# Walking with the Oracle: Efficient Use of Mobile Networks through Location-Awareness

Stefano Ferretti, Vittorio Ghini, Moreno Marzolla, Fabio Panzieri
Department of Computer Science, University of Bologna, Mura A. Zamboni 7, I-40127 Bologna, Italy
{sferrett, ghini, marzolla, panzieri}@cs.unibo.it

*Abstract*—**Always Best Packet Switching (ABPS) is a novel approach for wireless communications that enables mobile nodes, equipped with multiple network interface cards (NICs), to dynamically determine the most appropriate NIC to use. Using ABPS, a mobile node can seamlessly switch to a different NIC in order to get better performance, without causing communication interruptions at the application level. To make this possible, NICs are kept always active and a software monitor constantly probes the channels for available access points. While this ensures maximum connection availability, considerable energy may be wasted when no access points are available for a given NIC. In this paper we address this issue by investigating the use of an "oracle" capable of providing information on network availability. This allows to dynamically switch on/off NICs based on reported availability, thus reducing the power consumption. We present a Markov model which allows us to estimate the impact of the oracle on ABPS: results show that significant reduction in energy consumption can be achieved with minimal impact on connection availability. We conclude by describing a prototype implementation of the oracle based on Web services and geolocalization.**

## I. Introduction

The current generation of mobile devices (smartphones, tablets, laptops) is frequently equipped with several Network Interface Cards (NICs) (e.g. 3G, WiFi). The user must manually select which interface to use at any given time, according to the availability of Access Point (AP) within communication distance. In order to make better use of multiple interfaces, we recently proposed the Always Best Packet Switching (ABPS) mechanism [1]–[3]. ABPS keeps all interfaces active at the same time, and uses the best one for data transmission according to predefined criteria (e.g., availability or connection quality). Furthermore, ABPS can automatically switch to a different NIC when the current connection terminates; this is carried out transparently to the applications, and without any user intervention. A software component running on the Mobile Node (MN) dynamically selects a "preferred" NIC to use based on the available access points. If the performance of that preferred NIC degrades, or the connection fails, the MN switches to a different NIC. Specifically, for each datagram, the MN selects the best NIC to use.

A prototype implementation of ABPS is available [3], and supports interactive multimedia services based on SIP/RTP/RTCP. Availability and reliability of ABPS have been studied in [1], while power consumption has been investigated in [2]. A particular aspect of ABPS is that a software module at the MN continuously monitors all the NICs and probes the communication channels looking for access points to connect to. When there are no access points for a specific NIC in the current area, keeping that interface active has no practical benefit and negatively affects the MN battery duration [4].

In this paper we propose an extension to ABPS based on an additional software component–called "oracle"–that is responsible for predicting the availability (or lack of) of access points for the available interfaces. Using this information, the MN can decide to switch on or off a NIC base on its reported availability. We describe a prototype implementation of the oracle based on a software daemon which periodically queries the WiGLE Web service [5] to locate WiFi networks within the current MN location.

To evaluate the impact of the oracle on connection availability, power consumption and throughput, we define a performance model based on Continuous Time Markov Chains (CTMCs) with rewards. The model allows us to efficiently analyze the system in different scenarios; such analysis would be impractical using conventional measurement-based techniques. Results show that the oracle can significantly reduce the power consumption of the MN, with just a marginal reduction of availability and throughput with respect to plain ABPS. We believe that the performance model is useful by itself, as it can be easily extended to describe multiple alternative implementations of the oracle, allowing "what-if" analysis to be carried out efficiently.

This paper is organized as follows. Section II briefly introduces the ABPS mechanism and describes the extension based on the oracle. In Section III we describe an initial prototype implementation of the oracle based on the WiGLE service. Section IV describes the Markov model exploited to analyze the approach. The model is evaluated in Section V. Finally, concluding remarks are given in Section VI.

## II. The System

*The Basic ABPS Mechanism:* The idea behind ABPS is simple: during transmission and reception, the MN can use all the available NICs simultaneously, differentiating the choice of the actual NIC on a datagram-by-datagram basis. This is carried out transparently to both the application running in the MN and its peer at the Correspondent Node (CN);

This work has been partially funded by the Italian Ministry of Education, University and Research (MIUR) through project STEM-Net (Prot. 2009S7RLXY_003 - Year 2009). A revised version of this paper appears in the proceedings of Wireless Days 2012, Novembre 21–23 2012, Dublin, Ireland.

thus, applications can continue to employ classic end-to-end application-layer protocols such as TCP regardless of ABPS.

In order to let the data flow through different NICs, and to deliver the related contents as a single flow to the application, additional software components are needed. Specifically, ABPS requires the *ABPS Client Proxy* and the *ABPS Server Proxy* (see [3] for details). The Client Proxy runs on the MN and maintains the seamless multi-path communication channel between the MN and the Server Proxy running on the CN. The Server Proxy operates as a relay between the MN and the CN, i.e. it collects all the datagrams coming from the MN via different NICs, and sends them to the application running on the CN (application which is unaware of the presence of the ABPS Server Proxy).

*An Oracle to Predict Network Availability:* The ABPS uses multiple NICs at the same time, and assumes that all of them are always active (although possibly not always connected to an AP). This ensures maximum connection availability [1], and in some cases the use of ABPS can even reduce the power consumption with respect to a classic SIP based approach [2]. However, it remains true that the monitor at the ABPS client proxy continuously probes the communication channels looking for new access points to connect to. When there are no access points available in the current area, this procedure is useless and wastes energy, which is a particularly serious issue for battery-powered mobile devices. To avoid such undesired situation, we need an additional component (the "oracle") providing information on the available access points for a given NIC, without explicitly probing the communication channel. If the oracle reports that, for some NIC, no access point will be available for a sufficiently long period of time, then that NIC can be safely switched off. On the other hand, if the oracle detects that the MN is entering an area where access points are available, the NIC can be reactivated.

*The Algorithm:* Before describing how the oracle may be implemented, let us focus on how the oracle can be used by considering a practical scenario. We start by observing that most current smartphones are typically equipped with both a WiFi and a 3G NIC (e.g. UMTS). In general, 2G/3G network coverage is quite ubiquitous, while WiFi access points tend to be concentrated within densely populated areas (e.g., city centers). On the other hand, WiFi provides better bandwidth, lower latency and lower energy consumption [2], therefore UMTS networks could be used as a fallback when no usable WiFi connection is available.

However, if the MN is entering an area where the WiFi connection is likely to be available for a short time only (e.g., because the MN is approaching an isolated access point), switching off the UMTS NIC would not be a good idea since UMTS will be needed again in a short time. On the other hand, if the MN is entering an area where WiFi coverage is guaranteed for a sufficiently long time, then it is reasonable to switch off UMTS in order to save energy.

**Algorithm 1** ABPS Oracle

```
 1: upon EV_NO_WIFI                          {no available WiFi network}
 2:     $active_{\mathrm{WiFi}} \leftarrow false$
 3:     $active_{\mathrm{UMTS}} \leftarrow true$
 4: upon EV_SHORT_WIFI                 {WiFi available for a short period}
 5:     $active_{\mathrm{WiFi}} \leftarrow true$
 6:     $active_{\mathrm{UMTS}} \leftarrow true$
 7: upon EV_LONG_WIFI                   {WiFi available for a long period}
 8:     $active_{\mathrm{WiFi}} \leftarrow true$
 9:     $active_{\mathrm{UMTS}} \leftarrow false$
```

**Algorithm 2** ABPS Client Proxy

```
 1:                                                    {NICs management}
 2: upon available connection for $NIC \wedge active_{\mathrm{NIC}}$
 3:     SETUP($NIC$)
 4: upon successful setup for $NIC \wedge active_{\mathrm{NIC}}$
 5:     $availableNICs$.ADD($NIC$)
 6:     $inUseNIC \leftarrow$ SELECTPREFERRED($availableNICs$)
 7: upon disconnection for $NIC$
 8:     $availableNICs$.REMOVE($NIC$)
 9:     $inUseNIC \leftarrow$ SELECTPREFERRED($availableNICs$)
10:                                                      {Communication}
11: upon new $data$ to transmit
12:     SEND($data, inUseNIC$)
13:     SETTIMEOUT($data$)
14: upon $data$ received from ABPS server proxy
15:     PUSHORDEREDBUFFER($data, bufferToApp$)
16:     SENDORDEREDDATA($bufferToApp$)
17: upon ACK for $data$ received
18:     REMOVETIMEOUT($data$)
19: upon timeout ACK for $data$
20:     $NIC \leftarrow availableNICs$.SELECTALTERNATIVENIC()
21:     SEND($data, NIC$)
22:     SETTIMEOUT($data$)
```

We assume that the oracle has a list of available WiFi access points in the current area; also, the oracle knows the current position of the MN, and can also infer the future destination of the node (see Section III for a possible implementation). Algorithm 1 describes the high level behavior of the oracle. Based on the list of available WiFi networks, the oracle generates one of these events:

- EV_NO_WIFI: no WiFi access point is available; therefore, the WiFi NIC can be turned off so that connectivity is provided by the 2G/3G network only (lines 1–3).
- EV_SHORT_WIFI: a WiFi connection is going to be available for a short period of time; in this case both the UMTS and WiFi NICs are activated (lines 4–6).
- EV_LONG_WIFI: a WiFi connection is going to be available for a long period of time; in this case the MN can safely switch off UMTS (lines 7–9).

The events generated by the oracle are exploited by the ABPS Client Proxy as shown in Algorithm 2. When a given NIC is active, based on the oracle prediction, and if an access point is detected for that NIC, then the Client Proxy starts a setup procedure to activate a connection and use it. Upon successful connection setup, the Client Proxy includes that interface in a list of NICs available to transmit data. When a connection drops, the associated NIC is removed from the list of active interfaces.

Based on the choice of NIC to use, the communication between the MN and its CN is as follows. The Client Proxy intercepts and manages data coming from/to the application (lines 10–22). When there is data to transmit, the Client sends the data to the Server Proxy via the preferred NIC in use. As multiple networks can be used during the communication, it is possible that the transmitted datagrams are received out of order. To this end, a sequence numbering scheme is used that enables the receiver Server Proxy to order the received datagrams and discard possible duplicates. Finally, the Client Proxy exploits ACKs to identify if some datagram is to be re-transmitted. When a timeout for the reception of an ACK occurs, the Client Proxy tries to retransmit that datagram through a different NIC among those active at that time.

The ABPS Server Proxy runs on a remote host, separate from that of the Client Proxy. The Server Proxy operates basically as a relay between the MN and its CN. In practice, it is convenient to place the Server Proxy on a fixed node located outside any firewall and NAT system. Then, the ABPS Client and Server Proxies collaborate and adopt policies for load balancing and recovery, in order to maximize throughput and minimize loss rate and economic costs. In particular, the ABPS Server Proxy collects all the datagrams coming from the MN, via different NICs and sends them to the CN (which is unaware of the presence of the ABPS Server Proxy).

## III. Prototype Implementation

*WiFi Networks Detection with WiGLE:* We have developed a prototype software module that is in charge of periodically querying the WiGLE Web service to locate WiFi networks within the area of the mobile user. Wireless Geographic Logging Engine (WiGLE) is a submission-based Web catalog of wireless networks [5]. Such service provides location and information of wireless networks world-wide; it is possible to map, query and update the databse through a java application or a Web browser. WiGLE allows users to obtain statistics on the performance of networks, and for instance, it is possible to ask only for free (or commercial) networks. We have developed a Python script in charge of retrieving from WiGLE a list of WiFi access points available in the area of the MN, based on its geographical position, together with some related information.

To obtain the actual position of the MN, a positioning system must be used, i.e. GPS. It has been shown that the power consumption related to the use of a GPS is noticeably lower than that due to the use of a NIC [6], but in any case the evaluation described in Section V explicitly takes into account the additional energy consumption of the GPS module and the associating processing cycles. It is worth noticing that many apps running on users' smartphones require the use of GPS to locate points of interests [7]. Hence, it would be reasonable to assume that GPS is likely to be active anyway even without the execution of the oracle. In this case, the geolocalizaion does not introduce additional power consumption.

*Use Cases:* We report a couple of examples in order to show the benefits of having an oracle that provides the list of WiFi networks, and also some main related issues.

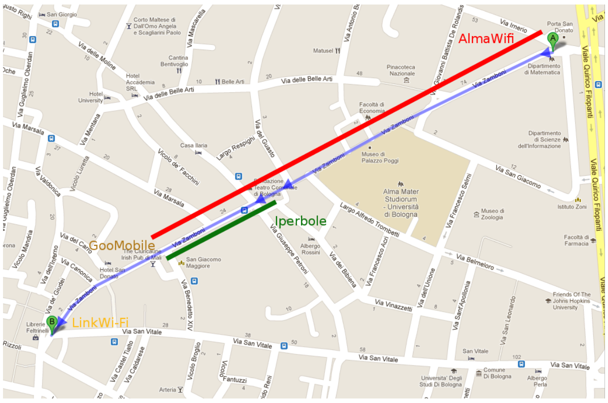


Fig. 1. WiFi networks in Via Zamboni, Bologna (Italy)

Let consider a mobile user that covers the path depicted in Fig. 1. The path is Via Zamboni in Bologna (Italy), the main street in the city center where several departments of the University of Bologna are located. In this area there are many active WiFi access points. In particular, there are two main networks available for students, i.e. the AlmaWiFi network, which is the main WiFi network for all students, faculty members and employees of the University of Bologna, and the Iperbole civic network, available to all citizens. For each point in the path, the oracle provides a list of available access points. For the sake of a clearer presentation, we report only the essid of the access points with the best signal in six main points of the path, i.e. those corresponding to a square. The red line in the Figure corresponds to the area in the path where there is a network coverage provided by the AlmaWiFi Network; the green line corresponds to the part of the path that is covered by some access points of the Iperbole network. Other networks are available in the network coverage of a single access point.

This example allows us to make some important considerations. Knowing that a main part of the area is covered by a single WiFi network (AlmaWiFi), the network manager of the MN might decide to switch off its UMTS interface. Obviously, during the path the mobile terminal will perform horizontal handover through different access points of the same network, but this is accomplished at the datalink layer, transparently to the transport and application layers. Then, once the user becomes in proximity of the end of the AlmaWiFi network coverage, the UMTS interface should be activated, so as to allow the MN to exploit that interface during the handover from a WiFi network to another one, or if no WiFi networks are available or open to the user.

Fig. 2 shows a different path, starting from Giardini Margherita, the main park in Bologna, and then straight to via Murri, a street originating from the entrance of the park. In this case, the oracle reports that there are private networks only (we do not show them in the picture); moreover, the access points are located at the beginning and the end of the path. Apparently, no WiFi networks are available in the middle of the path. In this context, the best strategy is to keep the UMTS NIC always active; the WiFi interface can be switched

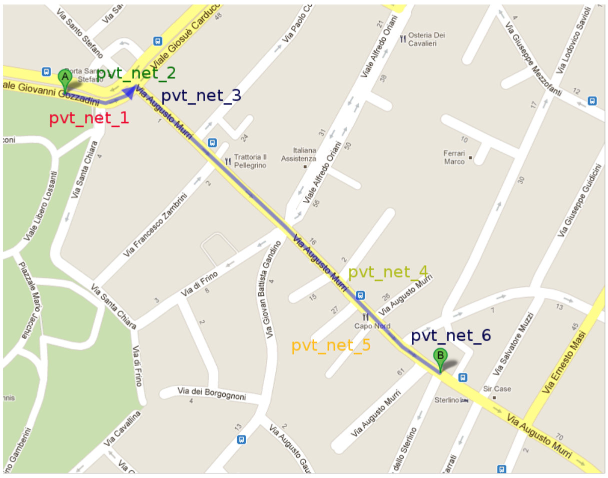


Fig. 2. WiFi networks in Via Murri, Bologna (Italy)

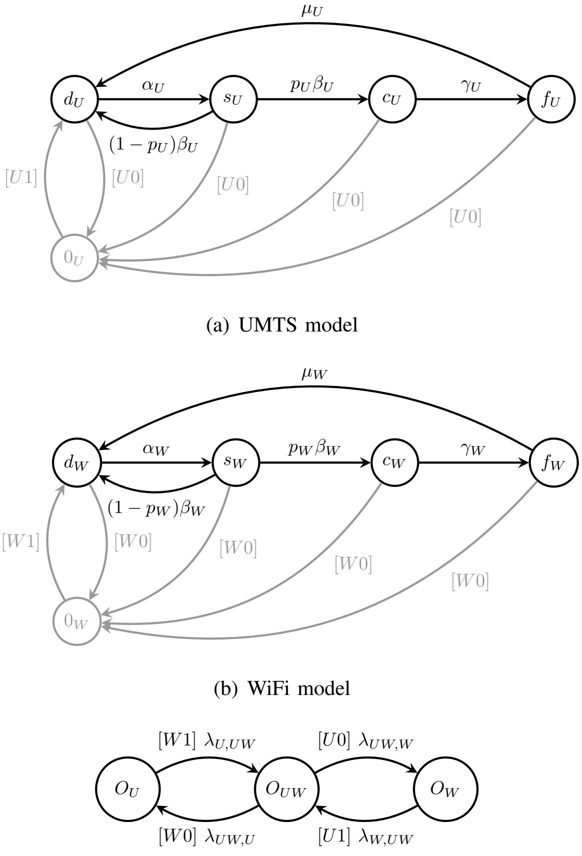
on when the MN is in proximity of the beginning and the end of the path, so that it can probe the available networks and see if these networks are open and available.

## IV. Modeling ABPS and the Oracle

In this section we describe a performance model for ABPS with oracle; the model is based on Markov chains with rewards, and extends previous works on modeling ABPS [1], [2]. Our model can be solved very quickly, and can be used to evaluate the proposed approach under different parameter settings. We use the model to estimate the impact of the oracle on the availability, power consumption and throughput of ABPS. The insights provided by the model are being used to drive current and future development of the oracle-enhanced ABPS prototype.

*Markov chains:* We start by giving a very brief introduction to Markov chains (see [8] for a more complete description). A stochastic process $\{X(t), t \geq 0\}$, defined over the discrete state space $\{1, \ldots, N\}$, is a CTMC if the probability that the system is in state $X(t_{n+1})$ at time $t_{n+1}$ only depends on the previous state $X(t_n)$ at time $t_n$, for any $t_n < t_{n+1}$. A CTMC can be fully defined in term of an *infinitesimal generator matrix* $\mathbf{Q} = [Q_{i,j}]$, where $Q_{i,j}$ is the transition rate from state $i$ to state $j \neq i$. Of particular interest for this paper are *Markov reward models*. We can associate to each state $i \in \{1, \ldots, N\}$ a nonnegative reward $r_i$. Rewards have the following meaning: for each period of duration $dt$ spent in state $i$, the total accumulated reward increases by $r_i dt$.

*Model Description:* We model ABPS with oracle using the three interacting CTMCs shown in Fig. 3. The chain in Fig. 3(a) represents the UMTS NIC, the one in Fig. 3(b) represents the WiFi NIC, and finally the chain in Fig. 3(c) represents the oracle. The UMTS and WiFi models are identical (except for the parameter values), so we focus on the UMTS chain shown in 3(a). The chain is defined over the state space $\{0_U, d_U, s_U, c_U, f_U\}$; note that, for readability, we



(a) UMTS model

(b) WiFi model

(c) Oracle model

Fig. 3. Markov models of ABPS with oracle

give symbolic names to states instead of numeric IDs. State $0_U$ (*off*) denotes that the UMTS NIC is shut down. In state $d_U$ (*disconnected*) the UMTS interface is active and scans for networks to connect to. A network is found after an average time $1/\alpha_U$, and the NIC enters state $s_U$ (*setup*) where it tries to connect to the newly found access point. A connection attempt has an average duration of $1/\beta_U$, after which it succeeds with probability $p_U$ and fails with probability $(1 - p_U)$. After a successful connection, the UMTS NIC enters state $c_U$ (*connected*); in this state data packets can be received and transmitted. The average duration of a connection is $1/\gamma_U$. When the connection is lost, the UMTS interface enters state $f_U$ (*failed*). In this state no datagrams can be received or delivered, but the MN is not yet aware of such failure and continues to send data until it detects, eventually, that the connection is not available and enters state $d_U$. The oracle is modeled by the chain in Fig. 3(c). State $O_U$ is entered when the EV_NO_WIFI event is fired (see Algorithm 1); in this state no WiFi access points are available, therefore only the UMTS NIC is kept active. State $O_{UW}$ is entered when the EV_SHORT_WIFI event is fired; here both UMTS and WiFi NICs are active. Finally, state $O_W$ corresponds to event EV_LONG_WIFI, denoting that a WiFi connection is available for a longer period of time; therefore, the UMTS interface can be seitched off and only WiFi is kept active.

Some of the transitions shown on Fig. 3 are labeled with *actions* within square brackets. Actions force two chains to make simultaneous transitions, at a rate equal to the product of the rates of synchronized transitions (when omitted, the rate is 1). Synchronized transitions model the fact that the oracle may affect the behavior of the UMTS or WiFi chain. Actions $U1$ and $U0$ are used to switch on and off the UMTS interface. Action $U1$ is executed when the transition $O_W \to O_{UW}$ is traversed, forcing the transition $0_U \to d_U$ on the UMTS chain. Action $U0$ is executed when the oracle traverses the transition $O_{UW} \to O_W$, forcing the UMTS chain to move to state $0_U$ (*off*) from whichever state it was in. Actions $W1$ and $W0$ have the same effect on the WiFi NIC.

The oracle has also another effect on the WiFi model, which is not shown in the figure to reduce visual clutter. The value of $\gamma_W$, which controls the average duration of WiFi connections, depends on the state of the oracle. Recall that state $O_W$ is entered when the event EV_LONG_WIFI is fired; therefore, in state $O_W$ the WiFi connection should last longer. On the other hand, state $O_{UW}$ is entered when the event EV_SHORT_WIFI is fired, and in this case the WiFi connection should have shorter duration. Let $T_W^+$ and $T_W^-$ be the mean duration of long and short WiFi connections. The transition rate $\gamma_W$ from $c_W$ to $f_W$ is defined as follows: if the oracle is in state $O_W$, then $\gamma_W = \gamma_W^+ = 1/T_W^+$, otherwise $\gamma_W = \gamma_W^- = 1/T_W^-$. With these rules the mean sojourn time in state $c_W$ is either $T_W^+$ or $T_W^-$.

The complete model of ABPS with oracle is built by composing the three individual chains shown in Fig. 3. The state space is the set of triples $(S_{\text{WiFi}}, S_{\text{UMTS}}, S_{\text{oracle}})$, where $S_{\text{UMTS}} \in \{0_U, d_U, s_U, c_U, f_U\}$, $S_{\text{WiFi}} \in \{0_W, d_W, s_W, c_W, f_W\}$ and $S_{\text{oracle}} \in \{O_U, O_{UW}, O_W\}$. Not all configurations are possible (e.g., when $S_{\text{oracle}} = O_W$ we know that UMTS must be in state *off*, hence $S_{\text{UMTS}} = 0_U$), therefore the state space is actually smaller. The model has been implemented using the PRISM model checker [9] and can be found in [10].

## V. Analysis

We use the Markov model to study how the oracle affects availability, power consumption and throughput with respect to plain ABPS. Plain ABPS is modeled by removing all gray elements from the models in Fig. 3.

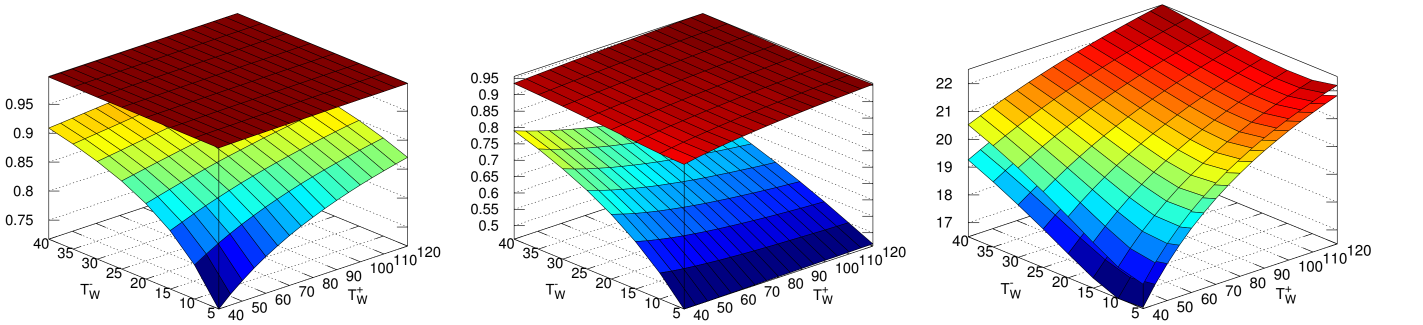
*Model Parameters:* The following parameter values have been empirically determined:

| | | |
|---|---|---|
| $\alpha_U = 1/6.024$ | $\alpha_W = 1/7.5$ | $\lambda_{U,UW} = 1/30$ |
| $\beta_U = 1/1.5$ | $\beta_W = 1/1.5$ | $\lambda_{UW,W} = \gamma_W^-/2$ |
| $\gamma_U = 1/500$ | $\gamma_W^- \in [1/5, 1/40]$ | $\lambda_{UW,W} = \gamma_W^-/2$ |
| | $\gamma_W^+ \in [1/40, 1/120]$ | $\lambda_{W,UW} = \gamma_W^+$ |
| $\mu_U = 1$ | $\mu_W = 1$ | |
| $p_U = 0.99$ | $p_W = 0.9$ | |

Transition rates are defined as the inverse of experimentally measured values. We set the duration of short WiFi connections (those for which the oracle triggers the EV_SHORT_WIFI event) in the range $[5s, 40s]$; long WiFi connections are set in the range $[40s, 120s]$. We keep the values of other parameters fixed, since the time needed to set up a connection, or detect that a connection has failed, are likely to be unaffected by the duration of WiFi connections. Transition rates for the oracle have been set so that the expected residence times in states $O_W$ and $O_{UW}$ matches the long and short WiFi connection duration, respectively. Recall that the mean residence time in a given state can be computed as the inverse of the sum of rations of all outgoing transitions [8]. Therefore, the mean residence time in state $O_W$ is $1/\lambda_{W,UW} = 1/\gamma_W^+ = T_W^+$, and the mean residence time in state $O_{UW}$ is $1/(\lambda_{UW,W} + \lambda_{UW,U}) = 1/\gamma_W^- = T_W^-$. The mean residence time in state $O_U$, where no WiFi connection is available, has been empirically determined as $1/\lambda_{U,UW} = 30$ seconds.

*Connection Availability:* The connection availability is the long term fraction of actually available service. Formally, the availability is the steady-state probability that either the UMTS or WiFi NICs are connected (i.e., in state $c_U$ or $c_W$, respectively). Fig. 4(a) shows the availability computed by PRISM as a function of $T_W^-$ and $T_W^+$. (higher is better), for both plain ABPS and ABPS enhanced with the oracle. Plain ABPS provides higher availability, especially when the duration of WiFi connection is short. This can be explained by observing that WiFi is more unreliable than UMTS (the average duration of a WiFi connection is much lower than UMTS), hence the oracle incurs the UMTS connection setup overhead each time the system moves from state $O_W$ to $O_{UW}$. Plain ABPS does not incur this overhead, since both interfaces are always active at the same time, and the setup on one interface may be overlapped with the other one being already connected. The availability improves if the value of $T_W^-$ (the duration of short WiFi connections) is increased. The reason is that longer values of $T_W^-$ give the interface being activated in state $O_{UW}$ enough time to connect before the system shuts down the other one. In practice, this result suggests that the oracle should not trigger a EV_SHORT_WIFI when the WiFi connection is going to be available for a very short time only.

*Power Consumption:* In order to estimate the power consumption, we follow the approach introduced in [2] and define a reward structure in which each state is assigned a reward equal to the power consumption (in Watts) of the NICs in that state. When both NICs are connected, we assume that only the fastest one (WiFi) is used for data transmission. Therefore, in this case we assume that the idle interface consumes 20% of its peak power. Additionally, for ABPS+oracle we increased the reward of all states by $0.1W$, to account for the need to keep the GPS connection active, and for executing the oracle application in background. The contribution of other components of the MN is assumed to be more or less the same in both scenarios (plain ABPS and ABPS+oracle), so it is omitted in this analysis. Rewards are shown in Table I, and are based on data from [11].



(a) Availability (higher is better)

(b) Power Consumption (lower is better)

(c) Throughput (higher is better)

Fig. 4. Model evaluation results for plain ABPS (top) and ABPS+oracle (bottom)

TABLE I
REWARDS FOR ENERGY CONSUMPTION ESTIMATION

| | 0 | $d$ | $s$ | $c$ | $f$ |
|---|---|---|---|---|---|
| UMTS | 0.0 | 0.12 | 0.31 | 0.62 | 0.25 |
| WiFi | 0.0 | 0.08 | 0.19 | 0.38 | 0.15 |

Results are shown in Fig. 4(b): we observe that the oracle significantly reduces the energy requirement of the MN, despite the fact that it uses the GPS device to get the current position. We also observe that availability and power consumption are correlated, as expected: higher availability requires more energy.

*Throughput:* We finally evaluate the throughput provided by the two communication mechanisms. Again, we use a reward structure in which we associate to each state the average throughput provided by the NIC in that state. This means that for we associate a reward of 0.2 Mbps to state $c_U$ of the UMTS chain, and we associate a reward of 26 Mbps to state $c_W$ of the WiFi chain. When both interfaces are connected at the same time, we only use WiFi for actual data transmission. The steady-state reward rate computed by PRISM gives the average system throughput. As can be seen in Fig. 4(c), the oracle guarantees a throughput that is only marginally lower than that provided by plain ABPS. Unsurprisingly, the duration of WiFi connections plays a major role on the system throughput. When WiFi connections last longer, the MN can make better use of that to achieve higher transmission rates. This happens for both plain ABPS and ABPS enhanced with the oracle.

## VI. Conclusion

In this paper we proposed an extension of the ABPS communication mechanism based on a software component (the oracle) which can predict the availability of access points as the MN moves. This information can be exploited to switch off the interface for which no access points are going to be available in the near future, in order to save energy. We described an actual prototype implementation of the oracle based on the WiGLE service; in order to better evaluate the impact of the oracle on ABPS, we defined a performance model based on continuous-time Markov chains that is quite simple and can be efficiently solved. We used the model to compare the connection availability, steady-state power consumption and throughput of plain ABPS and ABPS with oracle, for different scenarios. The results show that the oracle can provide a significant reduction of the power consumption with marginal reduction of availability and throughput. The Markov model is currently being used to drive the full implementation of the oracle-based ABPS, by providing quick performance estimates of various implementation alternatives.

## References

[1] S. Ferretti, V. Ghini, M. Marzolla, and F. Panzieri, “Modeling the always best packet switching mechanism,” in *Proceedings of the 6th International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST2012)*. IEEE, Sep. 2012.

[2] ——, “Modeling the energy consumption of multi-nic communication mechanisms,” in *Proc. IEEE Online Conf. on Green Communications (IEEE GreenCom’12)*. IEEE, Sep. 2012.

[3] V. Ghini, S. Ferretti, and F. Panzieri, “The ”always best packet switching” architecture for SIP-based mobile multimedia services,” *Journal of Systems and Software*, vol. 84, no. 11, pp. 1827–1851, 2011.

[4] W. Wanalertlak, B. Lee, C. Yu, M. Kim, S.-M. Park, and W.-T. Kim, “Scanless fast handoff technique based on global Path-Cache for WLANs,” *The Journal of Supercomputing*, pp. 1–30, 2012.

[5] “Wireless geographic logging engine.” [Online]. Available: http://wigle.net

[6] A. Carroll and G. Heiser, “An analysis of power consumption in a smartphone,” in *Proceedings of the 2010 USENIX annual technical conference*, ser. USENIXATC’10. Berkeley, CA, USA: USENIX Association, 2010, pp. 21–21.

[7] S. Ferretti and V. Ghini, “A web 2.0, location-based architecture for a seamless discovery of points of interests,” in *Fifth Advanced Int. Conf. on Telecommunications, AICT 2009*, 2009, pp. 226–231.

[8] G. Bolch, S. Greiner, H. de Meer, and K. S. Trivedi, *Queueing networks and Markov chains - modeling and performance evaluation with computer science applications; 2nd Edition*. Wiley, 2006.

[9] M. Kwiatkowska, G. Norman, and D. Parker, “PRISM: probabilistic model checking for performance and reliability analysis,” *SIGMETRICS Perform. Eval. Rev.*, vol. 36, no. 4, pp. 40–45, Mar. 2009.

[10] S. Ferretti, V. Ghini, M. Marzolla, and F. Panzieri, “Walking with the oracle: Efficient use of mobile networks through location-awareness,” Preprint arXiv:1209.5243, Sep. 2012. [Online]. Available: http://arxiv.org/abs/1209.5243

[11] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, “Energy consumption in mobile phones: a measurement study and implications for network applications,” in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, ser. IMC ’09. New York, NY, USA: ACM, 2009, pp. 280–293.