Source Routing

TEP:	138
Group:	Net2 Working Group
Type:	Documentary
Status:	Draft
TinyOS-Version:	> 2.1
Author:	Chieh-Jan (Mike) Liang, Doug Carlson, Maria A. Kazandjieva, and Omprakash Gnawali
Draft-Created:	28-May-2010
Draft-Version:	1.2
Draft-Modified:	2010-06-09
Draft-Discuss:	TinyOS Developer List <tinyos-devel at mail.millennium.berkeley.edu>

Abstract

This memo documents the interfaces and components used for source routing in TinyOS 2.x. Source routing allows its users to specify the path a packet must take to reach the destination.

1. Introduction

Source routes are paths explicitly provided by its users. These routes, often in the form of a list of nodes to be traversed on the way to the destination, are carried in the packet header. When a node receives a source-routed packet, it looks up the next hop in the source routing packet header and forwards the packet to that node.

A complete source routing system is comprised of two sets of functionality: the first deals with reading and writing source routes and the second handles forwarding based on source routes.

A source routing system can be used in many ways. This can be useful as a light-weight routing system for debugging or for environments that are static enough that the routes are known ahead of time. It can also be used by routing systems that compute the path at a given node based on topology information from the network. For example, given the link state information of a network, a node can source route a packet to any node in the network. Source routing is also useful to route a packet along the reverse path to the original sender.

Our approach provides the most basic source routing system as an example of a light-weight end-to-end protocol that can also be used in conjunction with other routing and forwarding systems. Our focus in this document is to specify a best-effort source routing system.

The rest of this document describes the source routing interfaces and components.

2. Source Routing Interfaces

A node can perform three different roles in source routing:

- Sender: Configuring the source route and send a packet
- Forwarder: Receive a source routed packet and forward it to the next hop
- Receiver: Receive a source routed packet at the destination

Multiple sender applications can use the source routing system. These senders are uniquely identified by sourceroute identifiers. These identifiers work like collection identifiers in collection protocols[1].

The sender uses the SourceRouteSend interface to set the contents of a packet. The sender uses the SourceRouteSend interface to introduce a packet into the network.

```
interface SourceRouteSend {
  command error_t send(am_addr_t *path, uint8_t pathLen, message_t* msg, ui
  command error_t cancel(message_t* msg);
  event void sendDone(message_t* msg, error_t error);
  command uint8_t maxPayloadLength();
  command void* getPayload(message_t* msg, uint8_t len);
}
```

The forwarder uses the SourceRoutePacket interface to update the header as it forwards the packet:

```
interface SourceRoutePacket
{
  command am_addr_t address();

  command error_t clearRoute(message_t *msg);

  command error_t setRoute(message_t *msg, nx_am_addr_t *path, uint8_t len)
  command nx_am_addr_t* getRoute(message_t *msg);

  command uint8_t getRouteLen(message_t *msg);
  command error_t setRouteLen(message_t *msg, uint8_t len);

  command am_addr_t getNextHop(message_t *msg);
  command am_addr_t getDest(message_t *msg);

  command uint8_t getCurHop(message_t *msg);
```

```
    command error_t setCurHop(message_t *msg, uint8_t hop);

  }
```

3. Source Routing Services

A source routing service MUST provide one SourceRoutingC, which has the following signature:

```
configuration SourceRoutingC {
  provides {
    interface StdControl;
    interface SourceRouteSend[uint8_t client];
    interface SourceRoutePacket;
    interface Receive[sourceroute_id_t id];
  }
  uses {
    interface SourceRouteId[uint8_t client];
  }
}
```

4. Implementation

An implementation of source routing service will be found in `tinyos-2.x/tos/lib/net/sourceroute`.

5. Author Addresses

Chieh-Jan (Mike) Liang
213 NEB, 3400 N Charles St
Johns Hopkins University
Baltimore, MD 21211

email - cliang4@cs.jhu.edu

Doug Carlson
213 NEB, 3400 N Charles St
Johns Hopkins University
Baltimore, MD 21211

email - carlson@cs.jhu.edu

Maria A. Kazandjieva
284 Gates Computer Science, 353 Serra Mall
Stanford University
Stanford, CA 94305

email - mariakaz@cs.stanford.edu

Omprakash Gnawali
S255 Clark Center, 318 Campus Drive
Stanford University
Stanford, CA 94305

phone - +1 650 725 6086
email - gnawali@cs.stanford.edu

6. Citations

[1] TEP 119: Collection