Abstract—Multi-homing allows the system to provide a wider range of Internet services with faster, cheaper, and more stable communication services to end-users. Personal Area Networks (PANs) is a typical example of such multi-homed mobile network. It can be formed with multiple devices including PDA, phones, laptops and sensors using a multitude of wireless technologies for PAN to infrastructure connectivity. In this paper, we propose a novel architecture based on a Distributed Virtual Network Interfaces (DVNI) concept. This architecture allows each device of a given PAN to share all network interfaces with other devices and hence enforce PAN to infrastructure connectivity. The choice of the current gateway to be used is based on node’s QoS requirement.

Keywords- Personal Area Network, Multi-homing.

1. Introduction

The Always Best Connected (ABC) [1] is an emerging tendency in the 4G wireless and mobile communication systems. In environments containing multiple wireless (and fixed) networks, user needs to dynamically change access in accordance with its preferences or desired QoS performance, without losing service connection. The ABC solutions have so far mostly been focused on providing multi-access to one terminal that moves between different network contexts. In most cases the user has more than one interface that is capable to connect to the network or has the potential to do so. This may be cellular phone, notebook computer PDAs, cameras, DVD players, mp3 players, game devices etc. A collection of such personal user devices may form personal area networks (PANs).

Different requirements of different applications can result in a different preference of the interface that should be used to the external networks. PAN connections should be placed into the best possible interface based on these requirements. In this context, two concepts have to be considered: PAN-to-infrastructure connectivity and intra-PAN connectivity. Automated solutions should be proposed offering a minimum complexity and maximum transparency to end users. This can be achieved if enough intelligence is available on one or more devices inside the PAN or through the use of third party enforcement such as network or service providers.

In this work, we propose to enhance the Virtual Network Interface (VNI) concept [2] and to adapt it to take into account PAN connectivity requirements. We here propose a distributed variety of this concept that we denote DVNI (Distributed Virtual Network Interfaces). This extension allows considering all the network interfaces of all the PAN gateways as unique pool of network resources. The proposed DVNI concept aims to make available all PAN connectivity resources for all PAN devices.

2. The Virtual Network Interface Design

The Internet routing model forces mobile hosts to acquire a new IP address for an interface when switching to another network. As many Internet-working applications and protocols (including TCP) uses IP addresses as non-volatile identifiers in data structures assuming that they never change, many applications cannot survive an IP address update.

Conceptually, by assigning to all the physical devices on the host the same IP and MAC addresses (as they are in the same subnet: the PAN), the mode of communication during vertical handover will be transparent to higher-layer transport protocols. But, since it is not possible with today implementations, we add a new virtual MAC layer between the hardware level and IP level. We assign a unique unicast IP address to this virtual MAC address, consequently, application might think of any flow as dealing with traffic corresponding to a single interface...
only. The mobile station is always identified in the network by this IP address.

The proposed interface hides the presence of the different network interfaces to the applications layer by providing the illusion of a unique MAC address (virtual one). All application’s flows are sent to this virtual MAC interface that tries to spread and redirect them on the available interfaces. At any time, the virtual interface must manage the changes in network connection’s status. That’s why; it will periodically evaluate interface’s status and use this information to select an available network interface for communication. The virtual MAC groups all information from the interfaces in a specific table that is dynamically updated.

3. Distributed Virtual Network Interface Architecture

1. PAN to infrastructure connectivity

Since a PAN has generally multiple gateways to connect to the global Internet and by consequence multiple egress interfaces; each device in the PAN can be considered as a multi-interfaced terminal when seen from the outside. When a host A (inside the PAN) has multiple paths to send packets to its peers because of the presence of multiple gateways; it must somehow make a decision of which path(s) to use. We call this remote interface selection process. Based on the information provided by the PAN’s egress interfaces discovery protocol, the remote interface selection is done locally by the VNI of A. Things are done as if the remote interfaces were locally connected to the VNI (section 1). The selection algorithm is based on user’s requirements and preferences such as the desired QoS or cost. Finally, packets are forwarded to the selected gateway through a L2 routing mechanism (section 3.2.2). We decide to route packets inside the PAN using a link layer mechanism because some equipment inside the PAN such as sensors- have no IP stack and then can’t understand L3 routing.

2. Intra PAN Connectivity

1. PAN’s egress interfaces discovery mechanism

Gateways periodically announce their presence on the PAN by broadcasting Gateway and Network Association (GNA) messages. A selected set of nodes that received a GNA packet shall forward it to their neighbors. GNA message is in this way flooded onto the entire ad hoc network. At each node in the network, all received GNA packets are registered in the gateway table (figure 1). This information repository is refreshed periodically. This means that entries are deleted as soon as their validity time expires. A GNA message contains triplets of (gateway interface type, cost, and throughput). The Interface type field indicates the technology used by the gateway Egress interface. Example Bluetooth, 802.11, Ethernet, UMTS … The Cost indicates the communication cost associated with the use of the interface described in the interface Type field. This parameter can be a combination of different metrics such as offered QoS, security issues, price and possibly other preferences. Finally, Throughput indicates the throughput associated with the gateway egress interface i.e. nominal data-rate.

Each GNA message will also contain a hop counter to prevent path loops. GNA messages also set up reverses routes to the gateways. These reverses routes allow PAN’s nodes to update their routes to the gateway if the advertisement arrives along a shorter path and to refresh the route entries if the route is already known.

2. L2 routing mechanism

Instead of making ad-hoc routes visible at the IP level; we decide to operate at the link layer. The underlay approach leads to immediate implementation benefits especially for equipments such as sensors as we do not have to interact with the IP layer. Moreover, the discovery mechanisms inside the DVNI permit to add self-configuration elements and routes updates in a very straightforward way.

Routing to the gateway is done on a hop-by-hop basis. As explained before routing tables are constructed and maintained using the PAN’s egress interfaces discovery mechanism (GNA message). Let’s take the scenario in the figure 1, a source node A (with 802.11 and Bluetooth interfaces) wants to send a packet to an Internet node. Using the VNI layer, it’s able to send packets through the 802.11 interface as well as the Bluetooth one without application disruption.

When packets arrive at A’s VNI, it consults the list of active gateways—constructed using the discovery mechanism (GNA messages) among the 802.11 and Bluetooth interfaces. The available PAN egress interfaces are: GSM, UMTS and WIFI. A decides for example to access to the global Internet using the UMTS connection because of some applications’ requirements. It first filters the list of gateways with UMTS egress interfaces. Then, it selects one of them (G1 in figure 1) according to some routes metrics (shorter path, power consumption...).
The local_interface colon (figure 1) indicates that to reach G1, A need to use its local Bluetooth connection. Packets are then sent to the selected gateway, with the destination MAC address field set to that of G1, the destination IP address field set to the CN IP address, the source MAC address equal to the virtual MAC address and the source IP address equal to A’s IP address. Each intermediary node inside the PAN will act as a wireless bridge and forwards packets toward that gateway. Finally, when packets reach G1, it will send them to the CN using normal IP routing protocol.

Packets destined for a node inside the PAN and arriving from nodes on the Internet are tunnelled -using NEMO protocol [3] for example- from the home network to the current gateway, which decapsulates them and forwards them to the PAN node using the here-before described L2 routing protocol.

4. IMPLEMENTATION

We developed a DVNI (Distributed Virtual Network Interface) prototype under linux, hereafter referred as DVNId (d for daemon). Our implementation was tested on the architecture shown in figure 2. The PAN is composed of 4 nodes. To reach a correspondent node (CN) in the Internet, the source node (SN) can pass through GW1 or GW2. When the SN receives GNA message from all PAN’s gateways, it compare between the QoS parameters offers by each GW and then attributes a priority for each GW (the highest to the best one). This priority depends on user preferences and application requirements. In our case, we attribute the highest priority to GW1 among GW2; even if GW2 is closer to the SN in terms of number of hops.

The SN was first connected to the global Internet through GW1 (because it has the higher QoS priority). It started sending UDP packets to the CN. We run different scenarios to evaluate the performance of the proposed DVNI.

Experience 1

In this experience, we focused on the behaviour of our implementation towards changes in PAN’s egress connectivity. To that end, we bring down the wireless interface of the gateway GW1. As it can be seen in figure 3, the total packet number remains relatively stable until the wireless interface of GW1 is down. Once GW1 is down, the total packet number decreases significantly, indicating a decrease in the number of packets successfully transmitted to the CN.

![Fig. 2. Platform of test](image)

![Fig. 3. UDP packet at the CN (Experience 1)](image)
3, after a little time of interruption (100 ms) the traffic resume (throw GW2).

Experience 2

For the second experience, we focused on the behaviour of our implementation upon changes in intra PAN connectivity. After some seconds, the SN get outside the radio range of IN, it must then select a new gateway. It switches to GW2 and continues sending traffic through it. The IP-tables feature of Linux was used to emulate that the SN was not in direct radio-range with IN.

As it can be seen from figure 4, the switching time in this case is more important than in the first experience (for a Hello period of 0.5s, the disruption time is equal to 0,8ms). In fact, in the first experience the gateway GW1 will detect immediately that its egress interface is down and inform the SN about this by sending an empty GNA message. When receiving this message, the SN will delete GW1 from its gateway table and switches to another gateway (according to our platform it will be GW2).

However, in the second experience, GW1 is unaware about the change in the connectivity between the node IN and the SN so it can’t inform the SN about this. To detect that the gateway GW1 is unreachable, the SN will relay on Hello messages.

Each node in the PAN sends periodically Hello messages to indicate that it’s always alive. Since Hello messages are periodic, at \( t = \text{Hello}_\text{period} \) (figure 5), a new Hello message should be received by the SN from GW1. Because the current gateway is unreachable, this will not happen. To be sure that the current gateway (GW1) is no more available, the SN wait for an additional time (\( T_\text{e} \)). If \( T_\text{e} \) expires and no Hello packet is received, then due a long period of silence, the current gateway is deleted from the gateway table and the SN switches to GW2. That’s why in the second experience the disruption is equal to \( (\text{Hello}_\text{period} + T_\text{e}) \).

\[
\text{Hello}_\text{int} = \text{period of Hello messages} \\
\text{Hello}_\text{hold} = \text{validity time of Hello messages} \\
T_\text{i} = \frac{(\text{Hello}_\text{hold}) - \text{t}_1}{2}
\]

Fig. 5. Detection of Hello losses

Compared to other solutions such as [4] [5], where the switching time is more than 1.5s, our solution reduce considerable the disruption time because it operates at the link layer and don’t need to interact with the IP layer which may cause additional delays.

5. CONCLUSION

The biggest challenge to the deployment and acceptance of mobile applications is the need for better connectivity. In this paper a way to provide global and QoS based internet access for Personal Area Networks was described. DVNI requires very few changes to existing protocol stacks. In particular, no changes to non-mobile hosts are needed.

REFERENCES