

# AntNet Algorithm Implementation on NS-2 .33



**antnet-1.0**

**Richardson Lima**

[securityfusion@gmail.com](mailto:securityfusion@gmail.com)

[richardson@gprt.ufpe.br](mailto:richardson@gprt.ufpe.br)

<http://antnet.wordpress.com>

**Networking and Telecommunications Research Group,  
Computer Science Center of the Federal University of Pernambuco at Recife, Brazil**

Friday, February 27, 2009

# Contents

## 1 Introduction

### 2 AntNet on NS-2

- 2.1 Configuration and Installation .....
- 2.2 Example Scripts .....

### Bibliography

## List of Figures

- 2.1 Mesh topology .....
- 2.2 Ring topology .....
- 2.3 Arbitrary topology .....

## Listings

- 2.1 common/packet.h: enum packet t .....
- 2.2 common/packet.h: p info() .....
- 2.3 trace/cmu-trace.h .....
- 2.4 trace/cmu-trace.cc .....
- 2.5 trace/cmu-trace.cc: format antnet() .....
- 2.6 trace/cmu-trace.cc: format() .....
- 2.7 queue/drop-tail.h .....
- 2.8 queue/drop-tail.cc .....
- 2.9 queue/priqueue.cc .....
- 2.10 tcl/lib/ns-packet.tcl .....
- 2.11 tcl/lib/ns-default.tcl .....
- 2.12 tcl/lib/ns-lib.tcl: create-wireless-node .....
- 2.13 tcl/lib/ns-lib.tcl: create-antnet-agent .....
- 2.14 tcl/lib/ns-lib.tcl: get-drop-queue .....
- 2.15 Makefile .....

# Chapter 1

## Introduction

AntNet is an Ant Colony Optimization (ACO) metaheuristic for data network routing proposed by Gianni Di Caro and Marco Dorigo ( <http://www.idsia.ch/~gianni/antnet.html> ). We have implemented AntNet on network simulator, NS2.

This manual describes how to modify NS2 and install our code to simulate and experiment with AntNet algorithm created by Lavina Jain. **We have tested our simulations on ns-2.33.**

## Chapter 2

### AntNet on NS-2

This chapter provides documentation on extending NS-2 to simulate AntNet. We have implemented new Agent called Antnet and new packet type called Ant. The simulations have been tested on ns-2.33.

## 2.1 Configuration and Installation

This section describes how to install antnet-1.1 and modify NS-2 files. We assume that you have NS2 installed and running on your system and you are familiar with NS2 basics.

1. Download antnet-1.0.tgz ( <https://antnet.wordpress.com> ) and extract to a location of your choice.
2. Copy folder “antnet” to NS2 base directory.
3. Copy folder “scripts” to any location of your choice where you wish to run simulation scripts and store results.
4. Edit common/packet.h.

### (a) Add packet type PT\_ANT to packet\_t as shown in Listing 2.1

```
typedef unsigned int packet_t;  
static const packet_t PT_TCP = 0;  
static const packet_t PT_UDP = 1;  
static const packet_t PT_CBR = 2;  
static const packet_t PT_AUDIO = 3;  
static const packet_t PT_VIDEO = 4;  
// insert new packet types here  
static const packet_t PT_ANT = 62;  
static packet_t PT_NTTYPE = 61;
```

Listing 2.1: [common/packet.h](#)

## Chapter 2

### AntNet on NS-2

(b) Add packet name to constructor of class `p_info()` as shown in Listing 2.2

```
class p_info {
    // XCP
    name_[PT_XCP]="xcp";
    // Bell Labs (PackMime OL)
    name_[PT_BLTRACE]="BellLabsTrace";
    name_[PT_ANT] = "Ant";
    name_[PT_NTTYPE]= "undefined";
}
```

Listing 2.2: `common/packet.h: p_info()`

5. Edit `trace/cmu-trace.h` as shown in Listing 2.3 to add new function that defines trace format for AntNet.

```
class CMUTrace : public Trace {
public:
    /*...*/
private:
    /*...*/
    void format_rtp(Packet *p, int offset);
    void format_tora(Packet *p, int offset);
    void format_imep(Packet *p, int offset);
    void format_aodv(Packet *p, int offset);
    void format_antnet(Packet *p, int offset);
    // This holds all the tracers added at run-time
    static PacketTracer *pktTrc_;
};
```

Listing 2.3: `trace/cmu-trace.h`

6. Edit `trace/cmu-trace.cc`.

(a) Include header file for Ant packet as shown in Listing 2.4.

```
#include <antnet/ant_pkt.h> //ANTNET
/*...*/
```

## Chapter 2

### AntNet on NS-2

#### Listing 2.4: [trace/cmu-trace.cc](#)

(b) Define `format_antnet()` as shown in Listing 2.5.

```
void
CMUTrace::format_antnet(Packet *p, int offset)
{
    struct hdr_ant_pkt *ah = HDR_ANT_PKT(p);
    if (pt_->tagged()) {
        sprintf(pt_->buffer() + offset,
            "-ant:o %d -ant:s %d -ant:l %d ",
            ah->pkt_src(), ah->pkt_seq_num(), ah->pkt_len());
    }
    else if (newtrace_) {
        sprintf(pt_->buffer() + offset,
            "-P ant -Po %d -Ps %d -Pl %d ",
            ah->pkt_src(), ah->pkt_seq_num(), ah->pkt_len());
    }
    else {
        sprintf(pt_->buffer() + offset,
            "[ant %d %d %d] ",
            ah->pkt_src(), ah->pkt_seq_num(), ah->pkt_len());
    }
}
}
```

#### Listing 2.5: [trace/cmu-trace.cc: format antnet\(\)](#)

Edit `format()` as shown in Listing 2.6 to add case for Ant packet.

```
void CMUTrace::format(Packet* p, const char *why)
{
    /* ... */
    default:
        format_ip(p, offset);
        offset = strlen(pt_->buffer());
        switch(ch->ptype()) {
            case PT_AODV:
                format_aodv(p, offset);
                break;
            case PT_ANT:
                format_antnet(p, offset);
                break;
            case PT_TORA:
                format_tora(p, offset);
                break;
        }
    /* ... */
}
```

## Chapter 2

### AntNet on NS-2

7. Edit queue/drop-tail.h to define a method getlength() in class DropTail as shown in Listing 2.7.

```
class DropTail : public Queue {  
public :  
    /*...*/  
protected :  
    /*...*/  
public :  
    int getlength ( ) ;  
};
```

Listing 2.7: queue/drop-tail.h

## Chapter 2

### AntNet on NS-2

8. Edit queue/drop-tail.cc to add method getlength() as shown in Listing 2.8.

```
int DropTail::getlength ()
{
    return q_>length ();
}
```

**Listing 2.8:** queue/drop-tail.cc

9. Edit recv() in queue/priqueue.cc to assign priorities to Ant packets as shown in Listing 2.9.

```
void
PriQueue::recv ( Packet *p , Handler *h )
{
    struct hdr_cmn * ch = HDR_CMN( p ) ;
    if ( Prefer Routing Protocols ) {
        switch ( ch->ptype () ) {
            case PT_DSR :
            case PT_MESSAGE:
            case PT_TORA:
            case PT_AODV:
                recvHighPriority ( p , h ) ;
                break ;
            case PT_ANT:
                if ( ch->direction () == hdr_cmn::UP ) {
                    recvHighPriority ( p , h ) ;
                }
                else {
                    Queue :: recv ( p , h ) ;
                }
                break ;
            default :
                Queue::recv ( p , h ) ;
        }
    }
    else {
        Queue::recv ( p , h ) ;
    }
}
```

**Listing 2.9:** queue/priqueue.cc

10. Edit tcl/lib/ns-packet.tcl to add new protocol as shown in Listing 2.10.

```
foreach prot {
    Antnet
```

## Chapter 2

### AntNet on NS-2

```
AODV
#...
NV
} {
    add-packet-header $prot
}
}
```

**Listing 2.10:** [tcl/lib/ns-packet.tcl](#)

11. Edit tcl/lib/ns-default.tcl as shown in Listing 2.11 to assign default values for binded attributes.

```
# Defaults for AntNet
Agent/Antnet set num_nodes_x_ 4
Agent/Antnet set num_nodes_y_ 4
Agent/Antnet set num_nodes_ 16
Agent/Antnet set r_factor_ 0.001
Agent/Antnet set timer_ant_ 0.03
```

**Listing 2.11:** [tcl/lib/ns-default.tcl](#)

12. Edit tcl/lib/ns-lib.tcl.

(a) Edit create-wireless-node as shown in Listing 2.12 to create an instance of Antnet Agent for a node.

```
Simulator instproc create-wireless-node args {
    #...
    switch -exact $routingAgent_ {
        Antnet {
            set ragent [ $self create-antnet-agent $node ]
        }
        #...
    }
    #...
}
```

**Listing 2.12:** [tcl/lib/ns-lib.tcl: create-wireless-node](#)

(b) Add code for method create-antnet-agent as shown in Listing 2.13.

```
Simulator instproc create-antnet-agent { node } {
    set ragent [new Agent/Antnet [ $node node-addr ] ]
    $self at 0.0 "$ragent start "
    $node set ragent_ $ragent
    return $ragent
}
```

**Listing 2.13:** [tcl/lib/ns-lib.tcl: create-antnet-agent](#)

## Chapter 2

### AntNet on NS-2

(c) Add method `get-drop-queue` as shown in Listing 2.14 to get queue length.

```
Simulator instproc get-drop-queue { n1 n2 } {  
    $ self instvar link_  
    set q [ $ link_ ($n1:$n2) queue ]  
    return $q  
}
```

**Listing 2.14:** `tcl/lib/ns-lib.tcl: get-drop-queue`

13. Edit Makefile to add object files to `OBJ_CC` variable as shown in Listing 2.15

```
OBJ_CC = \  
    tools/random.o tools / rng .o tools / ranvar.o \  
    common/misc.o common/timer -handler.o \  
    #...  
    antnet/antnet_common.o antnet/antnet_rtable.o antnet/antnet.o \  
    $ (OBJ STL)
```

**Listing 2.15:** `Makefile`

14. Touch `common/packet.cc` as only `packet.h` has been modified and not `packet.cc`.

```
[ns-2.28]$ touch common/packet.cc
```

15. Run `make`.

```
[ns-2.28]$ make
```

The Ant packets and AntNet agent can now be used to run AntNet algorithm using a tcl script.

## Example Scripts

The folder `scripts` contains 3 example tcl scripts to simulate AntNet algorithm on NS2.

1. `noc.tcl`:  
This script simulates a regular 3 x 4 mesh topology
2. `ring.tcl`:  
This script simulates a ring topology of 3 nodes
3. `antnet.tcl`:  
This script simulates an arbitrary topology of 12 nodes

The results of simulation are stored in “`rtable.txt`”. The file name is specified in `antnet common.h`.

## Bibliography

- [1] V. Laxmi, Lavina Jain and M. S. Gaur, “Ant Colony Optimisation Based Routing on NS-2”, International Conference on Wireless Communication and Sensor Networks (WCSN), India, December 2006.
- [2] The VINT Project, The ns Manual,  
<http://www.isi.edu/nsnam/ns/ns-documentation.html>.
- [3] Francisco and Pedro, Implementing a New Manet Unicast Routing Protocol in NS2,  
<http://ants.dif.um.es/masimum/nsrt-howto/pdf/nsrt-howto.pdf>.
- [4] Marc Gries, Tutorial for Network Simulator ns  
<http://www.isi.edu/nsnam/ns/tutorial/index.html>.
- [5] Gianni Di Caro, & Marco Dorigo. (1998). AntNet: Distributed Stigmergic Control for Communication Networks. Journal of Artificial Intelligence Research 9.
- [6] Marco Dorigo and Thomas Stutzle, Ant Colony Optimization.