On Demand Label Switching for Spontaneous Edge Networks

Vincent Untz, Martin Heusse,
Franck Rousseau, Andrzej Duda
LSR-IMAG Laboratory
Grenoble, France

{Vincent.Untz, Martin.Heusse, Franck.Rousseau, Andrzej.Duda}@imag.fr

ABSTRACT

We consider the problem of interconnecting hosts in spontaneous edge networks composed of various types of wired or wireless physical and link layer technologies. All or some hosts in a spontaneous network can be organized as a multihop ad hoc network, connected or not to the global Internet. We argue that this kind of networks requires a more sophisticated approach than standard IP forwarding: communication paths should be managed on a per flow basis, multiple paths need to be maintained to cope with link failures or changing topologies, and the interconnection architecture should provide information on destination reachability.

We have designed and implemented Lilith, a prototype of an interconnection node for spontaneous edge networks. We handle network dynamics by establishing MPLS *(Multi Protocol Label Switching)* label switched paths (LSP) on demand with a reactive ad hoc routing protocol. Interconnection at layer 2.5 makes all the hosts to appear as one single IP subnet so that configuration protocols can use the subnet broadcast for all forms of discovery (addresses, names, services). Performance measurements of the Lilith implementation on Linux show good performance compared with standard IP forwarding and important performance gains when multiple paths are used.

Categories and Subject Descriptors

C.2.2 [**Network Protocols**]: Protocol architecture; C.2.2 [**Network Protocols**]: Routing protocols

General Terms

Design, Algorithms

Keywords

Spontaneous networks, Ad-hoc networks, Autoconfiguration, MPLS

SIGCOMM'04 Workshops, Aug. 30+Sept. 3, 2004, Portland, Oregon, USA.

1. INTRODUCTION

We consider the problem of interconnecting hosts in *spontaneous edge networks* illustrated in Figure 1. With this term we designate networks composed of hosts that use heterogeneous wired or wireless technologies at physical and link layers. A spontaneous network can be connected to the global Internet via one or more border routers or form an isolated group of hosts with internal connectivity. This type of networks becomes increasingly important when deployed at user premises like homes or offices: when connecting different electronic equipment such as sensors and actuators, home appliances, consumer electronics, and various computing devices, spontaneous edge networks make ubiquitous computing become a reality.

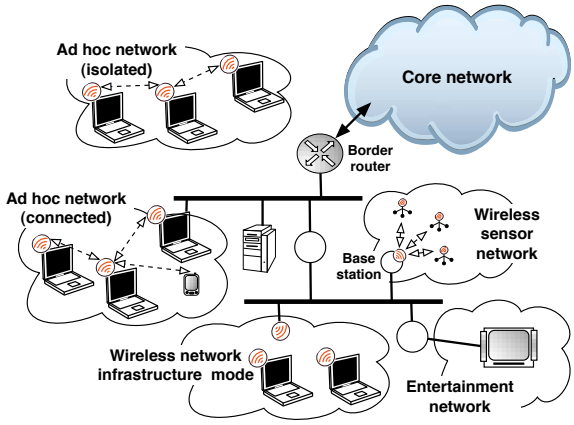

Figure 1: Spontaneous edge network

Our goal is to define an interconnection architecture suitable for spontaneous edge networks. We argue that this kind of networks requires a more sophisticated approach than standard IP forwarding: communication paths should be managed on a per flow basis, multiple paths need to be maintained to cope with link failures or changing topology, and the architecture should provide information on destination reachability. Moreover, the network needs to support TCP/IP applications without configuration or other technical effort from the user. The applications should work regardless of whether the spontaneous network is connected

to the global Internet or it forms a disconnected island, and even if its topology changes due to host mobility or switching network interfaces. At the same time, the proposed interconnection protocol should be lightweight and efficient.

Wireless local area networks become widely spread in such environments and with their increased performance they begin to connect audio/visual home entertainment devices, but only on short distances. To cover larger areas, we can organize nodes into a multi-hop network to provide increased aggregated capacity, greater redundancy, and potential multiple paths. Hence, we assume that all or some hosts in a spontaneous network are organized as a multi-hop ad hoc network.

Another goal is to take into account sensor networks in which small low power devices communicate over short range wireless links by using protocols optimized for low power consumption. Such sensor networks are usually organized in a hierarchical way: sensors communicate with base stations or application gateways having more power and communication bandwidth. A spontaneous edge network can suitably interconnect clusters of sensors, gateways, and appliances as well as provide access to the global Internet.

We also notice that a spontaneous network may carry various types of traffic ranging from low bandwidth sensor information to time sensitive interactive traffic or high bandwidth, low latency home entertainment streams. Hence, the interconnection architecture needs to support some QoS aspects such as traffic classification and prioritization. It is also desirable to take advantage of multiple available paths and adjust to varying traffic conditions and changing topology.

To address all these issues we have designed Lilith, an interconnection architecture for spontaneous networks based on the following principles:

- we propose to interconnect hosts at layer 2.5 by using MPLS *(Multi Protocol Label Switching)* [16] as the forwarding layer,
- Lilith establishes MPLS label switched paths (LSP) on demand based on routes found by a reactive ad hoc routing protocol,
- while the best path is used at a given instant, Lilith searches for other possible paths to use in case of a link failure or a change in topology,
- flows with different QoS requirements may use different LSP paths,
- based on periodic messages with traffic statistics received from all its neighbors, a Lilith node acquires the information on the reachability of established LSPs,
- interconnection at layer 2.5 makes all the hosts connected via heterogeneous links to appear as one single IP subnet so that configuration protocols can use the subnet broadcast (IPv4) or scoped multicast (IPv6) for all forms of discovery (addresses, names, services),
- broadcast or scoped multicast packets propagate to all hosts in the subnetwork via an underlying mechanism of the routing protocol.

As in some previous proposals we place the interconnection architecture at layer 2.5, because if we want to organize all hosts into one subnet, the interconnection should be done below layer 3 and above layer 2 (see more discussion in Section 3). Moreover, we note that to fulfill all the requirements of spontaneous networks, the interconnection architecture calls for more advanced functionalities than those offered by IP, the minimal network layer. In particular, we need more control over paths chosen for transporting data packets, because resources are scarce and we need to use them in an efficient way. As a result we propose to use MPLS, the standard IETF layer 2.5 that allows us to leverage the existing expertise and implementations.

Considering a spontaneous network as a single IP subnet enables all standard autoconfiguration protocols such as DHCP, router discovery in IPv4, IPv6 router and neighbor discovery, service discovery (mDNS, LLMNR, UPnP, SLP, JINI, DNS-SD) to work without modification. When the network is disconnected from the global Internet, schemes such as mDNS or LLMNR, which do not require a central server, can still be used.

To evaluate our approach, we have implemented a prototype of an interconnection node based on the Linux version of MPLS. It relies on a simple reactive ad hoc routing protocol for finding routes in a spontaneous network. In this paper we focus on the motivation of our approach and on the definition of the interconnection architecture. In a companion paper [20], we provide more details on the implementation and performance measurements.

In the rest of this paper, we first discuss the related work (Section 2) and the motivation for an interconnection architecture of spontaneous networks (Section 3), describe the architecture of our proposal (Section 4), and present conclusions (Section 5).

2. RELATED WORK

Perlman has considered large campus networks composed of switches or bridges. To make them appear as a single IP network, she proposed an interconnection architecture based on Routing Bridges [15]. A routing bridge runs the IS-IS routing protocol and encapsulates frames in an additional header if it needs to forward them to another routing bridge. Without explicitly claiming this, the proposal effectively consists of interconnecting routing bridges by means of a layer 2.5 protocol. However, we think that the proposal does not fit spontaneous networks well, because it was designed for static topologies of traditional LANs with the goal of improving existing layer 2 approaches such as the spanning tree protocol. As some parts of a spontaneous network can be organized as ad hoc networks, more adequate routing protocols are needed to cope with the dynamic structure, for instance those proposed by the MANET IETF working group [3, 8]: the main reason for the existence of this group is the search for efficient routing protocols in networks with varying topologies. Moreover, the proposal does not address other issues that raise in spontaneous networks: support for various types of traffic, need for backup routes and for information on destination reachability.

The MANET IETF working group has chosen to provide connectivity to a group of wireless hosts at layer 3: each host acts as a router to forward packets to other hosts. As there is no fixed infrastructure, the address space is flat and addresses are just used as host identifiers. Several schemes have been proposed for providing unique addresses [14, 11, 17]. However, the allocation of private addresses may encounter problems if global Internet connectivity is required.

Some proposals have chosen to provide routing in ad hoc networks at layer 2.5. Lunar (Lightweight Underlay Network Ad hoc Routing) [19] organizes wireless connected hosts into one IP subnet by intercepting and propagating ARP requests to search for a destination host farther away than a local link. The reply of the target node comes back on the reverse route fixed by the request. The routing protocol of Lunar is an on demand distance vector with expanding ring search. Once a route is established, data traffic goes along the route based on a selector added to the packet header between the IP and MAC headers. Selector management and packet forwarding are based on the SelNet (SELector NETwork) library [18].

Ananas [1] abstracts an ad hoc network as a broadcast network using ad hoc virtual addresses assigned to each network interface, while a single virtual broadcast interface per host is assigned an IP address. The latter plays the role of the host identifier used in MANET protocols [3]. Ananas has been designed to run with various routing algorithms, although it requires some adaptation as the routing protocol needs to maintain associations between ad hoc virtual addresses and IP addresses. Ananas was partly designed to solve the layer 3 broadcast issue and presents to the kernel a single virtual network interface that can be viewed as a single access point to the ad hoc network.

In the proposal for the Unmanaged IP, Ford accurately observes that the Internet Protocol is unsuited to routing within and between emerging ad hoc edge networks due to its dependence on hierarchical, administratively assigned addresses [4]. Moreover, he notices that the existing MANET ad hoc routing protocols address the management problem, but do not scale to Internet-wide networks. His goal is to design an edge routing protocol that is self-managing not only on a local scale, but also on a global scale. The proposal introduces an Unmanaged Internet Protocol, a scalable routing protocol that manages itself automatically, based on self-certifying, cryptographic node identities and a routing algorithm adapted from distributed hash tables. We agree with all the reasons that motivated UIP, especially with the crisis at the edge networks, however we do not share the vision in which the global Internet becomes a federation of very large scale ad hoc edge networks without the core, the vision that motivated a new scalable routing protocol. We believe that unmanaged edge networks will be relatively small compared to the managed core network—they correspond to various types of access networks: home networks, SOHO networks (small office, home office), or networks at public places. They will possibly connect to the high capacity core network interconnecting millions of hosts all over the planet. In our opinion, a spontaneous edge network may contain several orders of magnitude less hosts than the global network.

Similarly to the previously cited proposals (Routing Bridge, Ananas, Lunar), we think that the right place for the interconnection architecture is at layer 2.5, because we want to organize all hosts into one IP subnetwork. Below, we briefly recall the characteristics of existing IP underlays that can be used as layer 2.5 for our purpose.

ATM *(Asynchronous Transfert Mode)* presents the vision of virtual connections (virtual paths and virtual circuits) established between hosts. To interconnect hosts and carry IP traffic, the ATM architecture defines LANE (LAN Emulation) in which all hosts maintain permanent virtual paths. In addition to this, hosts need to establish virtual paths to a special broadcast server that propagates LAN wide broadcast packets required by the standard ARP address resolution. The problem of this approach is the number of required virtual paths that grows with the square of the number of hosts. Moreover, LANE has not been designed for handling the dynamics of topology. Another point is that ATM was designed with one physical layer in mind and it can hardly be used for heterogeneous links, even if in the past there were some attempts to introduce ATM into the wireless world.

MPLS *(Multi Protocol Label Switching)* is the standard IETF layer 2.5 for packet forwarding based on label switching [16]. A MPLS network comprises switch routers that forward packets according to labels in MPLS headers. An ingress router classifies an arriving packet to identify its FEC (Forward Equivalence Class), which corresponds to a group of packets that are forwarded in the same manner inside the MPLS network, for example, a FEC may correspond to a destination prefix, a source address, or a QoS traffic class. The ingress router inserts a label into the MPLS header and forwards the packet to the next switch router. The LSP paths followed by packets of a given FEC are created dynamically by establishing entries in the LIB (Label Information Base), the switching table of routers. In the MPLS forwarding paradigm, once a packet is assigned to a FEC, no further header analysis is done by subsequent routers; all forwarding is driven by the labels. Although designed to transport different protocols, MPLS is widely used as an underlay for IP traffic, especially in the internal networks of ISPs. It allows sophisticated performance optimization (traffic engineering) and logical traffic separation—different logical networks can coexist (VPN - Virtual Private Networks). One of the design goals of MPLS was to improve the scalability of ATM LANE: get rid of n-to-n connections between all hosts in the network.

3. MOTIVATION FOR ON DEMAND LABEL SWITCHING

The motivation for our proposal comes from the observation that the current Internet evolves: the traditional fixed infrastructure begins to interconnect edge networks with hosts that can be temporarily disconnected or moving between different points of attachment. The main difference between the core and the edge networks lies in network administration. While the core is mostly managed by network experts, edge networks are used by the end users who do not have sufficient skills nor knowledge, or just do not want to manage their networks like the core. This characteristic of edge networks is also related to wide spreading communication enabled consumer electronics devices and pervasive equipment such as sensors or actuators at homes or offices.

Let us first summarize the characteristics of a spontaneous network:

- it integrates heterogeneous wired (Ethernet, IEEE 1394, USB) or wireless (802.11, 802.15, Bluetooth, ZigBee) physical and link layer technologies,
- it includes some hosts connected via wired links and groups of hosts using wireless links either with or without a fixed infrastructure; in the latter case we assume that communication requires multi-hop forwarding,
- it can provide global Internet connectivity via one or more border routers or stay isolated,

- communication at a given instant may involve only a small subset of hosts, some of them performing special functions (servers, gateways, or sensor base stations),
- network resources, especially in the wireless part of the network, may be limited and should be used efficiently,
- network topology may vary in time due to host or interconnection node mobility,
- it may carry various types of traffic with different QoS requirements,
- its configuration should be done automatically.

Different forms of spontaneous networks may exist ranging from a pure ad hoc wireless network to a set of traditional wired LANs, both connected or not to the global Internet. The overall goal of our proposal is to define an interconnection architecture for such spontaneous edge networks that takes into account QoS aspects and simplifies network administration.

3.1 Need for an interconnection architecture

Consider first the problem of interconnecting heterogeneous links within a spontaneous network. Any interconnection architecture can work, however we will try to identify the most suitable one and to provide some arguments for it.

Obviously, the interconnection can be done at layer 2 with STP (Spanning Tree Protocol) bridges. However, this solution presents several drawbacks as noted in the Routing Bridge proposal [15]: routing via the spanning tree concentrates traffic onto selected links, is slow to bring new connectivity, and nodes must have a distinct layer 2 address for each point of attachments. Moreover, forwarding a frame towards a destination in an ad hoc wireless network often requires sending it on the same interface on which the frame has been received. This is not possible when intermediate nodes act as bridges. The Routing Bridge proposal for interconnecting different links at layer 2.5 overcomes these problems, but it is not suitable for the ad hoc parts of a spontaneous network as analyzed in Section 2.

Heterogeneous link layer technologies can be interconnected at IP layer, the solution that requires a router between each link. In the case of the ad hoc part of the network, the MANET routing protocols [8] can provide connectivity to a group of wireless hosts at layer 3. A large number of protocols has already been proposed, the most familiar being AODV [10], DSR [12], and OLSR [2]. AOMDV (Ad hoc On-demand Multipath Distance Vector) protocol, an enhancement of AODV that provides multiple routes, is the most relevant ad hoc routing protocol related to our work [10]. As mentioned before, the address space in ad hoc routing protocols is flat and addresses are just used as host identifiers. Connecting such a spontaneous network to the global Internet can be done by establishing a default route to the border router. However, the reachability of hosts from the Internet requires either address translation at the border or the advertisement of all prefixes existing in the network.

A spontaneous network can be organized as a collection of IPv6 routers. The v6ops IETF working group has setup a design team to describe various scenarios and analyze how IPv6 can be introduced in unmanaged networks [5, 6]. However, we observe that the addressing scheme in IPv6 also speaks in favor of providing the view of a single subnetwork. Theoretically, an IPv6 network can obtain a short prefix (between 48 and 64 bits) that can be shared within a spontaneous network. This requires a prefix distribution protocol of type Zerouter [9]. Home networks are likely to get a single 64 bits prefix which precludes assigning disjoint longer prefixes to each link and restricts the spontaneous network to use host routes.

We can see that the possible choices for an interconnection architecture discussed above present several drawbacks for spontaneous edge networks. We think that placing the interconnection architecture at layer 2.5 is a suitable approach for such networks, because we can easily address the requirement of autoconfiguration.

3.2 Need for a reactive route discovery

We observe that the space and time locality of communications are arguments for a reactive route discovery. Even if a spontaneous edge network may be composed of many hosts, communication at a given instant will only involve a small subset of them, because some hosts will be specialized to provide some well defined function (storage, format transcoding, measurement gathering). This speaks in favor of a reactive route discovery, because it is not necessary to maintain all possible routes between all nodes. Note that in a similar way ARP maintains dynamic address mapping in IP subnetworks.

Moreover, in a network with limited network resources it is important to reduce the overhead of route discovery, for example by adopting a lazy approach in which routes are searched for if there is some traffic to transport. A reactive routing protocol operates in this way, so that it is not necessary to maintain all possible routes all the time.

3.3 Need for a connection oriented approach

Spontaneous networks may have varying topology, limited network resources, and carry various types of traffic with different QoS requirements. We think that a connection oriented approach provides more control over the paths used for communication and complements the reactive aspect of route discovery with some proactive characteristics—in a volatile and dynamically changing environment, we prefer to reason in terms of an end-to-end connection than to leave to each intermediate node the role of deciding what to do with each packet. Once a route is discovered using a reactive protocol as pledged for above, there is a need for maintaining, testing, and optimizing existing paths, for example finding alternate paths that can be immediately used as a back up after a link failure or testing the reachability of established connections to detect a link failure. Such a hybrid approach allows elimination of broadcast storms if there is an existing alternate path when a link fails.

Moreover, a connection oriented approach enables us to address some QoS aspects in an easier way. Since we aim to place the interconnection architecture at layer 2.5, we have chosen to use MPLS, the standard IETF layer 2.5 for forwarding unicast packets. At the beginning, MPLS was developed for increasing forwarding performance by adopting switching techniques similar to ATM. With the development of recent high performance routers, this performance objective has disappeared and MPLS stays attractive mainly for traffic engineering in the network core. However, we observe that traffic engineering is also important for edge networks,

even to a larger extent, and especially for wireless edge networks in which resources may be limited. The price for using MPLS is a slight overhead, because each packet is encapsulated in a MPLS frame with a four byte header, but we can benefit from all other features of MPLS. Building upon the existing experience on traffic engineering in MPLS networks, we can easily provide support for traffic isolation, service differentiation, and even separating some parts of a spontaneous networks as different logical VPNs.

3.4 Need for handling QoS aspects

As a spontaneous network may carry various types of traffic the interconnection architecture needs to handle some QoS aspects. Obviously, we cannot guarantee any QoS parameters if for example radio transmission conditions are bad, hosts move out of the radio coverage, or some links are broken. However, we believe that between such situations and perfect transmission conditions, it is possible to benefit from fairly good conditions, so that there is a place for providing some QoS support, even in the presence of moving nodes and varying radio conditions.

For example, our goal is to achieve the following: when a communication session starts (e.g. a VoIP call is established, a video stream goes to a remote display, a sensor base gathers information from sensors), the network dynamically establishes an appropriate route to provide the best quality for the flow and to isolate it from possible influence of other flows. Once established, a route need to be maintained in spite of changing topology or changing conditions of a wireless transmission: the network switches to other routes. In addition to that, the wireless part of the network requires the possibility of explicitly probing links with neighbor nodes as pointed out previously [7]. A connection oriented approach coupled with an on demand reactive routing protocol seems to be the most suitable framework to satisfy all the requirements. By careful traffic engineering, flows can be protected and their QoS differentiated, for example different DiffServ classes can be sent over different LSP paths taking disjoint routes.

3.5 Need for autoconfiguration

From the autoconfiguration point of view, interconnection at layer 3 presents some drawbacks: standard IPv4 configuration protocols such as DHCP rely on the IP broadcast (255.255.255.255) limited to a single subnetwork. If several subnets exist in a spontaneous network, DHCP relays should be placed on each link. A spontaneous network organized as a collection of IPv6 routers can use the IPv6 scoped multicast for autoconfiguration. However, it also requires configuration of the multicast distribution tree via an appropriate protocol such as PIM-SM, which is complex.

So we think that autoconfiguration can be made easier if we organize a spontaneous network as a virtual IP subnetwork in which a packet sent to the broadcast (or a link-scoped IPv6 multicast) address propagates to all hosts. In this way configuration protocols can use this functionality for all forms of discovery (addresses, names, services). If we want to organize all hosts into one IP subnetwork, the right level to provide connectivity is layer 2.5 similarly to Lunar and Ananas. In particular, when the network is isolated from the global Internet, hosts can acquire addresses based on Auto IPv4 or stateless IPv6 configuration and use mDNS or LLMNR for name resolution. When connected to the border router, they may benefit from a DHCP service to learn the routable prefix. Moreover, protocols such as UPnP, SLP, JINI, or DNS-SD can readily be used for service discovery.

4. ARCHITECTURE OF LILITH

We propose an interconnection architecture that follows a connection oriented approach—it is based on the label switched MPLS forwarding coupled with a reactive ad hoc routing protocol. The idea is to organize a spontaneous network as a single IP subnetwork (broadcast or scoped multicast packets propagate to all hosts), forward packets over heterogeneous links with MPLS, and establish LSP paths on demand with a reactive ad hoc routing protocol. MPLS, the standard layer 2.5 protocol, glued together with a reactive ad hoc routing protocol results in several interesting features for spontaneous networking.

To experiment with our approach, we have designed and implemented Lilith, a prototype of an interconnection node for spontaneous edge networks.

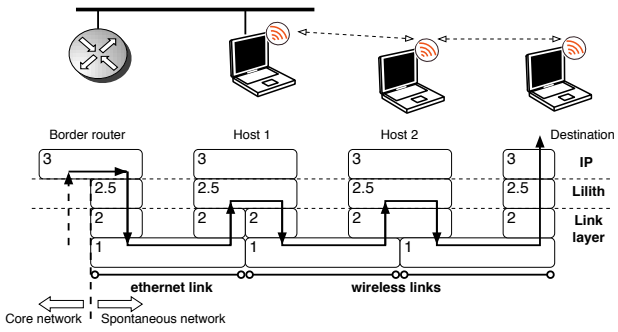

Figure 2: Protocol stack and packet path

Lilith forms with MPLS a 2.5 layer that presents to IP the abstraction of a single broadcast (scoped multicast in IPv6) link layer network. In this architecture, routes and data paths are dissociated so that different protocols can be used for finding a route and establishing a data path. A route is a list of intermediate nodes between a source and a destination. It is not used for sending packets, but serves to establish a path between the source and the destination. Figure 2 presents the structure of the protocol stack at hosts and intermediate nodes as well as the data path taken by a packet.

The functional architecture of Lilith is presented in Figure 3. It relies on an ad hoc routing protocol, which in the current implementation is based on a simple reactive routing mechanism similar to AODV [13] or AOMDV [10]. When describing the elements of Lilith below, we use our routing protocol as an example, however it can be replaced by any other reactive ad hoc routing protocol such as DSR [12].

Lilith is composed of the following main elements:

- Routing,
- Path Instantiation,
- Path Maintenance.

We describe the details of the architecture below.

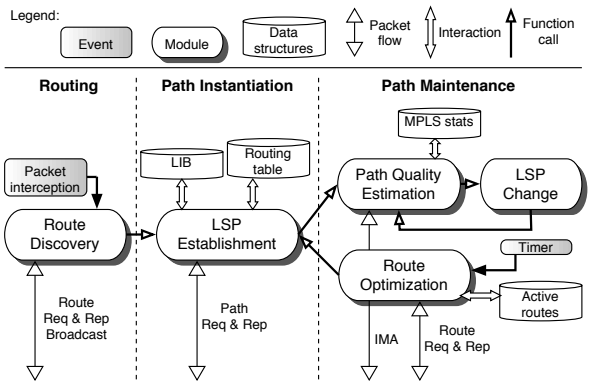

Figure 3: Functional architecture of Lilith

4.1 Routing

Packet Interception. The module intercepts unicast packets for which no LSP exists and all broadcast packets. The unicast packets are passed to the Route Discovery module that buffers them and starts the search for a route, which in turn triggers the establishment of a LSP. The buffered packets are sent by the MPLS layer once a LSP path is established for the destination. Broadcast packets are passed to the routing protocol that uses the underlying route discovery mechanism (flooding) to propagate them to all hosts in the subnetwork.

Route Discovery. The module implements the routing protocol, a reactive ad hoc protocol similar to AODV though less complex. Its purpose is to provide at least one path for a route, so we can establish a LSP to the destination.

The protocol makes use of two messages: ROUTE REQUEST and ROUTE REPLY. When a route is needed, it generates a ROUTE REQUEST message containing a record of visited nodes. The request is flooded across the subnetwork and each intermediate node adds itself to the record. Once the request reaches the destination host, it generates a ROUTE REPLY that comes back on the reverse path. The destination host can reply to more than one request and thus initiate the construction of more than one LSP so that the requesting host may choose among several possible paths.

Handling broadcasts. The protocol uses BROADCAST message to propagate a broadcast or a scoped multicast packet to all nodes of the subnetwork. Our current implementation uses flooding for this purpose.

4.2 Path Instantiation

LSP Establishment. This module uses a LSP establishment protocol based on two messages: PATH REQUEST and PATH REPLY. PATH REQUEST propagates to the destination that replies with PATH REPLY carrying the label bindings for LSP establishment. At each node on the reverse path the LSP Establishment module modifies LIB, the table used by MPLS for forwarding packets based on label switching. The LIB modification actually creates a LSP path at a given intermediate node or a host. In the current implementation the LSP establishment protocol is closely coupled with our reactive routing protocol—PATH REQUEST is included in ROUTE REQUEST message and PATH REPLY in ROUTE REPLY. In this way, we do not introduce an additional overhead compared to any other ad hoc routing protocol.

4.3 Path Maintenance

Path Quality Estimation. This module acknowledges traffic received over established LSPs by periodically sending I'M ALIVE (IMA) message to all immediate neighbors using a link layer broadcast (note that the message is limited to one hop). It carries a list of labels and the amount of data received on each active LSP during the last period of time. This information can then be used to decide if a given link is broken, in which case the LSP Change module switches to another path. If it is not the case, Lilith uses the information to estimate link quality, the metric that can be injected into the routing protocol.

If there is no traffic received for an outbound label during an interval, the corresponding entry in the LIB is deleted. Lilith detects existing long-lived TCP connections that do not send data for long periods of time and sends probe MPLS frames over the corresponding LSP to keep them active.

LSP Optimization module. It tries to find alternate routes for existing active destinations using the Route Discovery module. This search is done in the background and does not interfere with the data flow over existing LSPs. The module evaluates the quality of alternate routes and, if necessary, establishes paths using the LSP Establishment module. A newly established LSP can either be used as a backup or become the primary LSP if it is significantly better.

4.4 Discussion

In general, an interconnection architecture needs to provide the following features: topology discovery, route or path establishment, and forwarding. In Lilith, the first two features are merged, because it is straightforward with a reactive routing protocol whereas MPLS takes care of forwarding. Lilith adds a fourth feature: it periodically checks if established paths are operational. Usually, this task is implicitly done by the routing protocol that recomputes routes when it discovers a change in topology. But in Lilith, the reactive routing protocol is explicitly backed up by proactive path maintenance, triggered by periodic path probing.

All routing protocols require some kind of information propagation to construct routes. For instance, RIP aggregates distance vector advertisements and propagates them to neighbors, OSPF floods link state advertisements all over the network, OLSR optimizes flooding by carefully choosing routers (MPR - Multi-Point Relays) that actually propagate advertisements. Such information propagation corresponds to a network-wide broadcast: route discovery is a special kind of service discovery. We think that there should be a single means of forwarding both route discovery requests and broadcast packets generated by upper layers. This is the case of Lilith which uses the same mechanism to propagate broadcasts and route discovery requests.

Whenever spontaneous or ad hoc networks and MPLS are

mentioned together, they give rise to both confusion and surprise. Usually MPLS is considered as a connection oriented reminiscent of ATM, perhaps not particularly well suited for dynamic topologies such as spontaneous networks. However, we believe that when coupled with on demand establishment of LSP paths driven by a reactive ad hoc routing protocol, MPLS presents several advantages. First, transient loops can be easily avoided, because LSP paths are explicitly created between end point hosts. Such a property is particularly interesting in the context of wireless networks with limited resources. Second, we can provide multiple paths for load balancing or traffic isolation for different QoS classes, including per flow, or per source or destination address. Another advantage is the possibility of acquiring the information on the reachability of established LSPs: an interconnection node periodically sends a message with statistics of traffic received from all its neighbors.

A Lilith interconnection node does not require all of the functionalities provided by a fully-fledged MPLS router. As IP packets exchanged between hosts are encapsulated in MPLS frames with four byte shim headers, we only require that the network supports frame forwarding along LSPs according to the LIB. We do not rely on any other MPLS related functionality, because the Lilith layer takes care of label distribution. Neither does it make use of label stacking.

Lilith path quality estimation relies on measurements of traffic received over each LSP. This is a system-dependent feature that may not exist on some MPLS platforms, however some form of traffic metering is usually available.

Using on demand label switching is a first step towards handling QoS aspects. Route discovery in Lilith can take into account some QoS parameters of LSP paths, for instance preferring routes that use more wired hops or feature wireless links with better quality. As the ultimate goal of our architecture is to provide fine grained quality of service for heterogeneous flows (e.g. video stream, Web surfing traffic), we would probably require some sophisticated scheduling of MPLS frames belonging to different LSPs, for instance give a higher priority or a fixed bandwidth share to time-sensitive LSPs if they compete with other flows on the same link. Nevertheless, even if we do not rely on such mechanisms, which is the current state of our prototype, Lilith provides some valuable features: we can easily cope with changing conditions by detecting better paths, we can provide flow isolation based on injecting different flows into LSPs created on different paths in the network (the routing protocol can choose routes based on metrics that take into account link capacity) or load balancing according to the load measured on a link.

4.5 Performance

We have implemented Lilith as a user space daemon using the Linux version of MPLS for forwarding. The implementation of the packet interception module uses `netfilter` with `libipq`. We provide more details on the implementation and performance measurements elsewhere [20] and we summarize below some performance results coming from measuring an experimental network of Lilith nodes.

When experimenting with path establishment, we have observed an important overhead for the first packet to a given destination, but then the difference in the round trip time is fairly small. The throughput measured on a LSP path across three 100 Mb/s Ethernet links is only slightly decreased (0.3%) compared with standard IP forwarding.

We have observed important performance gains when traffic goes over multiple routes: we generate greedy TCP traffic on one path and measure the round trip time perceived by a time-sensitive flow on another path. Our label switched architecture makes it possible to use a second route to communicate with a given destination, which is not possible in the standard IP forwarding.

We have also tested the dynamic behavior of Lilith in a setup with several possible routes to see how Lilith reacts to link failures and performs route optimization if necessary. When Lilith detects a link failure after three missing IMA messages, it switches immediately to a backup path prepared in the background. If the primary route becomes operational again, the optimization module reroutes the traffic back to the first path.

5. CONCLUSION

Interconnecting various computing devices, consumer electronics, home appliances, and electronic equipment such as sensors and actuators naturally results in a spontaneous edge network. Usually, because the communication and computing resources of such networks are limited, the interconnection architecture requires careful design.

We have presented an original approach for interconnecting nodes in a spontaneous network based on dynamic label switching. Our goal is to allow different communication paths on a per flow basis, provide seamless switching between operational and back-up paths, and make available information on destination reachability. By interconnecting all links at layer 2.5 so that they appear as one single IP subnet, we are also able to get rid of administration burden—standard autoconfiguration protocols such as DHCP, router discovery in IPv4, IPv6 router and neighbor discovery, service discovery (mDNS, DNS-SD, LLMNR, UPnP, SLP, JINI) can run as if they were on a single LAN.

Using on demand label switching is a first step towards handling the network dynamics: not only the topology of a spontaneous network may change over time, but also the network may experience important load variations. Our approach can cope with these changes by detecting broken or better paths.

One step further will be to take into account some QoS parameters in route discovery, for instance give more preference to the routes that use more wired hops or feature wireless links with better quality. We can then explore different types of QoS based routing, because all possibilities are being opened by the connection oriented nature of Lilith. For instance, we plan to investigate flow-based load balancing and QoS scheduling of MPLS frames belonging to different LSPs.

Our first prototype shows fairly good performance compared with traditional IP forwarding. We observe only a slight degradation in performance when using Lilith, however we benefit from all features provided by MPLS. We still work on more performance experiments in a real network set up to gain more insight into the behavior of our protocols.

In the current implementation, broadcast relies on simple flooding. We want to investigate other approaches for supporting broadcast.

6. REFERENCES

[1] G. Chelius and E. Fleury. Ananas : An Ad hoc Network Architectural Scheme. In *Proc. MWCN*. IEEE, September 2002.

[2] T. H. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR). RFC 3626, 2003.

[3] S. Corson and J. Macker. Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. RFC 2501, 1999.

[4] B. Ford. Unmanaged Internet Protocol: Taming the Edge Network Management Crisis. In *Proc. HotNets-II*, pages 476–483, Boston, 2003.

[5] C. Huitema et al. Unmanaged Networks IPv6 Transition Scenarios. Internet draft, October 2003.

[6] C. Huitema et al. Evaluation of IPv6 Transition Mechanisms for Unmanaged Networks. Internet draft, February 2004.

[7] H. Lundgren, E. Nordstrom, and C. Tschudin. Coping with Communication Gray Zones in IEEE 802.11b based Ad Hoc Networks. In *Proc. 5th ACM International Workshop on Wireless Mobile Multimedia*, 2002.

[8] IETF MANET Working Group, 2000. `http://www.ietf.org/html.charters/-manet-charter.html`.

[9] O. Marce and D. Galand. Architecture for Zerouter. Internet draft, January 2003.

[10] M. K. Marina and S. R. Das. On-demand Multipath Distance Vector Routing in Ad Hoc Networks. In *Proc. IEEE ICNP*, 2001.

[11] S. Nesargi and R. Prakash. MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network. In *Proc. INFOCOM'02*, 2002.

[12] C. E. Perkins, editor. *Ad Hoc Networking*, chapter DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks, pages 139–172. Addison-Wesley, 2001.

[13] C. E. Perkins, E. M. Belding-Royer, and S. R. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561, 2003.

[14] Perkins, C. et al. IP address autoconfiguration for ad hoc networks. Internet draft, November 2001.

[15] R. Perlman. Rbridges: Transparent Routing. In *Proc. Infocom 2004*, Hong Kong, March 2004.

[16] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. RFC 3031, 2001.

[17] S. Toner and D. O'Mahony. Self-organising node address management in ad-hoc networks. In *Proc. PWC, Lecture Notes in Computer Science 2775*, 2003.

[18] C. Tschudin and R. Gold. Network Pointers. In *Proc. HotNets-I*, 2002.

[19] C. Tschudin, R. Gold, O. Rensfelt, and O. Wibling. LUNAR: a Lightweight Underlay Network Ad-hoc Routing Protocol and Implementation. In *Proc. Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN'04)*, 2004.

[20] V. Untz, M. Heusse, F. Rousseau, and A. Duda. Lilith: an Interconnection Architecture Based on Label Switching for Spontaneous Edge Networks. In *Proc. Mobiquitous 2004*, Boston, August 2004.