Internet Connectivity Sharing in Multi-path Spontaneous Networks: Comparing and Integrating Network- and Application-Layer Approaches

Paolo Bellavista and Carlo Giannelli

Dip. Elettronica Informatica e Sistemistica, University of Bologna
Viale Risorgimento 2, 40136 Bologna, Italy
{paolo.bellavista, carlo.giannelli}@unibo.it

Abstract. Spontaneous networking, where wireless mobile nodes opportunistically exploit multi-hop ad-hoc paths toward peers to share content and available resources in an impromptu way, has recently received growing interest from both industry and academia. In this paper, we specifically focus on the notable case of sharing connectivity to the traditional Internet, with the general goal of an overall better exploitation of connectivity resources, often underutilized as the population of wireless devices grows, as well as their local computing/memory/bandwidth resources. In particular, here we show how our novel middleware, called RAMP, can exploit both network- and application-layer solutions to dynamically manage mission-oriented paths toward peers offering Internet connectivity. Thanks to our middleware-level cross-layer approach, RAMP can dynamically select and combine different solutions for multi-hop multi-path ad-hoc path formation and can take proper management decisions based on run-time context. The reported results demonstrate the suitability of dynamically integrating network- and application-layer approaches to achieve the best overhead/performance tradeoff depending on specific application requirements.

Keywords: Internet Connectivity, Spontaneous and Collaborative Networks, Middleware, Heterogeneous Wireless Networks, Multi-hop Multi-path Connectivity.

1 Introduction

In the last couple of years spontaneous networking has received growing and growing attention for its promising aspects of better exploitation of available wireless connectivity, resource connectivity sharing, and immediate connectivity offer in regions with difficult coverage [1, 2]. Notwithstanding first interesting research results have been achieved [3-6], several technical challenges are still open, such as the concurrent and effective exploitation of heterogeneous wireless technologies (multi-hop paths made up by multiple heterogeneous links), of multiple wireless technologies/cards at the same node (different heterogeneous paths traversing a single node), and of the combination of single-hop infrastructure-based links and ad-hoc ones.

Anyway, it starts to be widely recognized not only the relevant potential of spontaneous networks for better exploitation of resources in collaborative smart environments of the future, but also that the complexity of spontaneous network management

makes it inadequate to handle it directly in the supported collaborative applications. We claim the need for novel middleware capable of simplifying the development of applications on top of spontaneous networks, by properly and effectively managing the complexity associated with multi-hop multi-path heterogeneous connectivity, with no need of complete, global, and strictly updated knowledge about the dynamic topology and characteristics of the exploited paths. To this purpose, we have developed an innovative middleware, called Real Ad-hoc Multi-hop Peer-to-peer (RAMP), which transparently manages the technical challenges related to i) global decisions based on limited local visibility, ii) erratic behavior of mobile peers sharing resources in an impromptu way, and iii) IP addressing in spontaneous networks [6].

In this paper, we specifically focus on a notable case of collaborative resource sharing, i.e., sharing bandwidth and connectivity toward the traditional Internet. The rationale is that many portable devices are nowadays equipped with multiple wireless interfaces and flat-rate/large-bandwidth subscription for Internet connectivity: their potentially available bandwidth is often underutilized, while it could be shared with other peers in current vicinity, thus better exploiting the growing availability of computing/memory/bandwidth resources at portable wireless terminals. In particular, this paper shows how RAMP can exploit both network- and application-layer approaches to dynamically handle mission-oriented, multi-hop, heterogeneous, and sporadic paths toward peers that offer a portion of their connectivity bandwidth to the Internet. We claim that the creation and management of these spontaneous intermittent paths at the network layer (L3 approach) can achieve good performance and limited overhead in the case of relatively stable and short paths, but at the expense of minor flexibility. Instead, application-layer solutions (L7 approach) can achieve relevantly better flexibility, e.g., by enabling the exploitation of different multi-hop heterogeneous paths traversing the same node, at the expense of a relatively greater overhead. Our solution guideline is that in spontaneous networks it is suitable to take path management decisions (either L3 or L7 or a combination of them) only at runtime and based on currently applicable context.

According to this principle, we have extended the RAMP prototype to support Internet connectivity sharing in spontaneous networks. In particular, three middleware components have been added: *InternetClient*, active on nodes requesting Internet connectivity to their peers, *InternetService*, running at Border Nodes (BNs, i.e., nodes directly connected to the traditional Internet and offering part of their underutilized connectivity), and *Layer3Manager*, active on peers that allow RAMP to modify local routing rules working at the operating system level. The RAMP extension originally presented in this paper exploits L3 and L7 approaches to support three different modes for Internet connectivity sharing in spontaneous networks: i) a low-overhead L3 Single-Path (L3SP) solution, ii) a highly flexible L7 Multi-Path (L7MP) one, and iii) a hybrid L3L7-Combo Multi Path (L3L7CMP) one. Potentially available paths are created and selected based on runtime context by comparing end-to-end path performance, estimated dynamically in a very lightweight way. In addition, RAMP enables even the same application instance to exploit different approaches/paths simultaneously (for different connection requests); in other words, approach/path management is performed dynamically with per-connection granularity.

The RAMP prototype is available for download as a useful tool for the community of researchers in the field and can be easily deployed over real environments with

standard wireless cards and execution platforms. The reported results demonstrate the suitability of dynamically integrating network- and application-layer approaches to achieve the proper overhead/performance tradeoff at runtime. In particular, RAMP has demonstrated to be able to effectively exploit dynamically available BNs depending on their provided bandwidth, estimated in a very lightweight way at service provisioning time. Moreover, the additional overhead imposed by L7 has demonstrated to be limited, largely counterbalanced by increased connectivity reliability and throughput thanks to the simultaneous exploitation of multiple paths.

The rest of the paper is organized as follows. Section 2 summarizes the pros and cons associated with network/application-layer approaches to Internet connectivity sharing in spontaneous networks, while Section 3 details the different modes that RAMP enables. Section 4 goes into the technical details of the RAMP architecture and of some notable implementation insights. Experimental results demonstrating the suitability of combining network/application-layer approaches in RAMP are in Section 5, while related work, conclusive remarks and on-going research end the paper.

2 Internet Connectivity Sharing in Multi-hop Multi-Path Spontaneous Networks

To better point out the challenging environments targeted by RAMP and to highlight the differences between network- and application-layer Internet connectivity sharing, let us rapidly sketch a practical example of multi-path spontaneous network. Consider the realistic case of a group of students in a lecture hall carrying on mobile clients equipped with multiple heterogeneous interfaces (see Figure 1), e.g., laptops with IEEE 802.11 and Bluetooth, cell phones with UMTS and Bluetooth, and smart phones with UMTS, IEEE 802.11, and Bluetooth. Some of the nodes (the BNs) get direct connectivity to the Internet, by taking advantage of their flat-rate UMTS subscription or by connecting to a free-of-charge IEEE 802.11 access point of the university campus, e.g., NodeA and NodeD. BNs can share connectivity via subgroups created in an impromptu way, by exploiting their local wireless interfaces to get and offer single-hop connectivity, even participating to multiple subnets simultaneously.

The wide variety of exploited interfaces strongly pushes for the adoption of standard IP as the common layer (as a useful and practical simplifying assumption). In addition, this choice enables solutions that can be practically deployed over already existing networks, thus promoting easy deployability and potentially rapid market penetration. In such a scenario, the sharing of Internet connectivity can be realized either through more traditional L3 path formation solutions or by supporting inter-node packet dispatching at the application layer (L7 store/carry/forward techniques).

In general, as a preliminary and introductory overview, L3 solutions aim at configuring peer nodes with proper settings for their default gateways, by using the traditional mechanisms designed and implemented for the wired Internet, to create multi-hop paths toward BNs. Once an L3 multi-hop path has been configured, nodes residing on that path can get Internet connectivity. For instance, in Figure 1, once NodeG, NodeF, and NodeB have set their default gateway, NodeG can use the NodeF-NodeB-

NodeA L3 path. Note that each single node can exploit only one L3 path, even if alternative multiple paths are potentially available, e.g., NodeF-NodeE-NodeD.

On the other hand, L7 approaches can enable Internet connectivity by dispatching packets among cooperative nodes at a higher layer of solution, without interacting with operating system-level routing rules. In this case, with packet routing performed at the application layer, packet delivery does not depend on default gateway configuration and each node can use the L7 multi-hop path currently deemed as the most suitable, e.g., because it provides largest bandwidth (less loaded path) or requires lowest power consumption (local exploitation of Bluetooth interface). For instance, NodeF can access the Internet via the NodeB-NodeA path, while nodeG via the NodeF-NodeE-NodeD one. In addition, the same node can exploit different L7 paths for different connections simultaneously, e.g., in order to maximize the total throughput. For instance, NodeF can exploit the NodeB-NodeA path to download a given Web page and the NodeE-NodeD one to connect to an FTP server. In other words, sharing Internet connectivity at L7 enables the simultaneous adoption of different overlay networks, e.g., different nodes in the same subnet may access the Internet in a different way. Furthermore, the same node can exploit different overlay networks at the same time. Moreover, the adoption of an L7 solution does not prevent from the capability of exploiting facilities available at L3: it is possible to exploit both L3 and L7 solutions simultaneously, e.g., the same NodeF accesses the Internet via NodeA by exploiting an L3 path and via NodeD by adopting an L7 path.

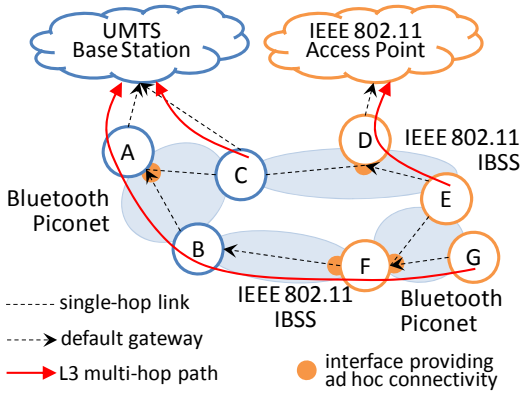


Figure 1. Multi-path spontaneous network scenario.

Considered the above scenario, we claim the suitability of adopting the following solution guidelines to share Internet connectivity in multi-hop multi-path heterogeneous spontaneous networks:

1) **supporting both L3 and L7 approaches**. Since they provide different technical pros and cons, as better detailed in the following section, a middleware support should enable both and dynamically select the most suitable one depending on runtime context (characteristics of the deployment environment, application requirements, most suitable overhead/performance tradeoff, …). In general, L3 approaches tend to impose little routing overhead after the first initialization phase, but require careful management because path formation at a node may impact on

the possible selections at other nodes. L7 approaches, instead, generally impose higher overhead, but enable multi-path Internet access;

2) **context-aware estimation of available paths**. The availability of multiple BNs can improve performance, but calls for suitable metrics to dynamically evaluate which is the most suitable path to be enabled for a given client node and for a given application connection. Metrics should provide quantitative estimations of path quality, at the same time with minimum impact on overhead, e.g., by avoiding frequent dissemination of monitoring information between nodes;
3) **differentiated metrics at session initialization and at service provisioning time.** The path evaluation process should be different at service initialization and provisioning time. In the former case it is appropriate to exploit rather static context data, easily retrievable before actual connections are established, e.g., estimated bandwidth based on path hops number, thus providing a coarse-grained but lightweight estimation of available paths. In the latter case it is adequate to adopt finer-grained metrics, by exploiting the visibility at zero cost of the actual path performance that nodes are currently experiencing at runtime.

3 RAMP for Internet Connectivity Sharing

According to the above solution guidelines, we have extended our RAMP middleware to support Internet connectivity in spontaneous networks exploiting both L3 paths, managed by exploiting routing configuration mechanisms and tools at the operating system level, and L7 paths, managed by exploiting application-layer middleware components to dispatch packets to collaborative nodes. As already stated, three middleware components have been added: *InternetClient*, active on client nodes requesting Internet connectivity, *InternetService*, running at BNs, and *Layer3Manager*, active on peers participating to L3 paths. To take advantage of proper dynamic selection based on currently applicable context, RAMP enables three different modes for Internet connectivity sharing: L3SP, L7MP, and L3L7CMP, as detailed in the following.

3.1 Multiple and Combined Layer Modes in RAMP

L3SP is based on the dynamic configuration of standard routing rules at the operating system level on intermediate nodes in order to create the needed L3 path from the client to a suitable BN currently offering Internet connectivity. In this case, the RAMP middleware transparently works to create the L3 path by modifying the default gateway configuration on any node along the path. For instance, in Figure 2, to access the Internet via BN_1, NodeC, NodeY, and NodeX must specify respectively NodeY, NodeX, and BN_1 as their default gateway. To this purpose, any node along the path has to collaborate to packet forwarding and to offer the possibility of modifying its local routing rules dynamically. By delving into finer details, the RAMP client exploits the InternetClient component to send an L3 path configuration request to the nodes along the path (dynamically determined in an innovative and effective way by RAMP [6]), while Layer3Manager components on any intermediate node modify

local routing rules. Once every node has enforced the required routing rule modifications, any application at the client node can access the Internet directly via the L3 multi-hop path towards BN, with no additional need for runtime support by InternetClient or Layer3Manager, thus imposing minimum overhead. However, all applications at a node can exploit only one path to the Internet, even in the case of multiple BN availability. Let us note that, due to the relatively high costs of path formation via routing configuration, L3 paths should be created only when necessary (reactive approach in response to client connectivity requests, no proactivity) and by adopting a lightweight coordinated view among neighbors to avoid clashing routing requests.

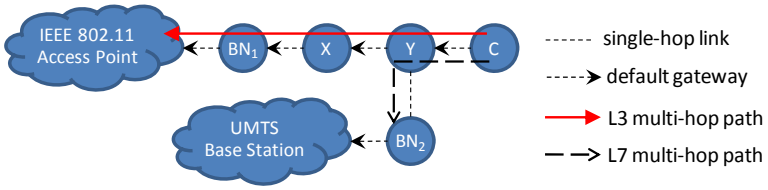


Figure 2. L3 and L7 paths.

L7MP is based on the exploitation of RAMP-supported L7 paths, with no need of any modification of underlying routing rules because packet forwarding is performed hop-by-hop at the application level. As a consequence, applications can exploit multiple BNs even simultaneously (additional implementation details about the RAMP support for L7 paths are in the following section). The L7MP mode adopts a double-proxy architecture: an InternetClient proxy on any node requesting Internet connectivity and an InternetService proxy at any BNs sharing Internet connectivity. InternetClients are in charge of receiving local application requests and of dispatching them to one of the previously discovered InternetServices. The selection of the most suitable InternetService is performed with per-connection granularity, based on currently applicable context (see the following). The InternetService receiving the client request performs the actual connection with the Internet end-point, waits for a response, and finally forwards the response to the origin InternetClient. Finally, InternetClient transparently forwards the response to the local application client. For instance, in the case of HTTP applications, InternetClient acts as a proxy receiving/sending HTTP requests/responses from/to the local Web browser, while InternetService contacts the remote Web server, performing the actual HTTP interaction on behalf of the browser.

Let us note that L7MP allows the simultaneous exploitation of multiple paths and multiple BNs by the same client node. On the one hand, this permits to potentially increase the overall achievable throughput, which is particularly important in the case of scenarios with single-hop links with limited bandwidth (quite common in spontaneous networking). On the other hand, this helps in achieving greater reliability because, in the case of disruption of a single path, on-going connections may benefit from being rapidly switched to other available paths. However, L7MP tends to impose an additional overhead due to application-layer routing, e.g., for packet data encapsulation into application-level RAMP packets.

L3L7CMP combines the exploitation of both L3 and L7 approaches. In this case, InternetClient is aware of the (possible) availability of one L3 path and multiple L7 paths; based on dynamically gathered context, it dynamically selects which is the

most suitable choice for the specific connection request. When choosing an L7 path, InternetClient uses the double-proxy architecture presented for L7MP; instead, in the case of L3 path, it adopts a single-proxy approach and directly contacts the requested Internet end-point via the selected BN.

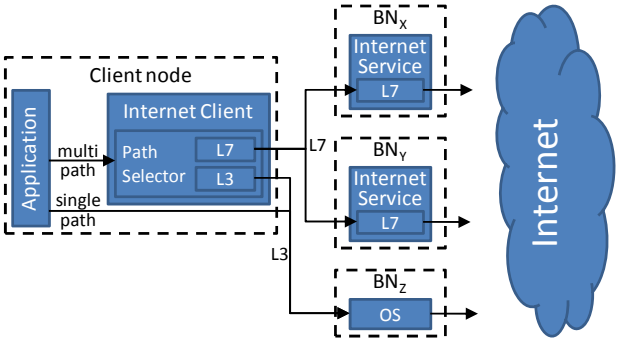


Figure 3. Internet requests adopting proposed approaches.

We have decided to support the different L3SP, L7MP, and L3L7CMP modes in RAMP because we claim that the most appropriate solution could be chosen only at runtime depending on application requirements and applicable context. There is not a single mode always preferable to the others: in fact, the three modes permit to achieve different tradeoffs in terms of path (re-)configuration costs, multi-path support, and communication capabilities/overhead, as better detailed in the following.

On the one hand, L3SP may be expensive in terms of management overhead because it requires i) routing rule modifications on intermediate nodes and ii) solving possible conflicts between routing requirements of different nodes. For instance, in Figure 2, if the previous default gateway of NodeY was BN_2, NodeC request could disrupt the active Internet connection due to default gateway change. In addition, an L3 path can be modified depending on successive requests of nodes partially sharing some peers: if, after a while, NodeY selects BN_2 as default gateway again, NodeC path to the Internet is modified even if NodeC has not asked for any change. Moreover, if nodes are highly mobile, L3 paths are ineffective because intermediary nodes can abruptly leave the network rather frequently, thus requiring repeated L3 configuration processes. Instead, L7MP does not require any path pre-configuration because the path to be exploited is specified anytime InternetClient sends a packet to an InternetService at a BN: packets encapsulating application requests are managed exactly as any RAMP packet. Moreover, RAMP supports advanced store&forward routing, permitting to correctly dispatch packets even in case of intermittent connectivity (additional details on the RAMP Web site [6]).

On the other hand, once a L3 path has been correctly configured, L3SP imposes minimum overhead at service provisioning time because it exploits operating system-level forwarding. Any Internet connection automatically exploits the L3 multi-hop path, despite the adopted application-layer protocol and with no need of any further InternetService/Client intervention; in other words, RAMP clients can access the Internet as if they were BNs. On the opposite, L7MP suffers from the additional overhead imposed by the exploitation of the double-proxy architecture and by the request/response encapsulation into RAMP packets. L3L7CMP can partially reduce

overhead, since it can sometimes adopt a single-proxy approach avoiding data encapsulation; however, InternetClient currently supports only HTTP (and Pseudo-HTTP), as better detailed in the following; therefore, for instance, L7MP and L3L7CMP cannot provide access to an RTP-based stream server.

In short, RAMP users can exploit the desired mode and switch among modes dynamically. The L3SP mode well suits the case of relatively stable topologies and is the only one to use to enable application-level protocols not currently supported by InternetClient/Service. L7MP is more suitable for highly dynamic scenarios where there is the need to support increased connectivity reliability (multi-path plus store&forward RAMP features). L3L7CMP is a compromise between the two since it couples both L3 and L7 approaches, with slightly lower overhead than L7MP, but limited store&forward capabilities.

3.2 *PathLength* and *PathThroughput* Metrics

RAMP adopts the *PathLength* and *PathThroughput* metrics to evaluate which is the most suitable BN to create an L3 path to and, in the case of multi-path, to which BN an application request should be forwarded to (either via L3 or L7 paths). These metrics associate potentially available paths with quantitative weights in the [0, 1] range (greater the weight, more suitable the path), adopting a lightweight end-to-end perspective. At session initialization, InternetClient discovers the available InternetServices and assigns them weights according to *PathLength* (basically greatest weights to shortest paths):

$$w_i = \frac{1 - (path_i Length / averageLength / \# paths)}{\# paths - 1}$$

where `path`$_i$`Length` is the number of `path`$_i$ hops and `averageLength` is the average length of the `#paths` available paths. Shortest path priority pushes traversing traffic to a limited set of nodes. In addition, while we are aware that path throughput depends on a wide set of parameters, such as adopted wireless technologies and traffic load, based on our previous work we believe that path length can also provide a rough estimation of the maximum bandwidth achievable, useful to quickly take an initial configuration decision in a very lightweight way [7].

L3SP exploits these weights to evaluate the most suitable BN: InternetClients running in the same neighborhood have a homogeneous vision of available paths, thus supporting the formation of a path evaluated as suitable for the whole locality (see weights in Figure 4). In fact, *PathLength* tends to partition the network in different parts; in this way, it is scarcely probable that neighbors try to activate conflicting L3 paths. In the current RAMP implementation, in the case of multiple paths with the same weight, the user has to explicitly select the preferred one. Path reconfiguration is triggered only in case of connectivity disruption, to prevent from disturbing working connections of other nodes. L7MP and L3L7CMP exploit weights to decide how to proportionally partition the request load among the available paths: for instance, considering NodeC in Figure 4, InternetClient computes $w_{BN1} = 0.4$ and $w_{BN2} = 0.6$; thus, every 5 application requests, it exploits BN_1 two times and BN_2 three times.

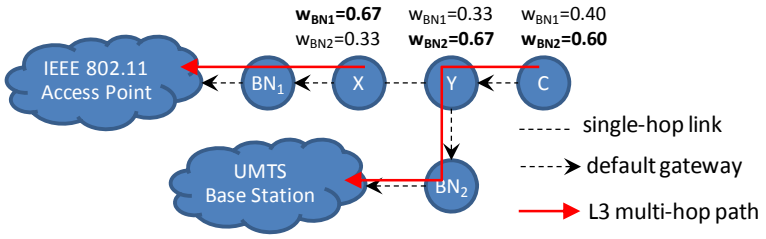


Figure 4. PathLength metric application (selected BN in bold style).

In addition, in L7MP and L3L7CMP modes, InternetClient monitors end-to-end throughput at service provisioning time and evaluates path quality via the *PathThroughput* metric. In particular, InternetClient keeps track of:

$$\frac{requestPayload + responsePayload}{elapsedTime}$$

values with per-connection granularity. Thus, it can achieve an approximated but lightweight estimation of path performance, with no additional communication overhead. Based on these values, InternetClient periodically (whenever it has gathered 20 throughput values for one of its paths) reassigns weights to paths by adopting the following *PathThroughput* metric:

$$w_i = path_i Throughput \,/\, averageThroughput \,/ \# paths$$

where `path`$_i$`Throughput` is the throughput of path `i` in the last time window and `averageThroughput` the average throughput of the `#paths` available paths. For instance, if BN_1 and BN_2 offer 25 and 10KB/s throughput respectively, w_{BN1}/w_{BN2} is equal to 0.71/0.29; therefore, 71% of the connections will exploit the former path, 29% the latter. L7MP and L3L7CMP do not partition the network topology as L3SP does: any node can exploit all the visible BNs depending on its locally perceived connectivity quality. For instance, if BN_2 becomes overloaded, NodeY assigns a higher weight to (and thus exploits more frequently) BN_1, by leaving almost all BN_2 bandwidth to NodeC, which could evaluate BN_2 as the most suitable. Let us note that, as better detailed in Section 5, the overall achieved throughput may depend also on factors not strictly related to spontaneous network path performance, e.g., payload size and HTTP server load, and *PathThroughput* can achieve good performance estimation anyway, by adopting a very lightweight and completely distributed approach.

4 RAMP Internet Connectivity Service: Architecture and Implementation Insights

RAMP is designed according to a 2-layer architecture, with a higher Service Layer and a lower Core Layer (Figure 5-left). The former supports peer-to-peer service provisioning via registration, advertising, and discovery; the latter provides communication abstractions for end-to-end unicast and broadcast. In particular, Service Manager allows the registration and advertising of local applications (`registerLocalService`), while Discovery supports available remote services (`findRemoteServices`), by allowing the identification of the path toward the targeted node and the retrieval of its

capabilities. E2EComm offers multi-hop unicast and TTL-bound broadcast primitives (`receive`, `sendUnicast`, and `sendBroadcast`); Dispatcher interacts with single-hop neighbors (e.g., via UDP/TCP depending on what dynamically specified) to collaboratively route packets; Heartbeater works to keep track of single-hop neighbors by periodically inquiring the available subnets via UDP broadcast.

In addition, RAMP exploits the mechanisms developed within the Multi-hop Multi-path Heterogeneous Connectivity (MMHC) project, already presented elsewhere [7], for the dynamic setting of ad-hoc subnets (layer-2 link creation and layer-3 network configuration). MMHC provides the best multi-hop Internet connectivity via proper local configuration by exploiting innovative context indicators (e.g., probability of joint peer mobility) to maximize connectivity reliability, throughput, and availability. In particular, RAMP takes advantage of MMHC to create/manage heterogeneous single-hop links and to identify nodes. Each subnet is created in a distributed way, without the need of global scope visibility, and by considering any available interface (e.g., IEEE 802.11 and Bluetooth); nodes assign IP addresses to the subnets they have created without any need of coordination [7]. The former MMHC rerouting mechanisms manage multi-hop paths only based on modifications of default gateway configuration, e.g., to use the wireless interface with minimum energy consumption or maximum throughput. Instead, RAMP more flexibly dispatches packets at the application layer by exploiting every available single-hop link enabled by the underlying MMHC, with the valuable additional advantage of simultaneous exploitation of any available path, despite operating system-level configuration of routing tables.

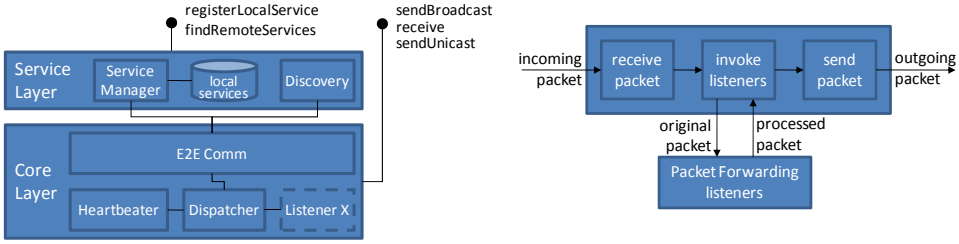


Figure 5. RAMP architecture (left) and activity flow in RAMP Dispatcher (right).

To the purpose of enabling the flexible introduction of any application-layer operation on RAMP transmitted packets at runtime, we have implemented the Dispatcher according to a listener-based architecture, which permits to efficiently and easily monitor and/or modify exchanged packets at any traversing node (Figure 5-right). The `addPacketForwardingListener` Dispatcher method permits to add and register listeners to monitor incoming packets. In this way developers can implement and deploy additional listener-based components to support novel features, by extending RAMP capabilities without any modification of the basic Dispatcher. A detailed and general description of the RAMP middleware is out of scope here (please see the RAMP Web site [6]); in the following we focus on the crucial and original technical aspects of the RAMP support for Internet connectivity sharing.

BNs willing to share Internet connectivity activate InternetService and register it via `registerLocalService`; clients requiring Internet connectivity activate InternetClient, which exploits `findRemoteServices` to discover nodes offering InternetSer-

vice. Service discovery is based on a TTL-bound broadcast research (default TTL=5), with neighbors that reply if they offer the required service; the reply message is used both to identify the service node (see below) and to invoke the service. This mechanism is suitable for medium-size networks, which is usually the case for the targeted spontaneous scenarios. In the rare case of very large-scale spontaneous networks (e.g., with hundreds of nodes and 20/30 hops diameter), it is possible to smoothly adopt more sophisticated discovery algorithms that cache information on intermediary nodes, analogously to AODV; anyway, this kind of wide-scale networks are not the primary RAMP target. L7 paths can be used via the `sendUnicast` primitive, which identifies the destination via the `dest` parameter, i.e., the ordered set of intermediary nodes composing the multi-hop path between sender and receiver. In fact, RAMP identifies a remote node via the IP addresses of the intermediary nodes in the path to that node. For instance, in Figure 6 NodeA identifies NodeB via the [2, 4, 6] sequence while NodeB identifies NodeA as [5, 3, 1] (sequences differ depending on path directions because different wireless ingress interfaces are exploited in the two ways).

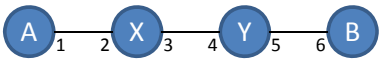


Figure 6. Node identification depending on traversed interfaces.

L3 path creation exploits the Layer3Manager component, registered as `PacketForwardingListener` to the local Dispatcher of every node along the client-to-BN path. A node requires to create an L3 path by sending (via InternetClient) a unicast Layer3Request packet to InternetService at the selected BN. On intermediate nodes, whenever Layer3Manager recognizes a traversing Layer3Request packet, it exploits `dest` and `currentHop` header fields to properly modify operating system-level routing rules. In particular, it sets as local default gateway the host in `dest` with position `currentHop` (the first host has index 0). For instance, in the path from NodeA to NodeB in Figure 6, when the Layer3Request packet reaches NodeX/NodeY, `currentHop` has value 1/2 and thus Layer3Manager sets NodeY/NodeB as default gateway. InternetService on the last node (e.g., NodeB) sends an ack to InternetClient to notify that the L3 path is ready, while its Layer3Manager component does not change its default gateway. Once the L3 path is ready, applications at the client node can adopt the L3SP mode to get Internet connectivity (no proxy).

On Linux nodes Layer3Manager exploits Linux `route` command to set the default gateway and `iptables` command to enable NAT traversal. For instance, on NodeX Layer3Manager executes:

```
route add default gw IP4 interf
iptables -t nat -A POSTROUTING -s IP1 -j MASQUERADE
```

where `IP1/4` is the IP address of the egress/ingress interface on NodeA/Y and `inter` is the name (e.g., eth0 or wlan0) of the NodeX interface with IP address `IP3`. In addition, at activation time, Layer3Manager temporarily enables operating system packet forwarding with the command

```
sysctl -w net.ipv4.ip_forward=1
```

In addition, InternetClient supports both L7MP and L3L7CMP acting as proxy and waiting at a well known port for local application requests, including the remote endpoint (IP address, TCP/UDP, and port number) and the payload to send. At each request, InternetClient, based on current weights, selects one of the available paths,

either L3 or L7: in the former case (only L3L7CMP) InternetClient contacts the end-point directly (single proxy); in the latter case it exploits `sendUnicast` and `receive` to send application requests and receive responses from the selected InternetService (double proxy).

InternetClient/Service can manage HTTP requests/responses. On the client-side, InternetClient parses requests to find the end-point IP address and port in the `Host` HTTP header. On the server-side, InternetService interacts with Web servers, receives their responses, and dispatches them to the client. In this way it is possible to transparently surf the Web in a spontaneous network by simply setting the local InternetClient as the Web browser proxy. Note that, given the instability of spontaneous networks, InternetClient uses `Connection:close` header, thus imposing to open new connections for each request/response pair. In addition, InternetClient/Service support a pseudo-HTTP format to allow also non-Web-based applications to exploit RAMP-based Internet connectivity: on the client-side, applications simply have to specify the standard `Host` and `Content-Length` headers plus our `Layer4Protocol` header with either `TCP` or `UDP` value, followed by an empty line and the payload, in either text or binary format (Figure 7); on the server-side, applications have only to indicate the desired `Content-Length` header and payload.

As summarizing implementation considerations, L3SP is certainly the simplest one: by directly exploiting L3-based connectivity, applications can access the Internet despite the adopted communication paradigm. Instead, L7MP and L3L7CMP currently support HTTP (and pseudo-HTTP), thus being easily applicable only to service components following the request/response communication paradigm. To support other application-level protocols, e.g., RTP for audio/video streams, there is the need to specifically add novel features to the RAMP InternetClient/Service. However, L3SP has a strong dependence on the underlying operating system, requiring modifying Layer3Manager to support different operating systems. For instance, the Layer3Manager version in the current RAMP prototype works only on Linux platforms. In addition, to enable packet forwarding and routing rule modifications, Layer3Manager requires running with administrator privileges (e.g., Linux superuser). This requirement can be viewed as a significant limitation. On the opposite, L7MP and L3L7CMP take advantage of available L7 paths with no need of administrator permissions and are available on any RAMP-enabled operating system.

```
Host: lia.deis.unibo.it:1234\r\n
Content-Length: 11\r\n
Layer4Protocol: TCP\r\n
\r\n
Hello World
```

Figure 7. Example of client-side pseudo-HTTP request.

5 Experimental Validation and Performance Results

To validate our Internet sharing solution, we have deployed and tested L3SP, L7MP, and L3L7CMP in the multi-hop multi-path spontaneous network depicted in Figure 8 (InternetClient resides on NodeC, InternetService on BNs). The targeted

scenario is simple for the sake of briefness and easy interpretation of the results reported in the following, but still complex enough to well point out the RAMP behavior in a real, multi-path, heterogeneous deployment environment. In the following, we mainly concentrate on L7MP and L3L7CMP because L3SP performance almost entirely depends on available bandwidth (limited influence of RAMP performance) and because L3 paths are less frequently used in highly dynamic spontaneous networks.

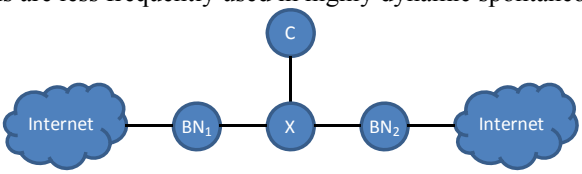


Figure 8. Testbed deployment scenario.

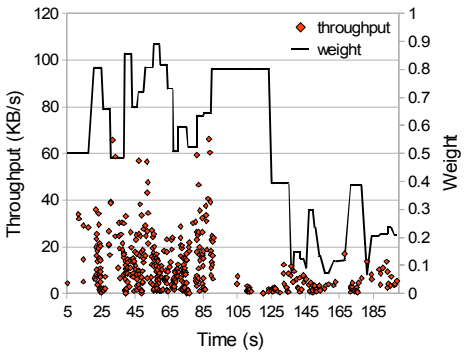
The nodes used in the tests are Core2 2.6GHz laptops with 2.0GB RAM, running MSWinXP (only L7MP) or LinuxDebian (both L7MP and L3L7CMP). NodeC-NodeX link is based on Ethernet (bandwidth limited to 2Mbit/s); all the other single-hop links are on IEEE 802.11b (infrastructure and ad-hoc modes) with CISCO access points and Orinoco cards (available bandwidth set as specified in the following); all the reported performance figures are representative examples selected over 100 runs.

InternetClient takes about 72ms to discover the two InternetServices. The starting value for weights is 0.5 for both BNs, since they are both two-hop distant (*PathLength* metric). We have tested RAMP while supporting the access of a standard Firefox/Iceweasel Web browser to Google Maps, by exploiting the local InternetClient as proxy. Google Maps, like many Web applications, is characterized by frequent interactions (up to 12 interactions/s), with relatively limited payload (from 1.1 to 29.45KB per request/response, 12.04KB average size, 7.52 standard deviation). This pattern of interaction is particularly challenging for spontaneous networking (frequent different interactions, each one with small-size payload); that is the reason why we have selected it, in order to evaluate the RAMP middleware under stress in a sort of worst case scenario in terms of application traffic type.

In particular, we have collected experimental results about i) the throughput achieved by NodeC and ii) the time evolution of computed weights, when adopting L7MP (Figure 9a) and L3L7CMP (Figure 9b). The goal is to quantitatively evaluate how well RAMP can adapt its behavior dynamically according to actual in-the-field path quality, at the same time by estimating the overhead associated with L7 paths.

About L7MP (Figure 9a), we have initially set BN_1 bandwidth to 125KB/s and BN_2 one to 25KB/s. Throughput depends on many factors, such as Web server load and payload size; these elements have demonstrated to be the main motivation why BN_1 and BN_2 throughputs resulted quite low, independently of RAMP performance. However, by focusing on the most interesting comparison between the two, BN_1 throughput has demonstrated to outperform BN_2 in many cases, due to the wider bandwidth of the RAMP path toward BN_1 if compared with BN_2. In addition, based on the monitored throughput, after about 20s, InternetClient increases/decreases BN_1/BN_2 weights respectively to 0.81 and 0.19 (*PathThroughput*), thus pushing to exploit BN_1 more frequently than BN_2. At t=105s, we have inverted bandwidth allocation (25KB/s for BN_1, 125KB/s for BN_2) to test the RAMP capability of dynamic adaptation: InternetClient has demonstrated to be able to react promptly with proper

weight modifications notwithstanding our lightweight monitoring approach simply based on application-level throughput observation.



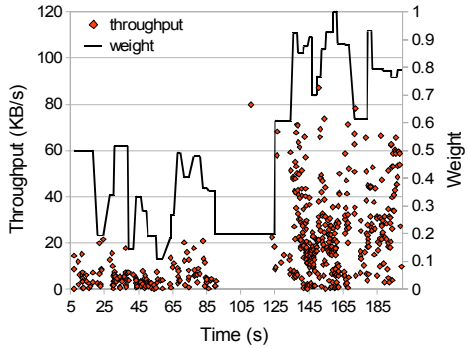
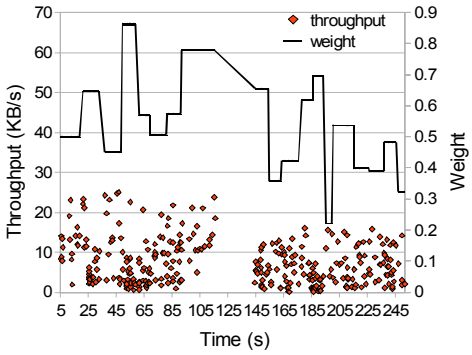


Figure 9a. Throughput and weights for L7MP BN_1 (up) and BN_2 (down).



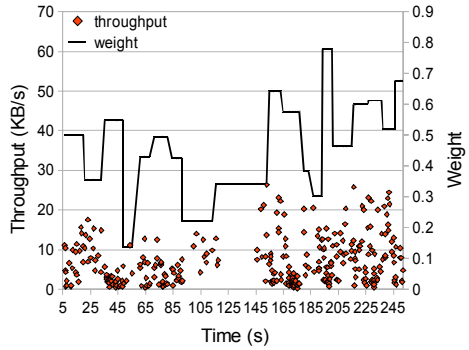


Figure 9b. Throughput and weights for L3L7CMP BN_1 (up) and BN_2 (down).

About L3L7CMP (Figure 9b), we have allocated the same bandwidth to BN_1 and BN_2 (30 KB/s) and observed RAMP behavior when creating an L3 path toward either BN_1 (before t=125s) or BN_2 (after t=125s). RAMP has demonstrated to require about 298ms to create the two-hop L3 path, measured on InternetClient (from request message to ack reception). It is worth noting that RAMP tends to provide greater priority to L3 paths than to L7 ones, given the usual slightly greater throughput associated to L3 paths (up to 27KB/s vs. 18KB/s). In other words, the RAMP middleware actually perceives that the L3 path provides a slightly greater throughput than the L7 one, and dynamically adapts its behavior by exploiting more frequently the former instead of the latter. However, the throughputs actually achieved via L3 and L7 paths, when measured in-the-field, are frequently almost equivalent (and accordingly the weights tend to become similar), because they tend to mainly depend on Web server load and payload size, as already stated. At the same time, this demonstrates the limited overhead introduced by RAMP when managing packet forwarding at the application level.

6 Related Work

Several proposals have investigated specific partial aspects of more "traditional" multi-hop connectivity: a few recent works are starting to propose the synergic and simultaneous exploitation of heterogeneous wireless interfaces at mobile terminals; most have focused on one specific technology, such as IEEE 802.11 or GPRS/UMTS. However, their primary accent is on seamless connectivity in environments where heterogeneous wireless technologies are integrated. For instance, [8] aims at extending cellular networks via relay stations to increase coverage; [9, 10], instead, specifically address client mobility management in heterogeneous multi-hop networks.

Different aspects of spontaneous networking have been addressed by a number of research activities in the recent literature; here, we focus only on the projects more closely related to RAMP. By focusing on multi-hopping in spontaneous networks, some contributions aim at increasing connection quality via low-level solutions. For instance, [1] improves wireless medium exploitation by opportunistically accessing the available spectrum. [2] optimizes bandwidth allocation by differently managing real-time and best-effort transmissions. The proper support of multi-path connectivity has gained increasing attention only very recently. Some proposals determine the best route towards a destination by exploiting evaluation metrics based on low-level context [11]. Others exploit network-layer context to estimate current path load and to appropriately distribute generated traffic among the available paths [3]. Some proposals specifically focus on multimedia streaming via multi-path channels, with the main scope of improving stream quality via rate allocation algorithms that properly interact with the operating system [4]. Finally, opportunistic and delay-tolerant networking is emerging as an interesting approach for connectivity in highly dynamic spontaneous networks [1]. For instance, [12] supports opportunistic data delivery in intermittently connected mobile ad hoc networks. However, the proposal in [12] is only based on simulations and only considers homogeneous networks with plain addressing.

In short, most related contributions in the literature aim at supporting spontaneous networking mainly in homogeneous networks, by introducing non-standard modifications to layer-2 protocols. In addition, they do not address the heterogeneity issues associated with the exploitation of multiple interfaces, with IP addressing, and with the specific support of Internet connectivity sharing services.

7 Conclusions

The effective and appropriate collaborative sharing of Internet connectivity can be a "killer application" for spontaneous networking, by relevantly promoting its adoption and enabling a better exploitation of the growing amount of computing/memory/bandwidth resources available on portable wireless terminals. This work makes a further step toward this vision and demonstrates that i) both network- and application-layer approaches are suitable and ii) the decision of which mode to adopt should be taken at service provisioning time based on application-specific requirements and spontaneous network performance evaluation. In particular, the RAMP L3SP mode (based on L3 approach) has demonstrated to be adequate for its transpar-

ency to application protocols and interaction paradigms; however, it may suffer from frequent topology changes due to the costs of L3 path reconfiguration. Instead, RAMP L7MP and L3L7CMP modes have demonstrated their suitability for highly dynamic network environments: simultaneous exploitation of multiple paths improves connectivity reliability and quality. In particular, the L3L7CMP mode can exploit both L3 and L7 approaches simultaneously. In addition, our in-the-field performance results demonstrate that RAMP imposes limited overhead if compared with more traditional network solutions based only on routing at the operating system-level.

The encouraging results already achieved are stimulating further research activities on spontaneous networking. In particular, we are validating an extended version of the RAMP prototype that supports application-layer splitting of multimedia streams via differentiated paths, in order to both increase throughput and minimize packet loss rate. In addition, we are working on enhancing the RAMP support to peer fairness through the adoption of innovative distributed trust management solutions based on lightweight monitoring/evaluation of users' behavior (resource offers/requests).

References

1. Salameh, H.B., Krunz, M.: Channel Access Protocols for Multihop Opportunistic Networks: Challenges and Recent Developments. IEEE Network, Vol. 23, No. 4, pp. 14--19, July-Aug. 2009.
2. Wu, H., Liu, Y., Zhang, Q., Zhang, Z.L.: SoftMAC: Layer 2.5 Collaborative MAC for Multimedia Support in Multihop Wireless Networks. IEEE Trans. on Mobile Computing, Vol. 6, No. 1, pp.12--25, Jan. 2007.
3. Chai Keong Toh; Anh-Ngoc Le; You-Ze Cho: Load balanced routing protocols for ad hoc mobile wireless networks. Communications Magazine, IEEE , vol.47, no.8, pp.78--84, August 2009.
4. Frossard, P.; de Martin, J.C.; Reha Civanlar, M.: Media Streaming With Network Diversity. Proceedings of the IEEE , vol.96, no.1, pp.39--53, Jan. 2008
5. de Amorim, M.D., Ziviani, A., Viniotis, Y., Tassiulas L. (eds.): Special Issue on Practical Aspects of Mobility in Wireless Self-organizing Networks. IEEE Wireless Communications, Vol. 15, No. 6, Dec. 2008.
6. RAMP Web site, lia.deis.unibo.it/Research/RAMP
7. Bellavista, P., Corradi, A., Giannelli, C.: Mobility-aware Middleware for Self-Organizing Heterogeneous Networks with Multi-hop Multi-path Connectivity. IEEE Wireless Communications, Vol. 15, No. 6, pp. 22--30, Dec. 2008.
8. Le, L., Hossain, E.: Multihop Cellular Networks: Potential Gains, Research Challenges, and a Resource Allocation Framework. IEEE Communications, Vol. 45, No. 9, pp.66--73, Sep. 2007.
9. Pack, S., Shen, X., Mark, J.W., Pan, J.: Mobility Management in Mobile Hotspots with Heterogeneous Multi-hop Wireless Links. IEEE Communications, Vol. 45, No. 9, pp.106--112, Sep. 2007.
10. Lam, P.P., Liew, S.C.: Nested Network Mobility on the Multihop Cellular Network. IEEE Communications, Vol. 45, No. 9, pp.100--104, Sep. 2007.
11. Campista, M.E.M. et al., Routing Metrics and Protocols for Wireless Mesh Networks. IEEE Network, Vol. 22, No. 1, pp.6--12, Jan.-Feb. 2008.
12. Musolesi, M., Mascolo, C.: CAR: Context-Aware Adaptive Routing for Delay-Tolerant Mobile Networks. IEEE Trans. Mobile Computing, Vol. 8, No. 2, pp. 246--260, Feb. 2009.