Improved Community Network Node Design using a DLEP based Radio-to-Router Interface

Christoph Barz and Henning Rogge
Fraunhofer Institute for Communication, Information Processing and Ergonomics FKIE
Neuenahrer Str.20
53343 Wachtberg
Germany
Email: {christoph.barz, henning.rogge}@fkie.fraunhofer.de

Abstract—Using a static link state routing metric in wireless mesh and mobile ad hoc networks without major efforts in optimizing the physical link properties has proven to be inefficient. It does not take into account link quality information. While some link metrics try to estimate channel properties based on layer-3 observations, more sophisticated metrics need to access layer-2 information directly. The new 802.11 netlink interface provides a common local interface for Linux based systems and thus the basis for practical usage in community networks. However, a standardized radio-to-router communication protocol to access layer-2 information will allow to go beyond the limitations of having all radio interfaces built directly into the router device. While the PPPoE protocol generally can be applied to this scenario, it has some drawbacks when used in wireless mesh networks. The new DLEP protocol is supported by the IETF MANET group to fill this gap. In this paper we describe the advantages of a flexible node design for community networks based on the DLEP protocol. In addition, we present our implementation of the DLEP protocol and discuss some important deviations from the current draft version of the standard.

I. INTRODUCTION

The availability of inexpensive, easy to deploy Wi-Fi equipment has paved the way for Community Networks as a model for Future Internet access, providing open wireless connectivity. Typical sites in community networks consist of several (outdoor) routers, each equipped with one or sometimes a few Wi-Fi cards. Unfortunately, this design leads to large encasements for the routers (weather sensitive), long antenna cables (link quality reduction), and multiple layer-3 devices per site (problematic scalability and configuration).

We suggest the use of small remotely attached, embedded radio devices providing layer-2 services. These radio devices are connected via LAN to a single router (layer-3) which can even be placed indoors. This may lead to a more scalable and robust design for the site. However, the access to link layer quality information for the router is more difficult as raw access to the wireless interfaces is no longer available. This information is important in situations where “weak” links have to be avoided by the routing process. Here, the use of radio-to-router communication protocols as a standardized way to allow for remote access to link quality data may fill the gap.

Section II provides an overview of radio to router communication protocols for link layer information exchange. In addition, routing metrics for wireless networks that can benefit from this information are presented. Section III identifies problems with the radio-to-router protocols introduced. Section IV gives an overview of the current best practice regarding the node architecture in Community Networks. Based on this, section V shows how a modified DLEP implementation can enhance the community network node design. Section VI presents details of our radio-to-router protocol implementation and section VII concludes this paper.

II. RELATED WORK

Radio-to-router communication protocols provide the basis for using link layer quality information for remote radio devices at routers.

A. Radio to Router Communication Protocols

Currently, there are two basic approaches for collecting this link layer information from a remote radio device.

The first is based on extensions to the well known PPP protocol. While being widely available, special means have to be taken when not used in a point-to-point fashion. A full mesh of PPP tunnels is required between all radios that are able to communicate directly. In addition, means have to be taken to allow for using the PPP interfaces in this radio context.

The second approach is still in its definition phase. The DLEP protocol is explicitly designed for the use of radios in a dynamic routing environment and does not involve tunneling.

The following subsections provide an overview of the protocol developments for both approaches.

1) Point-to-Point Protocol (PPP): The roots of the Point-to-Point Protocol go back in time as far as 1989 where its first standard RFC 1134 [1] was adopted to provide a method for transmitting datagrams over serial point-to-point links. Since then, the standard got revised and extended multiple times. The current version of the basic protocol is described in RFC 1661 [2]. Besides an encapsulation methodology for multi-protocol data, it also comprises a stateful Link Control Protocol (LCP) component in order to monitor and manipulate the status of the link. It also includes an authentication process and an echo-based probing mechanism. In addition, PPP supports a family of network layer control protocols for different network layer protocols, including IPCP for IP address exchange. Here, the IP address of the PPP session peer is inserted into the local routing table.
2) **PPP over Ethernet (PPPoE):** PPPoE is an encapsulation protocol to transport PPP frames over Ethernet links. It got popular with the advent of the Digital Subscriber Line (DSL) technology. Host PC or DSL router can establish a point-to-point connection to a remote access server via a DSL modem and a DSL access multiplexer (DSLAM) either directly by encapsulation (RFC 2684 [3]) or via stitching at the DSL Modem (RFC 2364 [4]). In the later case, PPPoE is terminated at the local modem which then uses PPP over ATM to transport the PPP data to the remote access server. Both configurations allow for a separation of the Host or router from the DSL Modem via an Ethernet connection. Unlike PPP itself, PPPoE introduces a client server relationship and a discovery mechanism between the host and the service providers remote access server.

3) **PPP over Ethernet (PPPoE) Extensions for Credit Flow and Link Metrics:** While PPPoE was designed to work with remote modems connected via Ethernet in environments with a relatively stable link between the modems, RFC 5578 [5] focuses on remote radio devices with highly variable link characteristics (e.g. link capacity) and limited buffering capabilities. The extended PPPoE is used between the router which acts as a PPPoE server and a connected radio which acts as a PPPoE client. This allows for connecting several radio devices to a single router. As depicted in figure 1, link quality information like link latency, current and maximum data rates, relative link quality, and neighbor up/down status is exchanged via PPPoE Active Discovery Quality (PADQ) packets. They can be used for both querying the data as well as transmitting the actual information. The credit based flow control mechanism extends the discovery mechanism by introducing two new message types and extending the existing ones. The PPPoE Active Discovery Session-Grant (PADG) and PPPoE Active Discovery Session-Credit Response (PADC) messages are used to grant incremental flow control credits between the router and the radio device.

4) **Virtual Multipoint Interfaces based on PPPoE:** In [6], Veyster and Cheng presented a Common Virtual Multipoint Interface (CVMI) as an open source implementation of the Multi Point Interface (MPI) [7] used by Cisco routers to allow for the use of PPPoE including link quality information in mobile ad hoc and mesh networks. A full mesh of PPP tunnels is required between all potential link neighbors. For details regarding the drawbacks of the application of PPPoE in a MANET and mesh radio scenario please refer to section III.

5) **Modem Link Properties Advertisement Protocol:** The Modem Link Properties Advertisement Protocol [8] is meant as a simple alternative to PPPoE for a modem to advertise link characteristics via UDP messages. The current status of the protocol is being an early non-working group IETF draft (version 01) with experimental status. The authors claim to be in contact with the DLEP protocol developers in order to evaluate if their scenario targeting modems and crypto devices is a special case of the scenario targeted by DLEP. DLEP is described in the following section.

6) **Generalized Mobile Ad Hoc Network (MANET) Packet / Message Format:** Generalized Mobile Ad Hoc Network (MANET) Packet/Message Format was standardized as RFC 5444 [9] in February 2009. RFC5444 describes a generic binary format designed to work well for most MANET routing protocols. The format consists of a packet container with an optional sequence number, which contains a sequence of messages. Each of these messages has a type, an address length definition (up to 16 bytes), up to four optional fields (sequence number, hopcount, hoplimit and originator address) and a list of addresses. Each of these three hierarchy levels (packet, message, address) can have a variable number of TLVs (Type-Length-Value) fields. The TLV structure allows a reader to skip over any type of information it does not understand without disrupting the parsing process. This provides an easy way to add additional data to new implementations of protocols without breaking compatibility. There is also a growing list of RFC 5444 related RFCs and drafts, for example for encoding time intervals [10] and cryptographic signatures on packet, message and address level [11].

7) **Dynamic Link Exchange Protocol:** The Dynamic Link Exchange Protocol (DLEP) [12] focuses explicitly on wireless mesh networks allowing for link quality information exchange and flow control between a router and a connected radio device. The concept separates the information channel from the data plane between the router and the radio. As depicted in figure 2, no full mesh of tunnels is necessary between the involved radio devices. DLEP communication is only happening between the router and its connected radio. Thus, it offers similar functionality as CVMI/PPPoE while trying to avoid its inherent complexity in the wireless mesh network scenario. As a MANET WG draft, DLEP targets to use the
Generalized Mobile Ad Hoc Network Packet Message Format (packetbb) [9]. DLEP uses RFC 5444 to provide an automatic discovery between radio and router. When this discovery phase is complete, the radio can deliver the router metrics and other data about the attached network and the link to its current neighbors. DLEP contains two extensions, which allow to request a change of the characteristics of a link to a neighbor and a credit based flow control scheme similar to [5].

8) Radio-to-Router Interface Framework and Requirements: Independent from the work on DLEP, [13] tries to define the scope and to analyze the general requirements of a standardized radio-to-router interface to support routing decisions based on link layer information from heterogeneous radio links. While the DLEP protocol fulfills many of the requirements (PPPoE does not), the Framework and Requirements draft has not been adopted as a Working Group, yet. However, it should be taken into account for future draft versions of DLEP.

B. Dynamic Routing Metrics

Most routing protocols provide the concept of “link cost”, which gives the protocol a way to prioritize certain links over others. This is especially important in wireless networks, where the quality of a link might constantly shift because of environmental changes or movement of nodes.

While some link metrics can be calculated directly from OSI layer-3 data, more advanced link metrics need access to layer-2 data, including transmission speed, signal strength or similar values. Both PPPoE (with the necessary extension) and DLEP can provide this data to improve routing metrics for the attached routers.

1) Static Metric: Many routing protocols for wired networks use static metrics, where every link has a constant cost. While early Mobile Ad hoc Network documents like Optimized Link State Routing (OLSR) [14] suggested to use the hopcount metric (a simple static metric) on wireless networks, experimentation has shown that most wireless networks contain “weak” links which have low transmission speed and high packet loss and that the hop count metric concentrates the traffic on these weak (long!) links. Because of this, static metrics are nearly unused in Community Networks.

2) ETX Metric: The ETX metric [15] is used in several OLSR-based community networks (see section IV) like Funkfeuer [16] and Freifunk [17]. The ETX metric of a link is the estimated number of transmissions required to successfully send a packet over that link, until an acknowledgment is received.

3) ETT Metric: The ETT metric [18] was proposed as an evolution of the ETX metric by combining the loss rate of a link with the transmission rate. By doing this, the metric calculates the average time the link will be occupied by the transmission. This should lead to a reduced usage of the available electromagnetic spectrum, which increases the capacity of the whole network.

Unfortunately, ETT is difficult to calculate without access to the link-layer. Only in the last few years Linux, which is the common operating system in Community Networks, got a generic IEEE802.11 MAC layer with good driver support, which provides a generic way to acquire the necessary data. Both PPPoE with the link metric extension and DLEP could be useful to implement a metric like ETT.

III. Problems regarding current IETF radio-to-router protocols

There are two major protocols in the IETF for radio-to-router communication, the PPPoE link metric extension and the DLEP draft in the MANET-WG. Unfortunately both of them have some disadvantages.

A. Problems with PPPoE Link Metric Extension

There are several general pitfalls when using PPP interfaces in an environment with dynamic routing and radio connections. PPP interfaces are set up and torn down with the availability of the links. In addition, PPP requires a full mesh of tunnels between all radios on a common channel, thus leading to large configuration and communication overheads. Multicast and broadcast packets have to be replicated for every PPP session, regardless of the broadcast capability of the radios.

While the integration of PPPoE into a routing framework, the replication of broadcast traffic for the different PPP sessions, and the configuration complexity of the different sessions are addressed by CVMI, the approach is still a workaround to hide the point to point nature of the PPP protocol. In addition, PPPoE tunnels are always connected end-to-end, which means that using PPPoE for getting link metric data from the radio always produces additional complexity and data on the wireless connection. This might be compensated by the built-in compression of PPPoE, but this depends highly on the traffic transmitted.

Another minor problem of the PPPoE extension is the encoding of the link metrics. It contains a value between 0 and 100 as the relative link quality, which is defined to be 100 (or perfect) if it cannot be measured. This problem is shared by the current DLEP draft.

B. Problems with current DLEP-02 draft

The current DLEP draft, although especially designed for the usage in mesh networks and MANETS, has its own problems. DLEP only describes communication between radio and router. Thus, the protocol completely evades the complexity and overhead of the end-to-end tunnels used in PPPoE. This allows the use of DLEP even on low-bandwidth networks. DLEP is better suited for mesh networks and MANETs and fulfills many of the requirements of [13].

The first current problem with DLEP is that it defines its own packet format inside a RFC5444 TLV, called sub-tlvs. This means that standard reader and writer implementations for RFC5444 must be extended with additional code to parse and generate DLEP packets. In our opinion, there is no reason...
why the same information could not be transported without the need of a “proprietary” subtlv-extension.

The second major problem with DLEP is that it defines a complex state machine between router and radio which switches between different modes during discovery and neighbor data transmission. We think that this state machine should be completely replaced by a validity time-based stateless approach, similar to the one used in many MANET routing protocols. This will also allow for other use cases for DLEP, as described in section VI-A and having multiple data consumers per radio.

The current draft also contains the same encoding problem for the relative link quality as the PPPoE extension. This could be easily resolved by not adding the corresponding TLV to the DLEP message if the value cannot be measured or calculated.

To exploit the full potential of DLEP, our current implementation is NOT DLEP-draft compatible but tries to present a less complex and fully RFC5444 using implementation. It also contains additional metric TLVs to transport more layer-2 data that can be acquired by the Linux mac80211 subsystem’s netlink interface.

IV. Node Designs in Community Networks

The typical node of a Community Network [19] is a site with one or multiple antennas which are connected to the computer hardware of the site. By connecting the different digital links of the site with a routing protocol, the sites can span a far reaching IP network together. Community networks either use distance vector or link state protocols for dynamic routing. Examples of Community Networks using BGP are the Athens Wireless Metropolitan Network (AWMN) [20] in Greece and Guifi.net [21] in Spain. Examples of Community Networks using OLSR are Funkfeuer [16] in Austria and Freifunk [17] in Germany.

One of the most important lessons about using wireless communication learned by Community Networks was that the computer part of a node (both hardware and software) does not matter if the radio links between the nodes are bad. This fact slowly transformed early Community Networks from single radio routers with omni-directional antennas placed on the window ledge to sites with multiple directional antennas mounted on poles or roof tops.

While it would be preferable to keep the computer hardware inside, where it is protected from the environment and easy to service, the length of the antenna cables is another factor which restricts node design. Every meter of cable to transport radio signals between antenna and transceiver is a loss in signal strength and link quality. Because of this, most Community Networks place their computer hardware as close as possible to the antennas.

While getting outdoor computer hardware (or placing other hardware in outdoor proof casings) is possible, the growing size of nodes quickly met another limit. Firstly, embedded hardware with multiple wireless radios is more expensive per radio than the cheap DSL routers used by a lot of Community Networks. Secondly, even with more expensive hardware, there is a limit in how many radio devices you can put into a single embedded computer. This can only be avoided by using multiple embedded computers, each connected with a local Ethernet.

The restraints above have produced two different kinds of nodes for sites with multiple antennas:

- The first one is a single embedded computer with multiple radios, mounted in a box on the roof top (or just below it) and connected to all the antennas. This design might be expensive and limited in the number of radios, but it only contains a single computer to configure, which makes the design easy to maintain.

- The second one is a distributed network of small embedded computers, most of them with a single radio device directly (or very closely) mounted on the antenna. These devices are cheaper in total and have nearly no loss of radio signal strength within the antenna cables, but are more complex to maintain because of their number. Changes in the site’s configuration often lead to reconfiguration of all installed devices.

V. Using the DLEP Protocol to Enhance Community Network Node Design

As described in chapter IV, the current Community Network node design is often a trade-off between an expensive but powerful centralized box with lots of wireless links and a high maintenance solution with multiple distributed small nodes close to the antennas.

A solution would be preferable that scales both beyond the maximum number of wireless radios that can be built into a single computer and allows the user to buy cheap single radio devices that can be mounted directly on the antennas. Especially for Community Networks using link state routing (see section IV), the design should also allow access to the layer-2 metric data necessary to support the future introduction of metrics like ETT.

Figure 3 shows our proposal to simplify the distributed node design. The proposed design is often cheaper and does not need long antenna cables, while maintaining access to layer-2 metric data. Radio devices (left and right device) should be flashed with a very simple firmware. They contain a DLEP radio agent and enough software to remotely configure the Wi-Fi card inside (a feature which also might be integrated into a DLEP extension). The router device (center node) detects the radios via DLEP’s automatic discovery. It then connects to each device, configures the radio and commands the device to bridge the Wi-Fi card directly to a VLAN-tagged Ethernet interface. This allows the router to communicate via all Wi-Fi devices without any routing software present on the radio devices which only operate at layer-2.

This transforms the distributed network of layer-3 routers into a distributed layer-2 network of Wi-Fi interfaces which are all configured by the central router. The radio devices should need no configuration at all. The whole Wi-Fi configuration is done on demand when the router establishes a connection to the radio device e.g. during its boot process.
Bridging Wi-Fi in Ad hoc mode is normally difficult, because most Wi-Fi cards in this mode do not allow sending layer-2 data with a different MAC address. To avoid this problem, the current OpenWRT trunk contains a small kernel module called TRelay, which allows bridging two interfaces together without any MAC checks. Using this interface, the router can configure its side of the VLAN-tunnel to the same MAC address as the Wi-Fi card on the radio, which sidesteps the bridging problem.

VI. DLEP - IMPLEMENTATION DETAILS

This section describes the details of our DLEP based protocol implementation using the OLSR.org [22] framework code for OLSRv2. Note that this implementation is accessible via OLSR.org and is intentionally not compliant with the draft 02 version of DLEP [12].

A. Use Cases for the implementation

During the discussion of the DLEP protocol within the IETF Manet Working group, within the CONFINE project [23], and community meetings, three basic use cases for the DLEP protocol have been presented.

In the default use case described in the DLEP-draft, the DLEP component on the radio announces its presence over a control channel with a multicast (or broadcast) discovery message. The DLEP component on the router receives the discovery messages of the radio and replies with a connect message. The connect message then causes the radio to transmit its metric and neighborhood data back to the router.

While the default use case allows to configure the radio (through the connect message of the router), the current draft version is more complicated than necessary in some scenarios. In the second use case which was discussed on the IETF Manet list, the radio proactively multicasts the metric and neighborhood data into the control channel, while the router is just passively listening to the data. This simplifies the state and message handling of the protocol substantially.

A third use case is important for the CONFINE research node architecture [24]. Depending on the configuration, the receiving DLEP router within an experiment container (e.g. to support cross layer routing or other experiments requiring layer-2 information) might not be within link-local multicast distance. Because of this, the router component is configured with the IP address of the DLEP-capable radio and unicasts its connect message to the radio without waiting for a discovery message. The radio then starts to unicast its data to the DLEP router.

B. Features of the DLEP implementation based on the OLSRv2 Framework

The DLEP implementation is based on the OLSRv2 Framework (see figure 4) which provides most of the basic functionality like logging, configuration management including runtime changes, RFC 5444 encoding, dynamic plugin handling and a lot of network related functions. It also has an integrated telnet and http server which can be used by both an application and the loaded plugins. Some available plugins are generic, which means they are directly based on the Framework code and can be used with any application. Others can be application specific.

The DLEP implementation is built around a number of plugins for the OLSRv2 framework and an internal layer-2 database (see figure 5).

The database can record both OSI layer-2 data about links between neighbors and general data about OSI layer-2 networks, both identified by the MAC addresses. It stores a list
of predefined metrics and data and also remembers which data is currently set for a specific data entry. This allows to query the database and check whether the necessary data is available.

The database also serves as an abstraction layer between the layer-2 data provider and the consumer, allowing to change one of them without any need to adapt the other one.

The primary link layer data source for the current implementation is the nl80211 listener plugin. The plugin uses a Linux-specific netlink socket to query the mac80211 network stack for available link layer metrics and statistics. The data gathered by this plugin is then put into the layer-2 database. The configuration of this plugin includes identifiers of the IEEE 802.11 interfaces to query and the time interval between queries.

The DLEP implementation itself is split into two parts. Together, the plugins allow the user to mirror a layer-2 database from one application to another one over UDP.

The first part is the DLEP radio plugin. The job of this plugin is to take the data present in the local layer-2 database and make its contents available over a UDP-based DLEP protocol. The plugin has configuration options for both the frequency and the validity time of discovery and layer-2 update messages. An interval of 0 for the discovery interval switches discovery messages off completely (see second use case). There is also a parameter to tell the plugin to send out layer-2 data updates on its own without a connected DLEP router (also for second use case). In addition, there is a full set of parameters to configure a dual stack UDP socket including multicast options and an access control list.

The second part is the DLEP router plugin which receives the information from the DLEP radio and puts it into the local layer-2 database. The plugin allows to configure both the interval and validity time of the connect messages to the DLEP radio. An interval of 0 switches off the connect messages completely (second use case). There is another configuration option to set the unicast address of a list of DLEP radio plugins to allow communication beyond link-local distance (third use case). The plugin also contains a set of socket configuration options similar to those of the DLEP radio.

A program based on the OLSRV2 framework and using the layer-2 database would either involve the nl80211-listener plugin directly to populate the database or it would use the DLEP router plugin to mirror a remote database. External programs can use the Layer2 viewer plugin in a framework based daemon, which makes the data of the local layer-2 database remotely available via the framework’s internal telnet service.

C. Direct access via Framework Code

If the application that needs the layer-2 data is directly using the Framework API code, it can get its data directly from the layer-2 database. There are two getter functions in olsr_layer2.c which each provide access to the stored network-wide or link-specific data blocks.

For each stored data item in both kinds of blocks, there is an inline function to check if the data is available in the block and another inline function to set the data value (which in turn marks the data as being available). Read access is done directly by accessing the fields of the C struct.

Each layer-2 network data is identified by a MAC address of the radio. For each of the networks, the database stores whether it is active or not. The data entries are automatically set to inactive if the validity time or the information has expired without an update. In addition, the interface index is stored (zero for remote networks via DLEP).

The optional data fields for layer-2 networks are currently:

- **SSID**: Identification of the network
- **Last Seen**: Number of milliseconds since the network was active the last time
- **Frequency**: Frequency of the network in Hz
- **Supported Rates**: Array of supported data rates in bit/s

The layer-2 link data is identified both by the radio MAC and the MAC of the link partner. The database stores if the neighbor is active, similarly to the layer-2 network data. It also stores the interface index if the neighbor belongs to a local interface.

The optional data fields for layer-2 neighbors are currently:

- **Signal**: Last measured signal strength of the neighbor in dBm
- **Last seen**: Number of milliseconds since the neighbor was active the last time
- **Tx/Rx bitrate**: Current transmission/reception unicast bitrate in bit/s
- **Tx/Rx packets**: Number of packets sent/received with this neighbor
- **Tx/Rx bytes**: Number of bytes sent/received with this neighbor
- **Tx retries**: Number of link layer retransmissions with this neighbor
- **Tx failed**: Number of failed transmissions with this neighbor

Because of the modular structure of the RFC 5444 packet format in DLEP, it is easy to add new layer-2 data fields to the DLEP transport format and the internal layer-2 database.

D. Data access via Telnet

For applications not being able or willing to use the existing Framework API code, there is a daemon based on the Framework to allow using a TCP-based Telnet or HTTP client to access layer-2 information. The telnet command layer2 for accessing the database is directly hooked into the Framework’s central telnet server. It provides several subcommands to access the neighbor and network database.

`neigh`, `neigh_inactive` and `neigh_full` are the first three subcommands. They provide a list of active, inactive and all known layer-2 neighbors in the database. The second three subcommands are `net`, `net_inactive` and `net_full`. They work similarly to the neighbor-related commands, but provide information about the known layer-2 networks. All these six subcommands support four different output modes.
• Without an additional parameter, they provide a key-value list for each data point, with an empty line between different networks or neighbors.
• The parameter list provides a short table with only the interface (if local) and the MAC addresses of the entries.
• The parameter json generates a full list of all data points in JSON format.
• The user can also provide a pattern for each line of output, which will be used by the template engine of the plugin in order to generate custom format feedback.

This combination of subcommands and output modes should make it easy for other applications to parse the output of the Telnet command.

```
echo /layer2 neigh json | nc 127.0.0.1 2007
{
  "neighbor" : "00:15:6d:84:0a:03",
  "radio" : "00:15:6d:84:08:bd",
  "ifindex" : 0,
  "interface" : "",
  "active" : true,
  "shortactive" : "",
  "lastseen" : 2.566,
  "signal" : 7,
  "rtxbitrate" : 56623104,
  "rtxbytes" : 13293418,
  "rtxpackets" : 359260,
  "txbitrate" : 56623104,
  "txbytes" : 1068,
  "txpackets" : 10,
  "txretries" : 0,
  "txfailed" : 0
}
```

E. Data access via HTTP

Instead of providing a full HTTP command for accessing layer-2 data, a generic plugin called httpTelnet can be used. This plugin creates a single HTML page, which provides an interface to the telnet server. By supplying the command and its parameter string with HTTP GET or POST requests, the user can trigger the telnet command described in the previous section.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we presented our first steps toward an implementation of the Dynamic Link Exchange Protocol (DLEP) for layer-2 information exchange between a router and one or several radio devices. In the context of Community Networks, DLEP will allow for an improved, more robust and scalable node design with separated radio and routing devices, while still providing access to layer information as a basis for complex routing metrics. In addition, we provide a discussion about the design of the current draft version of the protocol and provide suggestions, how the current draft version could be changed to support the requirements of Community Networks.

Future versions of our DLEP implementation will also support the remote configuration of radio devices by the router and allow for radios with multiple lines.

ACKNOWLEDGMENT

This effort/activity is supported by the European Community Framework Programme 7 within the Future Internet Research and Experimentation Initiative (FIRE), Community Networks Testbed for the Future Internet (CONFINE), contract FP7-288535.

REFERENCES