strong originality if compared with the state-of-the-art primarily relate to the TOMH efficiency over large deployment environments, especially when dealing with global network management optimization goals in a lightweight and dynamic way, as better illustrated in the following.

The remainder of the paper is structured as follows. Section II presents the motivations and main design guidelines of our novel TOMH framework. Section III describes how TOMH captures and efficiently maintains network state information by exploiting a dynamically built and mobility-aware tree-based overlay. The TOMH framework has been thoroughly validated through Qualnet [20] simulations; related performance results are reported in Section IV.

II. TOMH Design and Philosophy

The multiple and heterogeneous spontaneous networks targeted by TOMH are typically dynamic and large-scale. To capture and support dynamicity, TOMH design is based on an Observe-Analyze-Adapt (OAA) approach: a self-observing introspecting system monitors the dynamically changing multi-network state, analyzes the streams of state information, and adapts the multi-network usage and configuration to ensure reliable communication functionality for applications. To address scalability issues, we adopted a tree-based overlay organization (tiered and hierarchical architecture) where nodes higher up in the hierarchy aggregate and abstract away the details of the underlying network.

A. *TOMH Tier-Based Architecture*

The underlying TOMH infrastructure consists of a large number of heterogeneous devices, interconnected by links with different capabilities; typically, Ethernet devices are stationary and have a plug-in constant power supply while wireless devices are mobile and battery operated. To support scalable management and exploit diverse characteristics of TOMH components, we propose a tier-based architecture (see Fig. 1) where nodes logically reside at different network levels/tiers. The selection of the logical level/tier is based on hardware/software capabilities and types of connectivity associated to nodes: nodes residing in higher levels are typically more stable and resource-richer, serving as aggregation points for information delivered by lower, more mobile, and less stable nodes. At Tier 1, the TOMH root, there is a centralized server in charge of collecting state information from devices; stationary, resource capable nodes, e.g., routers, access points, and PCs, reside at Tier 2; finally, mobile nodes, e.g., smartphones, laptops, tablets, are designated to Tier 3. Multiple servers may be present in Tier 1, thus possibly generating a multi-tree topology of disjoint sets of nodes, suitable for very large networks (in this paper we will only consider the case of single tree for the sake of briefness).

It is worth noting that Tier 3 nodes are connected to Tier 2 ones either directly or by intermediate relay nodes, via multiple kinds of wireless radio networks. Moreover, Tier 3 nodes are likely to abruptly leave the network, either willingly due to node mobility or unwillingly due to low battery level. However, since these nodes reside in the bottom tier, TOMH can effectively manage their departure very efficiently, promptly reconfiguring the tree-based overlay network (additional details in the following).

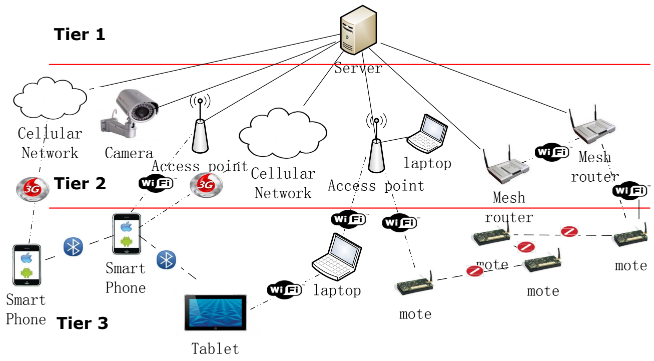


Fig. 1. Tier-based Architecture.

B. Tree-based Overlay Guidelines

Creating an overlay on the underlying heterogeneous topology of nodes/links allows us to maintain the collection topology at low cost, especially when mobility of nodes comes into play. Note that the overlay approach is based on the RAMP middleware [19], easing the task of integrating multiple networks regardless of the underlying heterogeneous link layer technologies and enabling TOMH to naturally use RAMP Node UIDs rather than IP addresses to organize nodes.

Our solution is based on three design criteria/goals for the overlay construction protocol that determines where nodes and links are positioned in the overlay structure. Lower overhead is the first design criteria. Due to the limitation of mobile devices, it is important to make sure adding new nodes to the overlay incurs a limited overhead by involving only a limited number of nodes already in the overlay. In fact, the performance of the overlay should not dramatically change when the number of nodes in the network grows. Secondly, promptness. The overlay should rapidly react to changes in the underlying network topology (mobile nodes moving around cause great dynamicity) so that it does not incur in large end-to-end delays on applications running above the overlay. Third, mobility awareness. The created overlay must accommodate node mobility and intermittent connectivity that is characteristic of mobile nodes. We observe that the tree-based overlay structure fits well with mobility management in the tiered architecture, since it pushes the more dynamic nodes towards the leaves.

The above three criteria potentially conflict with each other since there is an inherent tradeoff between these goals. In traditional wireless networks, nodes implement a periodic heartbeat mechanism to discover changes in network topology, i.e., joining and leaving of neighbors. Frequent heartbeats make easier the accurate discovery of paths, also decreasing the end-to-end delay of collected data, but obviously with higher costs in terms of network resources. Prior work has explored a broad range of techniques (theoretical analysis [21], machine learning [22], [23], mobility prediction [24]) to adaptively tune the performance under such tradeoffs in ad-hoc sensor networks. However, there are several new challenges in supporting such tradeoffs in the TOMH overlay: a) The convergecast pattern for data collection indicates nodes primarily care about maintaining a path to their parents. In contrast, solutions developed for MANET routing are intended for communication between two random nodes and hence are not suitable here. b) In heterogeneous networks, some nodes are stationary and some are mobile; as a result some links are inherently more stable than others. c) Nodes in Tier 3 are mobile and resource-limited: to avoid frequent overlay network modifications and limit their power consumption, they should not perform management tasks.

III. Overlay Construction in TOMH

Prior to present the tree-based overlay construction protocol, we first define notations, data structures, and message types used in TOMH overlay construction. The following functions are defined on a given node with unique identifier UID. *Neighbor* represents a node within a one-hop distance from a node UID. *IsParentCand* is true if the neighbor has offered to be a parent for a node UID (parent candidate). *IsParent* is true if the neighbor is currently the parent of this node, whereas *IsChild* is true if the neighbor is currently a child of this node. Furthermore, we store a *DescendantList* where each entry represents the set of descendants for a specific child of this node. *Generation* is an integer that is assigned to each node of the overlay, which is incremented by one from parent to child. The source node (root) has a generation number 0.

To better understand how the algorithm works, we also list and explain a set of important messages used in TOMH between two possible A and B nodes. *Parent Claim Broadcast*: Once A finds a new parent it will broadcast this message to claim itself as a potential parent of its neighbors. B marks A as a parent candidate when B receives this message. The source node performs the broadcast by default. *Parent Request Unicast*: B selects the best parent (based on some criteria, e.g., lowest generation number as default) among parent candidates, and sends a parent request message to this selected best parent. *Parent Confirm Unicast*: When A receives the parent request message from B, if there is enough space to admit a new child, A will send a parent confirm message to B. Otherwise A will send a ”parent refuse” message to B. *Parent Request Broadcast*: If B loses connectivity with its parent and currently there are no available parent candidates, it will broadcast a parent request message to find a new parent candidate. *Parent Accept Unicast*: when A receives a parent request broadcast message, if there is enough space to admit a new child, A will send parent accept message to B and B will mark A as parent candidate. *Descendant Update Unicast*: when B admits a new child, it will send the ID of the new child to its parent; therefore, each node knows all its descendants.

During overlay construction, appropriate runtime data structures are initialized and the overlay construction procedure is initiated. Overlay maintenance executes mechanisms to handle dynamic changes to the existing overlay, e.g., node mobility or node failure may trigger parent re-selection.

A. Overlay initialization and maintenance

Initialization is structured into two phases. In the first phase, nodes are iteratively added into the overlay from source node (root) to leaf nodes. The procedure begins with broadcasting parent claim messages by the source node. Nodes that receive parent claim messages from parent candidates will respond with parent request messages to the "'best parent candidate"; when the parent confirmation message is received, the parent-child relationship is established and the node is added into the overlay. This process is repeated until propagated to the leafs of the tree. The second phase starts when a node has already chosen a parent and advertises itself to accept new children. When a node admits a new child, it will send a Descendant Update message to its parent; the parent then adds a record to its DescendantList and forwards the message to the upper level parent, until the root is reached. Note that a node only knows the UID of its children and all the descendants of each child; it does not need to know the exact topology of its descendants in the subtree. Hence, our proposal does not require to consume too much memory and bandwidth to maintain a complete topology map in each overlay node. For example, in Fig. 2 Node S only knows that nodes 3, 4, and 5 are below Node 2 (its children) but does not know in which order. To send a message to Node 5, Node S only needs to know that the next hop node is Node 2, while the latter only needs to know that it should forward the message to Node 3, and so on. Moreover, solid lines in Fig. 2 indicate handshake procedures including Parent Claim, Parent Request, and Parent Confirm messages, while dashed lines indicate only Parent Claim messages.

Once overlay construction is completed, data collection is underway. Because of possible node mobility and failures, additional mechanisms are required at runtime to dynamically react to changes in the underlying topology (overlay maintenance). Two key issues to address are parent loss and children join. Every node periodically broadcasts a heartbeat message to its children. If a node does not receive a heartbeat message from its parent within a timeout, it will infer that its parent is not reachable anymore and will delete it from the neighborlist. If the node has other parent candidates, it will send Unicast Parent Request message to the best candidate; otherwise it will broadcast a parent request until a new parent is determined. Based on the generation number of the new parent, this node will decide to keep the parent-child relation with its current children or not (via explicit notice to children). By doing so, we can avoid the so called routing loop problem.

Child joining/leaving events are handled as follows in TOMH. When a new child joins the overlay, the parent will add it into its children list immediately and propagate this information upwards. To limit the number of propagated upward messages, we exploit the following mechanism: when a node leaves the old parent and joins a new one, the Descendant Update message is propagated only until it reaches the most recent ancestor between the new and the old parents. When a node receives the Descendant Update message, it will check whether this newly joined descendant already existed in other DescendantLists or not. If yes, this parent is the most recent ancestor and stops reporting this information further. Fig. 3 depicts the procedure: at time t1 Node 5 initially has a link to Node 3; when later (at t2) it joins Node 4, it updates its parent from 3 to 4; the descendant update is propagated only until Node 2.

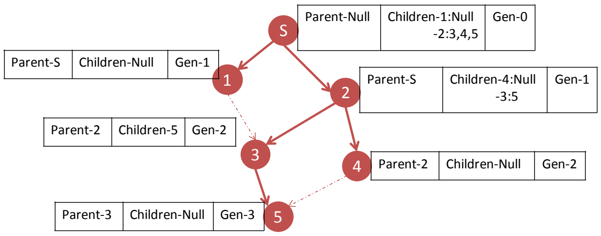

Parent-Null | Children-1:Null -2:3,4,5 | Gen-0
S
Parent-S | Children-Null | Gen-1
1
2
Parent-S | Children-4:Null -3:5 | Gen-1
Parent-2 | Children-5 | Gen-2
3
4
Parent-2 | Children-Null | Gen-2
Parent-3 | Children-Null | Gen-3
5

Fig. 2. Overlay Initialization.

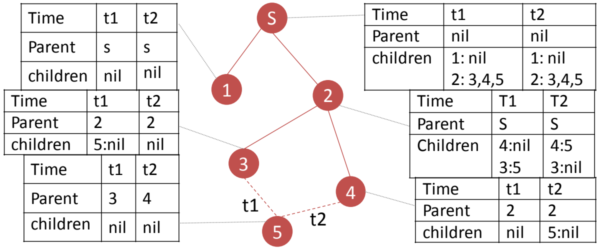


Fig. 3. Overlay Maintenance.

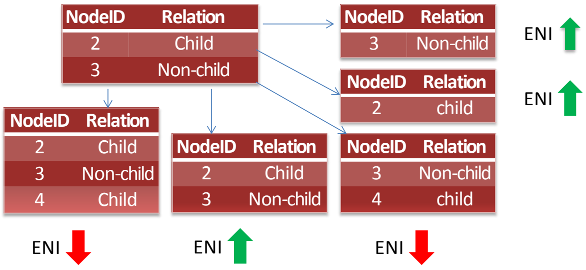


Fig. 4. ENI Policy.

B. Optimizing Overlay Maintenance with Expected Next Interval Messages

The basic TOMH overlay construction and maintenance mechanisms ensure that the source node can collect information about all nodes and that each node has an accurate image of its descendants at runtime. In this section, we present our enhancements to the basic TOMH overlay construction process in order to achieve the appropriate delay/overhead tradeoff in state collection. Recall that parent nodes use heartbeat messages to inform children of their existence; child nodes use a periodic neighborlist flush function to delete stale parent connections. On the sender side, increasing the rate of heartbeat messages will reduce end-to-end delays in data collection; however, the increased rate introduces additional messaging overhead. At the receiver end, a high neighborlist flush frequency will get rid of old routes; however, valid routes with longer lifetimes may also be deleted, which can result in increased collection delays.

To achieve the above goals, we have introduced a dedicated field to the heartbeat message, *Expected Next Interval* (ENI), that can provide hints to the recipient on when to expect the next heartbeat message. Upon receiving a heartbeat, a recipient uses the ENI as a timeout factor to determine when to flush the neighbor entry. The ENI is incremented/decremented at the sender side based on link dynamicity; the rate of increase/decrease can be tuned, also dynamically, to meet application requirements (possibly changing at provisioning time). For example, to support low end-to-end delays as compared to lower overheads, we employ slow increase and fast decrease functions as follows:

$$Increase: \quad ENI_{current} = ENI_{current} + Delta;$$
$$Decrease: \quad ENI_{current} = Max\{ENI_{current}/2, ENI_{def}\};$$

where the initial value of $ENI_{current}$ is ENI_{def}. ENI changes are triggered when there are changes to the set of children (of a parent node). Specifically, we decrease ENI when the child set changes and increase ENI when the set of children of a parent node is relatively stable. The rationale is as follows. When a new child joins a parent, there is a reasonable likelihood that the child is mobile and will hence leave soon, triggering more changes. When a current child leaves, it is likely that the sender (parent) is mobile and, if so, other children are likely to leave as well. Fig. 4 depicts a simple example where the receiver side uses the ENI information in the heartbeat to determine ”delete” or ”keep” actions for entries in the neighborlist.

IV. TOMH Overlay Evaluations

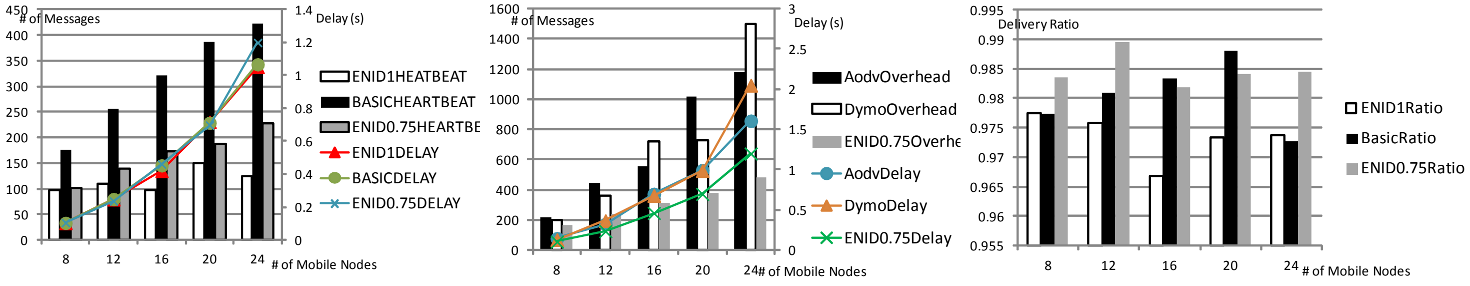
We have conducted extensive experiments to evaluate our tree-based construction protocol, with and without ENI-based enhancements. By using QualNET[20] as the simulation platform, we compare the performance of our overlay with more traditional AODV[5] and DYMO[6] approaches, well recognized and widespread in MANET scenarios. In all our simulations we have used one fixed node that acts as source (server), 4 fixed nodes that act as Tier 2 nodes, and several mobile nodes scaling from 8 to 24 as Tier 3 nodes that move around by following the random waypoint mobility model with speed ranges [1m/s,2m/s] and [10m/s,20m/s]. Each mobile node is configured to generate Constant Bit Rate traffic (periodically sending a 500 byte message to the server), thus emulating network state collection. Each simulation round has a 250s duration. We evaluate our techniques using three metrics: application end-to-end delay, message overhead, and delivery ratio. For ENI, we set $ENI_{def} = 7s$ and $Delta = 0.75s/1s$; motivations of this parameter settings are in the following.

Fig. 5 shows the results collected from the simulations under the random way point mobility model with speed range $[10, 20]$. Fig. 5(a) shows the heartbeat message overhead and end-to-end delay comparison between different ENI $Delta$ values. With ENI enabled, the heartbeat message overheads are greatly reduced (from 70% to 42%) as compared to the basic version of the TOMH protocol; note that the end-to-end delay (with an average deviation of 1.2%) and delivery ratio (max. deviation of 1.6%) are hardly impacted by ENI modifications, as shown in Fig. 5(c). If we compare the end-to-end delay between ENI with $Delta = 0.75s$ and AODV/DYMO, our ENI-based algorithm reduces the message overhead a lot as compared to both AODV (from 60% to 26%) and DYMO (from 68% to 20%), as shown in Fig. 5(b). Similar results have been obtained under the same mobility model with lower speed range $[1, 2]$. In Fig. 6(a), enabling ENI with $Delta = 0.75s$ can reduce the message overhead by a range from 53% to 37% and the range is even higher with $Delta = 1s$, which is from 78% to 44%. The end-to-end delay is not changed much (with an average deviation of 5%) and the experienced delivery ratio is almost the same (with a maximum deviation of 2%), shown in Fig. 6(c). Also if we use ENI with $Delta = 1s$, our overlay construction protocol generates much less messages than AODV and DYMO, as shown in Fig. 6(b). Note that although the TOMH overlay techniques do not explicitly focus on reliable state collection, our solution inherently achieves high delivery ratios of over 96.5%.

As a general final consideration, no matter there is data to exchange among nodes or not, TOMH always initializes and maintains the overlay for multiple purposes. So, the queue delay (which is a major part of the end-to-end delay) is unlikely to happen. More specifically, the broadcast Parent Request message and the ENI-based enhancements can improve the end-to-end delay further. Fig. 5(b) and Fig. 6(b) show that TOMH can provide a very small end-to-end delay (around 65%) compared to AODV and DYMO (which are on-demand), and even with much less message overhead. We can find that our overlay, without and with ENI ($Delta = 0.75s$), has proportional heartbeat message overhead to the number of mobile devices, shown in Fig. 5(a) and Fig. 6(a). More interestingly, we also find that, when enabling ENI with $Delta = 1s$, the message overhead is almost constant: the reason is that when $Delta$ exceeds a threshold, the message sending interval is big enough that, when the node is about to send the next message, it misses some children changes, ans then the ENI continues to increase (which may lead to inaccurate route and packet loss). Let us conclude by stating that the choice of ENI_{def} and $Delta$ cannot be arbitrary, since it determines the default heartbeat rate in the baseline TOMH protocol. These values depend on average link duration in the multi-network, which is a function of the number of nodes, mobility model, and transmission ranges. We estimate link duration times based on input from simulation scenarios: our measurements yield a default ENI value of 7s. The $Delta$ factor can serve to further tune the delay/overhead tradeoff. To compare the overhead of ENI techniques with the basic TOMH protocol, we have tuned the $Delta$ and let it achieve a similar end-to-end delay/delivery performance with the basic one. This is the methodology with which we have chosen 0.75s and 1s as the two different $Delta$ values in our experiments. The results from Fig. 5.(a) and Fig. 6.(a) illustrate these points further.

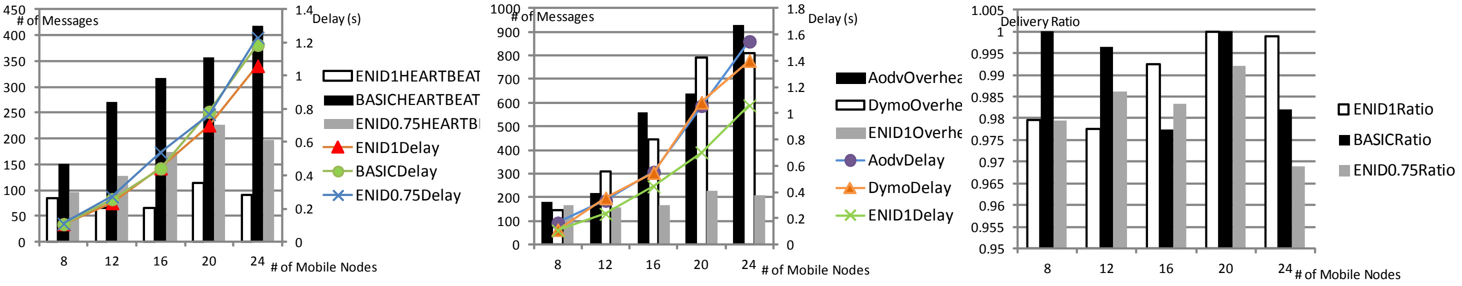
V. Concluding Remarks

This paper presents the design, extensions/enhancements, and simulation-based evaluation of the TOMH solution. A



(a) Overhead and Delay with Different ENIs (b) Comparison with AODV and DYMO (c) Delivery Ratio with different ENIs

Fig. 5. Performance Evaluation and Comparison for speed range [10,20].



(a) Overhead and Delay with Different ENIs (b) Comparison with AODV and DYMO (c) Delivery Ratio with different ENIs

Fig. 6. Performance Evaluation and Comparison for speed range [1,2].

key aspect of the TOMH approach is the development of a novel, dynamically constructed, and mobility-aware tree-based overlay structure that can effectively balance end-to-end data collection delay and overhead.

The encouraging results achieved so far are stimulating our further research activities along two main directions. On the one hand, we are developing more sophisticated analytical tools based on formal methods that can help with the effective on-the-fly analysis of TOMH gathered data. On the other hand, we are working on dynamic adaptation techniques for further tuning multi-network state collection parameters (with a better and more self-adaptive tradeoff between overhead and state staleness, decided at runtime depending on current application-level requirements).

References

[1] 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.

[2] C. Davison, D. Massaguer, L. Paradi, M. R. Rahimi, B. Xing, Q. Han, S. Mehrotra, and N. Venkatasubramanian, "Practical experiences in enabling and ensuring quality sensing in emergency response applications," in *PERNEM10 held in conjunction with IEEE PERCOM 2010*, 2010.

[3] C. L. Klemm, A. and O. Waldhorst, "A special-purpose peer-to-peer file sharing system for mobile ad hoc networks," in *Vehicular Technology Conference*, 2003.

[4] e. a. Castro, M.C., "Performance evaluation of structured p2p over wireless multi-hop networks," in *Sensor Technologies and Applications*, 2008.

[5] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (aodv) routing," RFC, 2003.

[6] P. C. Chakeres I, "Dynamic manet on-demand (dymo) routing," RFC, 2008.

[7] R. S. Gruber, I. and W. Kellerer, "Performance evaluation of the mobile peer-to-peer service," in *Cluster Computing and the Grid*, 2004.

[8] W. Kellerer and R. Schollmeier, "Proactive search routing for mobile peer-to-peer networks: Zone-based p2p," in *ASWN*, 2005.

[9] e. a. Caesar, M., "Virtual ring routing: Network routing inspired by dhts," in *ACM Sigcomm*, 2006.

[10] H. Bany Salameh and M. Krunz, "Channel access protocols for multihop opportunistic networks: challenges and recent developments," *Network, IEEE*, vol. 23, no. 4, pp. 14 –19, july-august 2009.

[11] 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, jan. 2007.

[12] M. Campista, P. Esposito, I. Moraes, L. Costa, O. Duarte, D. Passos, C. de Albuquerque, D. Saade, and M. Rubinstein, "Routing metrics and protocols for wireless mesh networks," *Network, IEEE*, vol. 22, no. 1, pp. 6 –12, jan.-feb. 2008.

[13] 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.

[14] S. Mohanty and I. F. Akyildiz, "Performance analysis of handoff techniques based on mobile ip, tcp-migrate, and sip," *IEEE Transactions on Mobile Computing*, vol. 6, no. 7, pp. 731 –748, july 2007.

[15] 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.

[16] H. Luo, R. Ramjee, P. Sinha, L. Li, and S. Lu., "Ucan: a unified cellular and ad-hoc network architecture," in *MOBICOM 2003*, 2003.

[17] J. Pang, B. Greenstein, M. Kaminsky, D. McCoy, and S. Seshan., "Wifi-reports: Improving wireless network selection with collaboration," in *MobySis*, 2009.

[18] G. Ananthanarayanan and I. Stoica., "Blue-fi: Enhancing wi-fi performance using bluetooth signals," in *MobySis*, 2009.

[19] 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.

[20] "Qualnet," www.scalable-networks.com/.

[21] F. Ingelrest, N. Mitton, and D. Simplot-Ryl, "A turnover based adaptive hello protocol for mobile ad hoc and sensor networks," in *MASCOTS '07*, 2007, pp. 9–14.

[22] M. Di Francesco, K. Shah, M. Kumar, and G. Anastasi, "An adaptive strategy for energy-efficient data collection in sparse wireless sensor networks," in *EWSN'10*, 2010, pp. 322–337.

[23] V. Dyo and C. Mascolo, "Efficient node discovery in mobile wireless sensor networks," in *DCOSS 2008. LNCS*. Springer, 2008, pp. 478–485.

[24] X. Li, N. Mitton, and D. Simplot-Ryl, "Mobility prediction based neighborhood discovery in mobile ad hoc networks," in *Proceedings of the 10th international IFIP TC 6 conference on Networking - Volume Part I*, ser. NETWORKING'11, 2011, pp. 241–253.