

Context-aware Middleware for Reliable Multi-hop Multi-path Connectivity

Paolo Bellavista, Antonio Corradi, Carlo Giannelli

Dip. Elettronica, Informatica e Sistemistica - Università di Bologna

Viale Risorgimento, 2 - 40136 Bologna - Italy

Phone: +39-051-2093001; Fax: +39-051-2093073

{paolo.bellavista, antonio.corradi, carlo.giannelli}@unibo.it

Abstract. The widespread diffusion of portable devices with multiple wireless interfaces, e.g., UMTS/GPRS, IEEE 802.11, and/or Bluetooth, is enabling multi-homing and multi-channel scenarios, possibly made up by multi-hop cooperative paths towards the traditional Internet infrastructure. There is the need for novel middleware supports, aware of innovative context information, to select and dynamically re-configure the most suitable interfaces and connectivity providers for each client application. In particular, novel middlewares should effectively exploit concise and lightweight context indicators about expected node mobility, path throughput, and energy availability to take proper connectivity management decisions at session startup and to promptly re-configure them with limited overhead at runtime. Here, we present how our MMHC middleware originally uses mobility/throughput/energy context to manage connectivity opportunities effectively, i) by filtering out connectivity opportunities that are considered insufficiently reliable, and ii) by carefully evaluating the residual candidates in two distinguished local/global management phases to achieve the most suitable tradeoff between promptness and management costs.

1 Introduction

Nowadays mobile devices, usually equipped with multiple wireless interfaces, can get connectivity to the traditional wired Internet by taking advantage of multiple connectivity opportunities provided by many infrastructure-based components, which tend to be ubiquitously available, e.g., IEEE 802.11 Access Points (APs) or UMTS Base Stations (BSs). In the following, we will call these connectivity components as *infrastructure connectors*. In addition, the increasing and increasing resources of mobile terminals potentially enable novel and more complex scenarios where client nodes can also help other clients to achieve Internet connectivity in a peer-to-peer fashion, e.g., via Bluetooth Personal Area Network (PAN) or IEEE 802.11 Independent Basic Service Set (IBSS) connections, by acting as intermediate entities in a multi-hop (possibly heterogeneous) path towards the Internet. We use the term *peer connectors* to indicate these novel connectivity opportunities. Peer connectors are in charge of creating and properly managing a simple and small Mobile Ad-hoc Network (MANET) with the peers in proximity and of correctly routing packets between their MANET and the Internet by exploiting the near infrastructure connectors.

The increased complexity of this scenario enabled by the concurrent exploitation of infrastructure/peer connectors is widely counterbalanced by the potential benefits of exploiting a significantly wider set of heterogeneous connectivity opportunities, among which to dynamically choose based on system/user/node/application-specific requirements, e.g., load balancing available connectors, always exploiting connectivity opportunities that are for free, preserving node battery, or respecting bandwidth requirements, as exemplified in Section 2. Of course, it is inappropriate to leave to client application designers the whole burden of properly managing the wide set of Multi-hop Multi-path Heterogeneous Connectivity (MMHC) opportunities that are dynamically available. Therefore, we claim the crucial role of client-side middleware solutions for effective MMHC management.

These middlewares should have effective visibility of different kinds of innovative context data to take proper MMHC decisions, especially to ensure usability of enabled MMHC opportunities by selecting the ones expected to be more reliable during the service session that will be established. In particular, lightweight estimations about client mobility (with regards to both fixed infrastructure and mobile peer connectors) could allow to exclude the connectors that are probably going out of the coverage area of the considered client soon, thus reducing the space of connectivity opportunities to take into account. Similarly, context data about the estimated throughput achievable by a single wireless hop and by the multi-hop path composed by that hop can help filtering out connectivity opportunities that do not comply with session quality requirements. Finally, context data about the residual energy of involved connectors could help in balancing energy consumption among connectors and in taking proactive re-configuration operations of currently exploited paths if some composing hops are expected to fail soon due to power exhaustion.

According to these context awareness needs, we have designed and implemented our innovative middleware for multi-hop multi-path connectivity management, called MMHC [1]. MMHC properly handles different kinds of context data, from user preference profiles to application requirements, from Received Signal Strength Indications (RSSI) for mobility estimations to battery power indicators, to select and dynamically re-configure the most suitable MMHC opportunity for each running application. In particular, in this paper, we originally focus on how MMHC portably gathers mobility/throughput/energy context data and exploits them to perform lightweight connectivity management. The primary ideas are i) of exploiting context data to reduce the space of potential candidates for selected connectivity opportunities and ii) of splitting management operations into a local phase (where mainly local context is exploited to achieve rapid, effective, but sub-optimal MMHC decisions) and a global phase (where lightweight distributed context guides proactive path re-configuration and procedures for role switch to counteract node failures/exits). Given the extreme dynamicity of the addressed deployment scenarios, the main goal is the selection of connectivity opportunities with an expectation of reasonable reliability for the served applications. The first results obtained by deploying the MMHC middleware prototype* over real test-beds demonstrate the feasibility of the approach, with limited overhead and MMHC selection/re-configuration times compatible with most applications.

* The code of the MMHC prototype is available for download, together with additional implementation insights and experimental results, at <http://lia.deis.unibo.it/research/MMHC/>

2 Deployment Scenario and Problem Statement

The MMHC scenario relevantly improves the traditional networking capabilities of wireless environments. First of all, it extends connectivity opportunities via multi-hop ad-hoc paths, thus allowing the Internet access of nodes not directly in the coverage area of infrastructure connectors. Second, it enables the exploitation of multiple paths simultaneously, e.g., to improve the overall throughput available at a client node. Third, it permits to increase connectivity availability, e.g., by enabling the rapid re-routing of traffic flows to other paths when the exploited one becomes unavailable.

To better and practically point out these advantages, let us rapidly sketch an example of a possible MMHC deployment scenario. Consider the realistic case of a group of tourists moving together and sharing pictures via Wi-Fi/Bluetooth single-hop links. Due to their limited coverage range, there could be the need for multi-hop paths to reach target friends who are currently lingering in a shop; that is enabled by collaborating tourist devices that, for instance, can transparently exploit IEEE 802.11 in ad-hoc mode to receive packets and Bluetooth to forward them along the right direction, e.g., node C in Figure 1. In addition, some tourists may be willing to periodically publish their pictures on their Web blogs even if they have no direct UMTS connectivity, e.g., they do not want to subscribe to a local UMTS provider while visiting Italy. These tourists can benefit from Bluetooth multi-hop ad-hoc connectivity toward the devices of friends with flat-rate UMTS subscription, who offer them free Internet connectivity, e.g., node A. Note that tourists' mobility may reduce the reliability of MMHC opportunities; usually there is the need to favor the selection of MMHC opportunities with compatible reliability (especially in terms of expected durability).

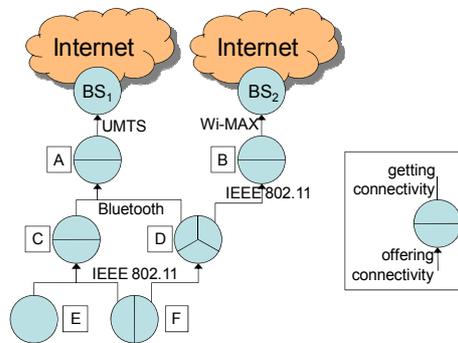


Figure 1. An example of MMHC scenario.

Similarly, when moving from city to city by train, tourists should be able to exploit MMHC opportunities offered by other passengers, possibly in other wagons, reachable via multi-hop heterogeneous paths, and connected to the Internet via Wi-Fi/WiMAX APs, such as node B. In this case the nodes tend to move together (joint mobility) and MMHC opportunities have similar expected durability. Therefore, MMHC selection should not only be mobility-aware, but also consider application-specific quality requirements, e.g., expected throughput. Moreover, if node A leaves the network, e.g., to limit its battery consumption, node D can reroute its active connections from node A to B, thus minimizing user-perceived service disruption. How-

ever, in that case, node C would have no access to the Internet anymore, since A was its only connector. It could be useful that nodes in that simple MANET self-organize themselves to provide new Internet access opportunities, for instance with node F starting to play the role of connector, thus providing C with connectivity towards BS₂.

We claim that, to support the effective self-organization of MMHC networks, there is the need of proper, effective, and concise context data describing capabilities and characteristics of available connectivity opportunities. Novel context indicators about expected node mobility, path throughput, and energy availability are needed to take proper MMHC management decisions at session startup and to promptly re-configure them at runtime, with limited overhead and impact on on-going service sessions.

3 Context Data for MMHC Management

We claim that MMHC management decisions should primarily take into account enhanced forms of context data, such as expected *node mobility* and *path throughput*, which are specific representatives, respectively, of the general properties of reliability and quality. On the one hand, given that clients and peer connectors are all mobile and may join/leave their networks abruptly, MMHC reliability is far more “fragile” than in traditional AP/BS single-hop connectivity. On the other hand, once reliability is potentially ensured as the primary goal, it is reasonable to perform MMHC management depending on coarse-grained estimated throughput. Let us note that, as better detailed in the following, it is possible to obtain these context data with reasonable accuracy by means of localized and lightweight exchange of monitoring information. In addition, MMHC management should consider the *energy availability* of the whole network. Based on coarse-grained and lightweight information about the battery of peer connectors, it is possible: i) to fairly exploit node energy capabilities; and ii) to proactively reconfigure the network when the battery level of a peer connector goes under a threshold, thus avoiding abrupt path disruptions due to battery exhaustion.

3.1 MMHC Node Mobility

We claim that mobility awareness is the most important context information needed to take proper MMHC management decisions, especially with the aim of choosing reliable connectivity opportunities based on durability expectations. Even if the literature is starting to recognize that claim, there are currently no practical, lightweight, decentralized, and client-side ways for coarse-grained estimation of node mobility. In our previous work [1], we have experimentally shown how to obtain mobility indicators by exploiting only lightweight local monitoring.

In particular, we claim that, in first approximation, single-hop connection durability depends on mutual mobility of involved nodes and coverage range of the employed wireless technology. These two simple parameters concisely summarize two main properties affecting reliability in wireless environments: user mobility, as the inclination to either stay close to or move away from nodes offering connectivity, and wireless technology characteristics, e.g., higher durability of medium-range IEEE 802.11 links if compared with short-range Bluetooth ones.

By delving into finer details, we define mutual mobility as the mobility relationship between a given participating node X and a fixed/mobile device offering connectivity to X , such as an AP or a collaborating peer connector. We introduce two indicators: i) *CMob* to measure X 's mobility with regard to a fixed AP/BS device; ii) *Joint* to evaluate X 's tendency to move together with another mobile peer (relative stillness). Both indicators have a value in the $[0, 1]$ range and are inferred via a simplified technique based on RSSI measurement at X and on RSSI variation in a recent time-frame; additional details about how to effectively obtain these indicators are in [1, 2].

For each single-hop path opportunity, we propose to quantitatively evaluate its Endurance Estimation (EE), i.e.:

$$EE = (1 - CMob) \cdot CR \quad \text{for APs/BSs} \quad (1)$$

$$EE = Joint \cdot CR \quad \text{for mobile peers} \quad (2)$$

where Coverage Range (CR) is in $[0, 1]$ and, in first approximation, only depends on the exploited wireless technology.

While EE provides single-hop context information about expected durability, obtained locally without any access to distributed monitoring data, we introduce Path Mobility (PM) for coarse-grained evaluation of multi-hop path durability:

- PM is equal to EE in the case of a single-hop path;
- the PM of a k -hop path is equal to the EE of the k^{th} hop multiplied by the PM of the remaining sub-path starting from the $(k-1)^{\text{th}}$ node.

Let us observe that PM quickly degrades while increasing the number of path hops, to model the desired effect of strongly favoring the selection of short durable paths. In fact, the MMHC goal is not of supporting the complex realization of any kind of MANET, but only to enable short reliable ad-hoc paths towards infrastructure connectors, even by abruptly filtering out connectivity opportunities that are estimated too unreliable because of excessive mobility.

3.2 MMHC Path Throughput

Similarly to context data about mobility for coarse-grained estimations of connector reliability (to infer MMHC opportunity durability), we have worked to properly model the expected throughput of potentially available multi-hop heterogeneous paths depending on lightweight monitoring data. In particular, based on our large campaign of measurements on heterogeneous wireless networks, we have observed that three elements are crucial, in first approximation, for throughput: i) the wireless technology of each single-hop sub-path, ii) the number of hops in the path, and iii) the number of clients/peer connectors simultaneously served by each connector in the path. Other factors, which have partial influence on the overall path performance, are not so relevant for a coarse-grained throughput estimation. For instance, about iii), we have experimentally verified that in the challenging case of simultaneous transmit/receive operations by all clients over the same single-hop link up to throughput saturation, competing devices tend to fairly share the total bandwidth. We adopt the conservative simplifying assumption that in any case a node can achieve a maximum throughput inversely proportional to the number of active nodes on that single-hop (see Figure 2). Given the above considerations, we propose a simplified lightweight model to evaluate Estimated Throughput (ET):

$$ET = NB \quad \text{for APs/BSs} \quad (3)$$

$$ET = (1 - HD) \cdot MT / \#clients \quad \text{for mobile peers} \quad (4)$$

where Nominal Bandwidth (NB) depends on the exploited wireless technology, Hop Degradation (HD) models per-hop throughput degradation (experimentally measured and set to 20% in first approximation), which is almost independent of the number of local clients, and Maximum Throughput (MT) is the expected maximum throughput toward the wired Internet, i.e., $\min \{ET \text{ of previous single-hop sub-path, NB of the considered single-hop sub-path}\}$. Note that the number of clients is not considered in the case of direct connections to APs/BSs, also given the practical impossibility to portably obtain this information when working with currently deployed AP/BS network equipment. Let us finally stress again that this procedure for ET estimation is only a rough calculation of actual runtime throughput, but is very simple and lightweight, thus enabling scarcely intrusive comparison of multi-hop paths.

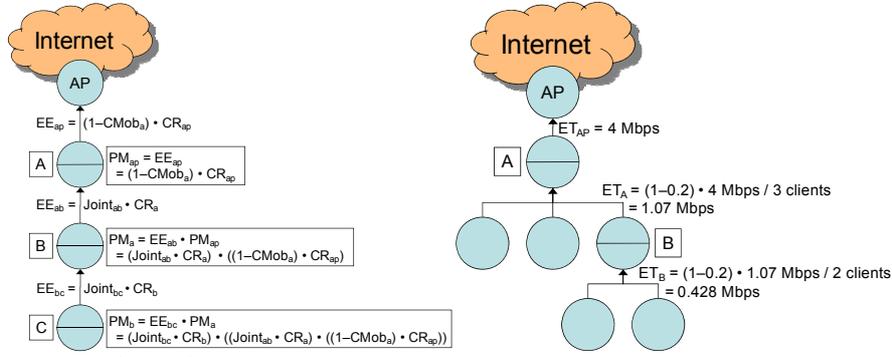


Figure 2. Our coarse-grained PM (left) and ET (right) estimation.

3.3 MMHC Energy Availability

While PM and ET are useful to provide estimations about mobility-related durability and throughput, they do not provide any information about expected path durability due to energy consumption. Analogously to what presented before, MMHC adopts a simplifying approach for coarse-grained and lightweight energy considerations. The primary ideas are of simply avoiding the paths composed by nodes with low battery levels and of not overloading a small set of connectors with a large amount of traversing traffic to avoid to quickly consume their batteries due to traffic routing. The goal is twofold: i) preserving the battery level of each node, by focusing on those nodes whose battery level is running out, and ii) trying to increase path durability. Let us rapidly point out that the MMHC approach does not replace but is additional to other more sophisticated and effective techniques for power consumption reduction, e.g., IEEE 802.11 awake/doze periodic state switch or Bluetooth Sniff/Park states.

By going into finer details, MMHC distributes context information related to Node Battery Level (NBL) and thus permits to take informed decision sufficiently in advance for reconfiguring the network prior to path disruption. In particular, we define the Average Path Energy (APE) indicator of the k^{th} hop of the path as:

$$APE_k = \frac{(APE_{k-1}) \cdot (K - 1) + NBL_k}{k} \quad (5)$$

i.e., the average battery level of nodes in the path to the Internet. In addition, we define the Residual Path Energy (RPE) indicator as:

$$NBL_1 \quad \text{for the 1}^{st} \text{ hop of the path} \quad (6)$$

$$NBL_k \cdot RPE_{k-1} \quad \text{for the } k^{th} \text{ hop of the path} \quad (7)$$

Note that APE and RPE convey different context information. The former gives a fairness estimation about the distribution of power consumption, useful to quantitatively compare available paths. For instance, given two paths with good ET values, MMHC can choose to exploit the one with greater APE to optimize peer connector power consumption. The latter alerts about the possibility that a given path becomes unavailable in a short time, e.g., since one of the connectors is running out of energy. Again, the RPE indicator is built to favor the exploitation of short paths. Considering the example in Figure 3, based on APE, node F should prefer the BS₁-A-C path (APE=0.51) instead of BS₂-B-D (APE=0.45). However, the first path has a considerably lower RPE than the second (respectively 0.0665 and 0.20), correctly modeling the fact that a node of the first path, i.e., node C, is exhausting its battery.

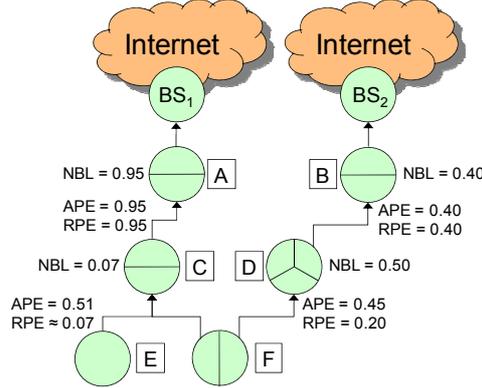


Figure 3. APE and RPE estimation for two different paths.

4 Local and Global Management for Reliable Paths in MMHC

We envision the self-organization of MMHC networks as a two-phase procedure: a local phase where nodes aim to quickly achieve a form of Internet connectivity at session startup and a global phase where nodes coordinate themselves to incrementally improve their network exploitation in terms of availability and quality.

The local phase performs connector evaluation rapidly and efficiently to ensure prompt but sub-optimal response; it is based on context data that is either locally available (EE) or gathered at single-hop connection establishment time (PM and ET), thus providing coarse-grained estimation of path reliability and quality. In particular, the local phase is reactively activated only when an active single-hop connection fails, e.g., because one in-use connector becomes unavailable. In this phase, nodes:

- 1) gather RSSI sequences of their visible peer connectors to compute CMob/Joint;
- 2) perform a single-hop connection with the most reliable connector from the point of view of mobility, i.e., with greatest EE;
- 3) estimate PM and ET of the whole path, by gathering and exploiting PM and ET of previous hops in case of peer connectivity.

Nodes connected to multiple connectors exploit PM and ET values to estimate which is the most suitable path. Due to the volatility of MMHC networks, the main purpose of these evaluations is to ensure path durability, while throughput is considered only as a secondary objective. In fact, MMHC allows users to specify the Required Reliability (RR) for each of their applications (RR ranges in the [0,1] interval, with 1 for maximally privileging reliability at the expense of throughput). By delving into finer details, MMHC nodes:

- 1) as a first try, select the path with greatest ET among the only paths with $PM \geq 80\% RR$. If at least one compliant path is found, the algorithm stops;
- 2) otherwise, they also examine paths with $PM \geq 50\% RR$. If at least one compliant path is found, the algorithm stops;
- 3) otherwise, they take into account any potentially available path, with no more limitations on the space of connectivity opportunities.

Let us point out that the local management phase leads to the establishment of a tree-network topology: connections can only follow bottom-up paths because they are built up from clients towards the Internet access points. For instance, in Figure 1 clients can achieve Internet connectivity by establishing 1-to-many tree-like connector/client relationships; clients connected to multiple connectors can access multiple tree-networks simultaneously; instead, connectors cannot exploit connectivity offered by their clients at the same time.

The global phase is in charge of enhancing the connectivity paths established in the local phase, by ensuring long-term availability. It exploits a wider set of context data and connectivity opportunities. On the one hand, APE and RPE data are spread to proactively modify network topology to avoid nodes with scarce battery. On the other hand, the already established connectivity allows clients with simultaneous connection to multiple connectors to periodically notify their single-hop connectors that they can potentially work as bridges among different tree-networks. In this phase, nodes:

- 1) periodically collect up-to-date context data about PM, ET, APE, and RPE of available paths from peer connectors/clients;
- 2) change routing rules when the currently exploited path becomes unavailable or its RPE value goes below 0.1;
- 3) select new paths, as the local phase does, by privileging paths with APE in the [0.5, 1.0] range (preferred exploitation of nodes with high battery resources):
 - a. if the new path exploits a new connector, the involved nodes simply have to change their local routing rules;
 - b. if the new path uses a client connected to other tree-networks, a role-switch procedure is triggered (see below).

This metric is conservative, by proactively triggering a network reconfiguration only based on APE. However, it is easy to change MMHC behavior to adopt more aggressive approaches that take into account also ET and other parameters, at the cost of additional monitoring overhead.

It is worth noting that the local phase is rather static, letting nodes establish new

connections only when already available ones disappear. Instead, the global phase provides dynamic network management not only by changing the exploited connector via routing rule updates, but also by switching the role between connectors and clients (role-switch procedure). In fact, role-switch relevantly improves topology dynamicity and widens networking opportunities: for instance, a connector can select, as next-hop to the Internet, one of its current clients such as node F in Figure 4. Then, MMHC starts its role-switch procedure as follows:

- 1) the connector notifies its client that there is the need for role-switch;
- 2) the client enables forwarding capabilities and update routing rules;
- 3) the connector starts forwarding packets to the selected client.

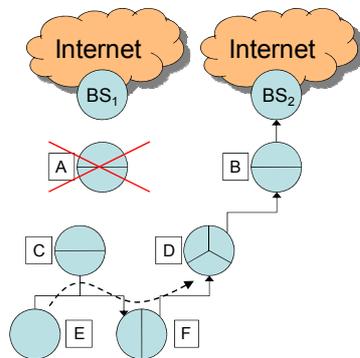


Figure 4. Nodes C/F role-switch procedure after node A failure.

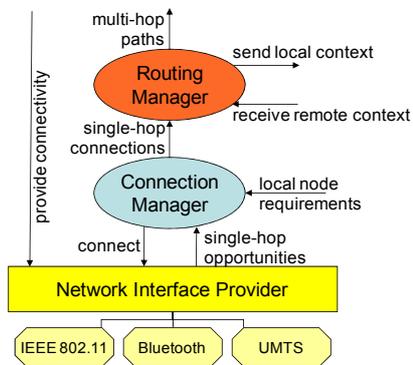


Figure 5. MMHC middleware architecture.

The role-switch procedure affects only the pair of nodes directly involved in it (localized management operation). After the switch, the original connector is still the node contributing to the existence of the physical network, e.g., working as Bluetooth master and DHCP server; the novel element is that the old client starts playing the role of gateway. In this way, the role-switch procedure imposes limited overhead, e.g., not requiring the time consuming establishment creation of new single-hop links (see Section 5) and permits the decoupling of the roles of connection establisher and gateway. In addition, other possible clients of a connector are not affected by role-switch: they keep on sending packets to their old connector, thus possibly delving into sub-optimal node configurations but limiting reconfiguration actions to minimize manager overhead. Figure 4 shows how C and F reconfigure their network after A failure; note that E continues to exploit C as peer connector, which forwards packets to F.

5 Architecture and Implementation Insights

Figure 5 gives a high-level overview of our middleware architecture, which is layered to properly separate connection/routing level local/global management operations and to limit the unnecessary visibility of implementation details, thus increasing usability. *Network Interface Provider* (NIP) provides homogeneous access to heterogeneous interfaces and local context sources; it provides a common API by hiding low-level

peculiarities of underlying drivers and operating systems. *Connection Manager* performs single-hop connections; it gathers RSSI sequences to evaluate CMob/Joint and EE for any single-hop MMHC opportunity; on this basis, it takes local decisions on the subset of single-hop paths to activate. *Routing Manager* works to perform multi-hop paths; it manages routing rules and triggers role-switch procedures when needed.

For the sake of brevity, to give a practical idea of some MMHC implementation issues, here we focus on how (lower layer) MMHC achieves portability over different platforms. Additional details are available on the MMHC Web site. The current MMHC prototype supports IEEE 802.11 and Bluetooth interfaces, by including wrappers for both Linux and Windows XP/Vista. Wi-Fi interfaces are accessed via Linux Wireless Extensions on Linux client nodes and via the Microsoft Network Driver Interface Specification User-mode I/O (NDISUIO) on Windows XP/Vista (NDISUIO is platform-dependent but portable among different wireless interface implementations). For instance, NIP exploits the NDISUIO function `DeviceIOControl()` to query the `OID_802_11_BSSID_LIST_SCAN` object to retrieve the complete list of currently reachable connectors, either IEEE 802.11 APs or peer nodes in ad-hoc configuration. Bluetooth interfaces are accessed via the standard API provided by the BlueZ protocol stack on Linux client nodes, while via the API provided by the Windows Driver Kit and the Software Development Kit on Windows XP/Vista. For example, NIP achieves visibility of the set of Bluetooth devices in proximity by invoking `BluetoothFindFirstDevice` and `BluetoothFindNextDevice` functions.

In addition, NIP can gather battery-related context information on both Linux and Windows XP. In the former case, it exploits `status` and `info` files in the `/proc/acpi/battery/BAT0` directory; it estimates the NBL parameter comparing the remaining capacity and the last full capacity values. In the latter case, the System Event Notification Service (SENS) `BatteryLevel` property is exploited to access the battery status and directly get the NBL parameter.

We have worked and are working on the extensive experimental validation of the MMHC prototype. Due to space limitations, here we rapidly present some performance measurements about MMHC overhead, by referring to the MMHC Web site for additional experimental results. MMHC has demonstrated to add a limited overhead, negligible if compared with the long delays imposed by several wireless technologies to handle handovers and establish new connections, e.g., the Bluetooth inquiry [3]. In particular, we have tested Connection and Routing Manager performance when creating new single-hop connections and managing routing rules for multi-hop paths. In the case of a new Wi-Fi/Bluetooth connector joining the managed network, Connection Manager spends 3.102/17.916s to configure the new single-hop connection, e.g., due to 3.041/14.370s to discover the connector (almost all deriving from long Wi-Fi/Bluetooth standard operations), only 0.039/0.116s to evaluate the connector suitability (under MMHC responsibility), and 0.022/3.430s to connect to it via association/PAN connection. Routing Manager is much faster, requiring only 273ms on average to establish a new path: 60ms to select the best path and consequently update routing rules, the remaining time to distribute context data.

The main performance differences between the two interface types have been exhibited for connector discovery and connection establishment: the longer IEEE 802.11 discovery phase is mainly due to the time for setting up the ad-hoc mode, which is of infrequent usage and not optimized in several Wi-Fi cards; Bluetooth inquiries and

PAN connections are slower than IEEE 802.11 scans and associations [3]. In addition to interface types, the reported indicators have demonstrated to significantly depend on card model and driver implementations. For instance, Orinoco Gold interfaces have exhibited larger IEEE 802.11 ad-hoc throughput than PROWireless cards (about 6 times) because the latter only support ad hoc transmission at 1MB/s. Similarly, MMHC can halve the Bluetooth inquiry period over MS operating systems at the expense of risking not to sense a small fraction of connectors, as proposed first in [13]; that optimization is impossible with Linux-based BlueZ drivers.

Let us stress that the greater delay for network setup than for network reconfiguration justifies the MMHC approach, with a local management phase reactively activated only when a single-hop connection is lost and a global phase that periodically updates routing rules to optimize performance once the network is working (to shorten the long and expensive startup phase of first connection establishment).

6 Related Work

Several proposals have recently investigated some specific partial aspects of the MMHC scenario. For instance, [4] points out the primary technical aspects of WLAN-based multi-hop networks, while [5] aims to extend cellular network capabilities via relay stations, with the main goal of increasing cellular coverage. [6] and [7], instead, specifically address the issue of managing client mobility among heterogeneous multi-hop networks. Other proposals focus on the effective allocation of the shared wireless medium frequencies and the scheduling of time slots to minimize interferences and packet collisions [8, 9]. These contributions were crucial for the full understanding of both the theory and the main characteristics of multi-hop networks. However, they did not concentrate on realistic, feasible, and practical solutions to guide the design and implementation of prototypes for seamless and mobility-aware MMHC. Also [10] and [11] provide some relevant contribution by identifying major drawbacks and weaknesses of theoretical work in the literature; however, they do not propose practical solutions to address these weaknesses.

In a wider perspective, it is possible to note that most work in the literature proposes elegant but complex models for MMHC, without considering concise context indicators to simplify MMHC management (reduced overhead at the expense of limited distance from decision optimality). In particular, only recent contributions start to recognize the importance of providing lightweight mechanisms to maximize reliability. To the best of our knowledge, [12] is the only notable proposal that practically addresses the issue of improving network reliability by spreading context data about path robustness; however, it does not estimate availability based on mobility/energy considerations and does not consider path quality as our MMHC prototype does.

7 Conclusions

Recent research activities are starting to recognize the suitability of novel middlewares to leverage the adoption of MMHC scenarios, thus fully exploiting the fre-

quent, ubiquitous, and heterogeneous networking opportunities available nowadays. Our research work points out how innovative context data are crucial to inform management solutions that effectively answer to the reliability, throughput, and availability requirements of running applications. In particular, our MMHC prototype demonstrates the feasibility of our approach, with prompt sub-optimal connectivity decisions and limited costs, thanks to the proper adoption of a reactive local management phase for connectivity establishment at session startup and a proactive global management phase for connection re-configuration.

The promising results already achieved are stimulating our further work. In particular, we are investigating effective models to dynamically evaluate and evolve the trust degree that clients, in a completely decentralized way, associate to their peer connectors, in order to affect connectivity offerings via incentives. In addition, we are extending the MMHC prototype to transparently handle also the splitting of the traffic flow of a single application at a client along different multi-hop heterogeneous paths.

References

1. P. Bellavista, A. Corradi, C. Giannelli, "A Layered Infrastructure for Mobility-Aware Best Connectivity in the Heterogeneous Wireless Internet", First Int. Conf. MOBILE Wireless MiddleWARE, Operating Systems, and Applications (Mobilware), Austria, Feb. 2008.
2. P. Bellavista, A. Corradi, C. Giannelli, "Evaluating Filtering Strategies for Decentralized Handover Prediction in the Wireless Internet", 11th IEEE Symp. Computers and Communications (ISCC), Italy, June 2006.
3. E. Ferro, F. Potorti, "Bluetooth and Wi-Fi Wireless Protocols: a Survey and a Comparison", IEEE Wireless Communications, vol.12, no.1, pp.12-26, Feb. 2005.
4. S.M. Faccin, C. Wijting, J. Kenck, A. Damle, "Mesh WLAN Networks: Concept and System Design", IEEE Wireless Communications, vol.13, no.2, pp. 10-17, Apr. 2006.
5. L. Le, E. Hossain, "Multihop Cellular Networks: Potential Gains, Research Challenges, and a Resource Allocation Framework", IEEE Communications Magazine, vol.45, no.9, pp.66-73, Sept. 2007.
6. S. Pack, X. Shen, J.W. Mark, J. Pan, "Mobility Management in Mobile Hotspots with Heterogeneous Multihop Wireless Links", IEEE Communications Magazine, vol.45, no.9, pp.106-112, Sept. 2007.
7. P.P. Lam, S.C. Liew, "Nested Network Mobility on the Multihop Cellular Network", IEEE Communications Magazine, vol.45, no.9, pp.100-104, Sept. 2007.
8. L. Badia, A. Erta, L. Lenzini, M. Zorzi, "A General Interference-Aware Framework for Joint Routing and Link Scheduling in Wireless Mesh Networks", IEEE Network, vol.22, no.1, pp.32-38, Jan.-Feb. 2008.
9. Ho Ting Cheng; W. Zhuang, "Joint Power-Frequency-Time Resource Allocation in Clustered Wireless Mesh Networks", IEEE Network, vol.22, no.1, pp.45-51, Jan.-Feb. 2008.
10. M. Conti, S. Giordano, "Multihop Ad Hoc Networking: the Theory", IEEE Communications Magazine, vol.45, no.4, pp.78-86, Apr. 2007.
11. M. Conti, S. Giordano, "Multihop Ad Hoc Networking: the Reality", IEEE Communications Magazine, vol.45, no.4, pp.88-95, Apr. 2007.
12. R. Baumann, S. Heimlicher, B. Plattner, "Routing in Large-Scale Wireless Mesh Networks Using Temperature Fields", IEEE Network, vol.22, no.1, pp.25-31, Jan.-Feb. 2008.
13. B.S. Peterson, R.O. Baldwin, J.P. Kharoufeh, "Bluetooth Inquiry Time Characterization and Selection", IEEE Trans. on Mobile Computing, vol.5, no.9, pp.1173 - 1187, Sep. 2006.