

## LAMPIRAN

```
blackholefix.cc (~/Downloads/ns-allinone-3.25/ns-3.25/scratch) - gedit
blackholefix.cc
~/Downloads/ns-allinone-3.25/ns-3.25/scratch

#include "ns3/core-module.h" //prosesor pengarah yang mengatakan kepada kompiler untuk meletakkan kode dari header file iostream.h kedalam
program. terletak pada direktori src
#include "ns3/network-module.h"//
#include "ns3/mobility-module.h"
#include "ns3/config-store-module.h"
#include "ns3/wifi-module.h"
#include "ns3/internet-module.h"
#include "ns3/ipv4-flow-classifier.h"
#include "ns3/applications-module.h"

// routing protocols
#include "ns3/aodv-module.h"

// header file specific to this code
#include "myapp.h"

// added for flow monitoring
#include "ns3/gnuplot.h"
#include "ns3/flow-monitor-module.h"
#include "ns3/flow-monitor-helper.h"

// animator
#include "ns3/netanim-module.h"
//
#include <iomanip>
#include <iostream> //menampilkan perintah: cin, cout, endl, ends,
#include <fstream>
#include <vector>
#include <string>
#include <sstream>

NS_LOG_COMPONENT_DEFINE ("Blackhole");

using namespace ns3;
void ThroughputMonitor(FlowMonitorHelper *fmhelper, Ptr<FlowMonitor> monitor, Gnuplot2dDataset DataSet, Gnuplot2dDataset DataSet1,
Gnuplot2dDataset DataSet2);

NS_LOG_COMPONENT_DEFINE ("Blackhole");

using namespace ns3;
void ThroughputMonitor(FlowMonitorHelper *fmhelper, Ptr<FlowMonitor> monitor, Gnuplot2dDataset DataSet, Gnuplot2dDataset DataSet1,
Gnuplot2dDataset DataSet2);

int main(int argc, char* argv[])//hanya deklarasi fungsi utama program.
{
    std::string phyMode ("DsssRate11Mbps");//IEEE 802.11b
    double interval = 0.001; //seconds(waktu pertama dlm simulasi)
    std::string rtslimit = "1500";//bytes(unit informasi digital dalam komputasi dan telekomunikasi)
    double distance = 40;//jarak antar node(meter)
    int m_Xsize = 2;//5 nodes in a row
    int m_Ysize = 10;//5 columns in total

    //enabled for mobility
    int nodesSpeed = 19.4; // km/jam dirubah ke m/s
    int nodePause = 0; // menit

    CommandLine cmd;
    cmd.AddValue ("x-size", "The number of nodes in a row grid", m_Xsize);
    cmd.AddValue ("y-size", "The number of rows in a grid", m_Ysize);
    cmd.AddValue ("distance", "distance (m)", distance);
    cmd.AddValue ("rtslimit", "RTS/CTS threshold in bytes", rtslimit);
    cmd.AddValue ("phyMode", "Wifi Phy mode", phyMode);
    cmd.Parse (argc, argv);

    //memuat topologi
    NS_LOG_DEBUG ("Grid:" << m_Xsize << "*" << m_Ysize);
    // Konversi ke objek waktu
    Time interPacketInterval = Seconds (interval);
    // matikan RTS / CTS untuk frame di bawah 2200 byte
    Config::SetDefault ("ns3::WifiRemoteStationManager::RtsCtsThreshold", StringValue (rtslimit));
    // Perbaiki laju data non-unicast agar sama dengan tingkat unicast
    Config::SetDefault ("ns3::WifiRemoteStationManager::NonUnicastMode", StringValue (phyMode));
    //buat status data dan status net
}
```

```

// Perbaiki laju data non-unicast agar sama dengan tingkat unicast
Config::SetDefault ("ns3::WifiRemoteStationManager::NonUnicastMode", StringValue (phyMode));

//buat simpul jahat dan simpul sah
NS_LOG_INFO ("Create nodes.");

NodeContainer c; // ALL Nodes
NodeContainer not_malicious;
NodeContainer malicious;
c.Create(m_Ysize*m_Xsize);
using namespace std;

//All nodes are not malicious
/*
int n = 0;
while(n<=39){
    not_malicious.Add(c.Get(n));
    n++;
}
*/

// Only one attacker
int n = 5;
while(n<=19){
    not_malicious.Add(c.Get(n));
    n++;
}
not_malicious.Add(c.Get(1));
not_malicious.Add(c.Get(6));
//not_malicious.Add(c.Get(3));
//not_malicious.Add(c.Get(4));
//not_malicious.Add(c.Get(5));
//not_malicious.Add(c.Get(6));
malicious.Add(c.Get(2));
malicious.Add(c.Get(3));
malicious.Add(c.Get(4));
/*

```

C++ Tab Width: 8 Ln 1, Col 1 INS

```

// Only one attacker
int n = 5;
while(n<=19){
    not_malicious.Add(c.Get(n));
    n++;
}
not_malicious.Add(c.Get(1));
not_malicious.Add(c.Get(6));
//not_malicious.Add(c.Get(3));
//not_malicious.Add(c.Get(4));
//not_malicious.Add(c.Get(5));
//not_malicious.Add(c.Get(6));
malicious.Add(c.Get(2));
malicious.Add(c.Get(3));
malicious.Add(c.Get(4));

/*
// three attackers
int n1 = 0;
while(n1<=3){
    not_malicious.Add(c.Get(n1));
    n1++;
}
int n2 = 15;
while(n2<=29){
    not_malicious.Add(c.Get(n2));
    n2++;
}
not_malicious.Add(c.Get(4));
not_malicious.Add(c.Get(18));
not_malicious.Add(c.Get(11));
not_malicious.Add(c.Get(8));
not_malicious.Add(c.Get(7));
not_malicious.Add(c.Get(5));
//not_malicious.Add(c.Get(8));
//not_malicious.Add(c.Get(9));
//not_malicious.Add(c.Get(15));
//not_malicious.Add(c.Get(16));
not_malicious.Add(c.Get(6));

```

C++ Tab Width: 8 Ln 1, Col 1 INS