

How to hook your designed routing protocol into GlomoSim

By Jian ZHAO
zhao@cs.queensu.ca

Recently, there are lots of questions asked by GlomoSim users on how to hook their own routing protocol. At this time, I would like to write my own experiences below.

(I) First, let me assume that I have designed a routing protocol called AQL (Average Queueing Length Based Routing Protocol in Ad Hoc Networks). Also our three variables are protocolString=AQL, routingProtocolChoice=ROUTING_PROTOCOL_AQL. Also, in user_nwip.pc, there are routingProtocolString=protocolString, routingProtocol=routingProtocolData.

(II) Go to NWCOMMON.h file, and add one line like below

```
/* protocol number for IP */
#define IPPROTO_TCP 6
#define IPPROTO_UDP 17
#define IPPROTO_OSPF 87
#define IPPROTO_BELLMANFORD 520
#define IPPROTO_FISHEYE 530
#define IPPROTO_AODV 123
#define IPPROTO_DSR 135
#define IPPROTO_ODMRP 145
#define IPPROTO_LAR1 110
#define IPPROTO_ZRP 133
#define IPPROTO_AQL 188
#define NETWORK_UNREACHABLE -2
#define DEFAULT_INTERFACE 0
```



(III) Go to NETWORK.H file, and add one line into it

```
typedef enum {
    NETWORK_PROTOCOL_IP = 0,
    ROUTING_PROTOCOL_AODV,
    ROUTING_PROTOCOL_DSR,
    ROUTING_PROTOCOL_LAR1,
    ROUTING_PROTOCOL_ODMRP,
    ROUTING_PROTOCOL_OSPF,
    ROUTING_PROTOCOL_ZRP,
    ROUTING_PROTOCOL_ALL,
    ROUTING_PROTOCOL_AQL,
    ROUTING_PROTOCOL_NONE
} NetworkRoutingProtocolType;
```



(IV) Go to USER_NWIP.PC file, and make some modification on function NetworkIpUserProtocolInit(.....):

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <assert.h>
#include <math.h>

#include "api.h"
#include "structmsg.h"
#include "fileio.h"
#include "message.h"
#include "network.h"
#include "ip.h"
#include "nwip.h"

// The Ip Protocol is 0..255, and must be different than the GloMoSim values for IP protocols (some standard, some //arbitrarily picked).
// "IPPROTO_" is the current prefix for these numbers in GloMoSim.

#define IPPROTO_AQL 188 ← MODIFICATION HERE

// The GLOMOSIM protocol number is a "unsigned short" that must be unique within the layers. So pick a random //one to get near zero
probability of conflict.

#define ROUTING_PROTOCOL_AQL 188 ← MODIFICATION HERE

void NetworkIpUserProtocolInit(
    GlomoNode *node,
    const GlomoNodeInput *nodeInput,
    const char* routingProtocolString,
    NetworkRoutingProtocolType* routingProtocolChoice,
    void** routingProtocolData)
{
    if (strcmp(routingProtocolString, "AQL") == 0) {
        *routingProtocolChoice = ROUTING_PROTOCOL_AQL;

        // Put function to create your datastructures and
        // initialize your protocol here. For Example,

        // YourRoutingProtocolInit(node, nodeInput,
        // (YourType**)routingProtocolData);
        RoutingAqlInit(node, nodeInput, (GlomoRoutingAql**)routingProtocolData);
    }
    else {
        fprintf(stderr, "CONFIG.IN Error: Unknown Routing ProtocolType "
            "Type %s.\n", routingProtocolString);
        assert(FALSE); abort();
    }
}
```

(V) Go to USER_NWIP.PC file, and make some modifications on function NetworkIpUserHandleProtocolEvent (.....):

```
void NetworkIpUserHandleProtocolEvent(GlomoNode* node, Message* msg) {
    switch (msg->protocolType) {
        case ROUTING_PROTOCOL_AQL: {

            // Put function to handle GlomoSim Events for your protocol here.
            // For Example,
            // YourRoutingProtocolEventHandler(node, msg);
            RoutingAqlProtocolEventHandler(node, msg);
        }
        default:
            printf("GloMoSim: Network IP Protocol %d Not Handled\n",
                msg->protocolType);
            assert(FALSE); abort();
    } /*switch*/
}
```

(VI) Go to USER_NWIP.PC file, and make some modifications on function NetworkIpUserHandleProtocolPacket (.....):

```
void NetworkIpUserHandleProtocolPacket(
    GlomoNode* node,
    Message* msg,
    unsigned char ipProtocol,
    NODE_ADDR sourceAddress,
    NODE_ADDR destinationAddress,
    int ttl)
{
    switch (ipProtocol) {
        case IPPROTO_AQL: {

            // Put function call to handle Ip protocol packets (with IP header
            // already stripped).
            // For Example,
            // YourIpPacketHandlingFunction(node, msg, sourceAddress,
            //     destinationAddress, ttl);
            RoutingAqlHandleProtocolPacket(GlomoNode *node, Message *msg, NODE_ADDR srcAddr,
                NODE_ADDR destAddr, int ttl);

            break;
        }
        default:
            printf("Ip Protocol %d Not Handled\n", ipProtocol);
            assert(FALSE); abort();
    } /*switch*/
}
```

(VII) Go to USER_NWIP.PC file, and make some modifications on function NetworkIpUserHandleProtocolPacket (.....):

```
void NetworkIpUserProtocolFinalize(
    GlomoNode* node,
    int userProtocolNumber)
{
    switch (userProtocolNumber) {
        case ROUTING_PROTOCOL_AQL: {

            // Put function call to handle routing protocol finalization here:
            // For Example,
            // YourFinalizationRoutine(node);
            RoutingAqlFinalize(GlomoNode *node);

        }
        default:
            printf("Glomosim Protocol %d Not Handled\n", userProtocolNumber);
            assert(FALSE); abort();
    } /*switch*/
}
```

(VIII) At last, go to BIN directory of GlomoSim, and define your designed routing protocol in CONFIG.IN file.

```
NETWORK-PROTOCOL  IP
NETWORK-OUTPUT-QUEUE-SIZE-PER-PRIORITY 100

#RED-MIN-QUEUE-THRESHOLD 150
#RED-MAX-QUEUE-THRESHOLD 200
#RED-MAX-MARKING-PROBABILITY 0.1
#RED-QUEUE-WEIGHT .0001
#RED-TYPICAL-PACKET-TRANSMISSION-TIME 64000NS
#####

ROUTING-PROTOCOL  AQL
#ROUTING-PROTOCOL  BELLMANFORD
#ROUTING-PROTOCOL  AODV
#ROUTING-PROTOCOL  DSR
#ROUTING-PROTOCOL  LAR1
#ROUTING-PROTOCOL  WRP
#ROUTING-PROTOCOL  FISHEYE

#ROUTING-PROTOCOL  ZRP
#ZONE-RADIUS        2

#ROUTING-PROTOCOL  STATIC
#STATIC-ROUTE-FILE  ROUTES.IN
```