ตรงนี้จะเป็นส่วนของการทำ simulation โดยไฟล์ simulation จะทำการรันบนโปรแกรม NS-3 และบันทึกผลด้วยการใช้การรันบนฟังชั่น NetAnim และ .xml เพื่อให้เกิดอนิเมชั่น

ฟังชั่นคร่าวๆของ node แต่ละ node นั้น จะมีเพียงการใช้ IP Address ในการสื่อสารเท่านั้น (กำหนดเพียงแค่ IP ไม่มี MAC) และหลังจากรันไฟล์ aodv.cc เสร็จแล้ว จะได้ไฟล์ที่ชื่อว่า netanim.xml ใน directory ns-3-dev เพื่อที่เราจะได้ทำไฟล์ netamim.xml ไปทำการเปิดโปรแกรม NetAnim เพื่อใช้ในการรัน animation

double txpDistance = 100; // m

uint32\_t packetSize = 64; // bytes

uint32\_t numPackets = 10;

double interval = 0.5; // seconds

bool verbose = false;

uint32\_t nNodes = 8;

std::string protocol = "ns3::UdpSocketFactory";

uint16\_t port = 80;

//LogComponentEnable ("AodvRoutingProtocol", LOG\_LEVEL\_DEBUG);

CommandLine cmd;

cmd.AddValue ("phyMode", "Wifi Phy mode", phyMode);

cmd.AddValue ("txpDistance", "Specify node's transmit range [m], Default:100", txpDistance);

cmd.AddValue ("packetSize", "size of application packet sent", packetSize);

cmd.AddValue ("numPackets", "number of packets generated", numPackets);

cmd.AddValue ("interval", "interval (seconds) between packets", interval);

cmd.AddValue ("verbose", "turn on all WifiNetDevice log components", verbose);

cmd.AddValue ("transport TypeId", "TypeId for socket factory", protocol);

ตัวอย่าง environment ในการ simulation ครั้งนี้

MobilityHelper mobility;

Ptr<ListPositionAllocator> positionAlloc = CreateObject<ListPositionAllocator> ();

positionAlloc->Add (Vector (0, 10, 3.0)); //0

positionAlloc->Add (Vector (50, 10, 3.0)); //1

positionAlloc->Add (Vector (100, 10, 3.0)); //2

positionAlloc->Add (Vector (150, 10, 3.0)); //3

positionAlloc->Add (Vector (0, 60, 3.0)); //4

positionAlloc->Add (Vector (50, 60, 3.0)); //5

positionAlloc->Add (Vector (100, 60, 3.0)); //6

positionAlloc->Add (Vector (150, 60, 3.0)); //7

mobility.SetPositionAllocator (positionAlloc);

mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");

mobility.Install (c);

position ของแต่ละ node

Ipv4AddressHelper ipv4;

NS\_LOG\_INFO ("Assign IP Addresses.");

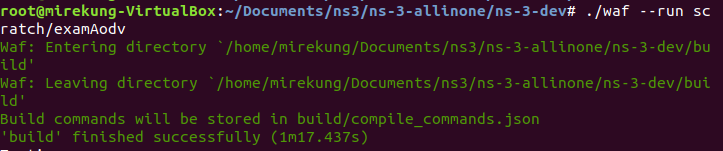
ipv4.SetBase ("192.168.1.0", "255.255.255.0");

Ipv4InterfaceContainer i = ipv4.Assign (devices);

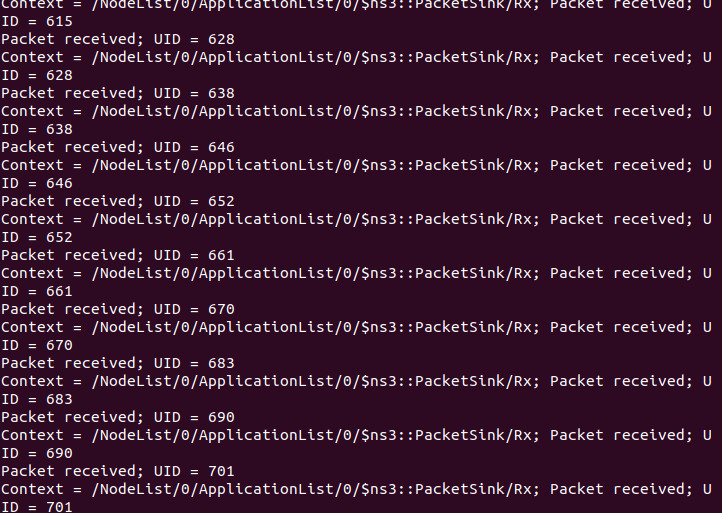
Ip บนอุปกรณ์ subnet /24

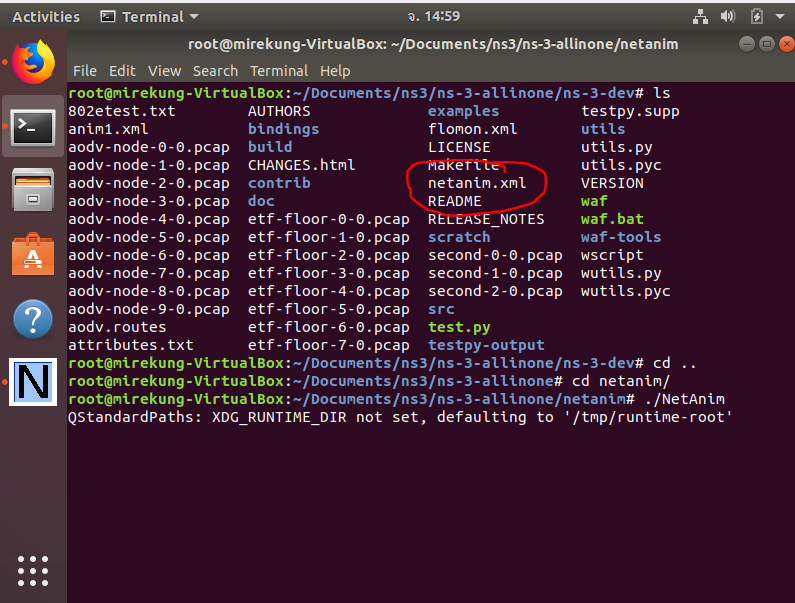
AnimationInterface anim ("netanim.xml");

การกำหนดชื่อไฟล์ที่จะทำการ นำไปรันเป็น animation บน NetAnim

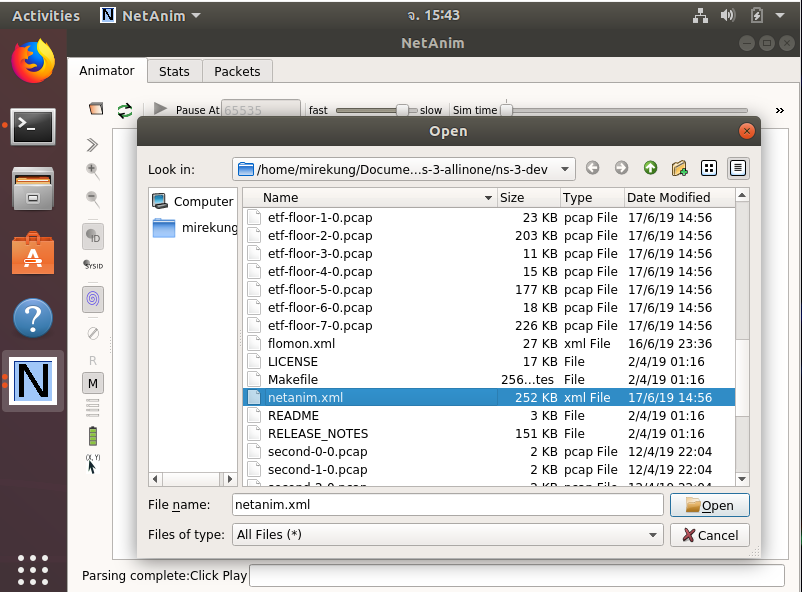


หลังจากเราทำการสั่งรัน เราจะได้การ print out ออกมาเป็นรูปแบบการส่ง packet ของ aodv ซึ่งดูแล้วไม่เข้าใจ

แต่ว่าเหตุผลที่แท้จริงของเราในการรัน คือเราต้องการไฟล์ netanim.xml

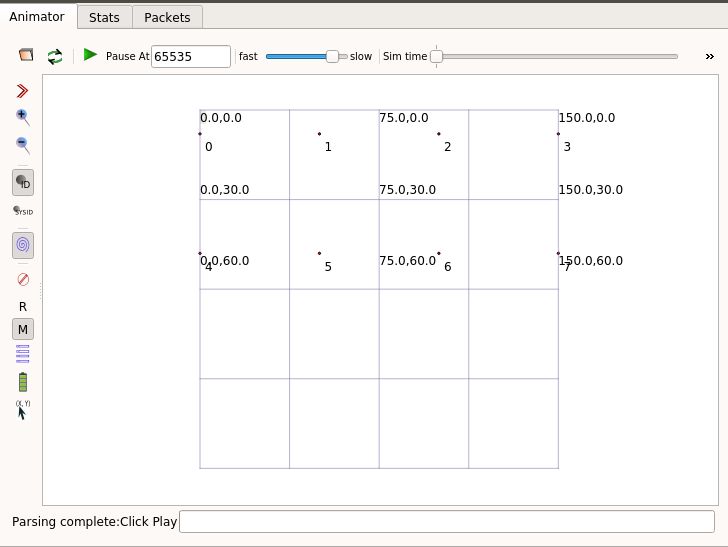


จากนั้นก็ทดสอบด้วยการรันโปรแกรม ./NetAnim

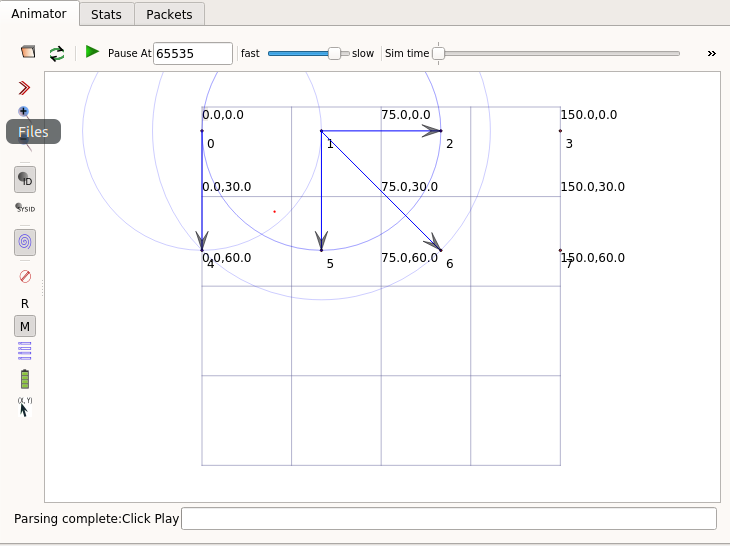


เลือกไฟล์ netanim.xml

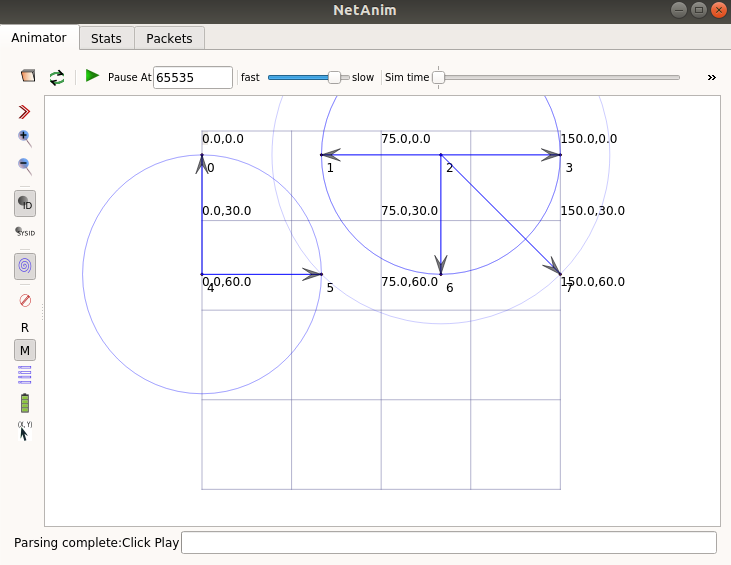
จะได้หน้าต่างประมาณนี้



จากนั้นเราจะทำการรันแบบ step แล้วมาไล่ทีละเสตปเพื่อดูการทำงานของ protocol



Step 1 : source node ของเราคือ 0 ต้องการส่งไปยัง 7 แต่ตัวnode ไม่รู้ว่าตัวเองมี routes ไปถึงหรือไม่จึงทำการ broadcast packet ไปที่node รอบๆ ในที่นี้ก็มี source node อีกตัวนึงคือ 1 ก็ทำการ broadcast



Step 2 : ดู node ที่ได้รับ จาก source node จะมี 0 -> 4 , 1 -> 2 , 1 -> 5, 1 -> 6 จากนั้นแต่ละ node ก็ทำการ rebroadcast RREQ ซึ่ง node ที่ทำการ rebroadcast คือ 1, 4, 5, 6 ไปยัง neighbor ของ node ซึ่งคือ

4 -> 5

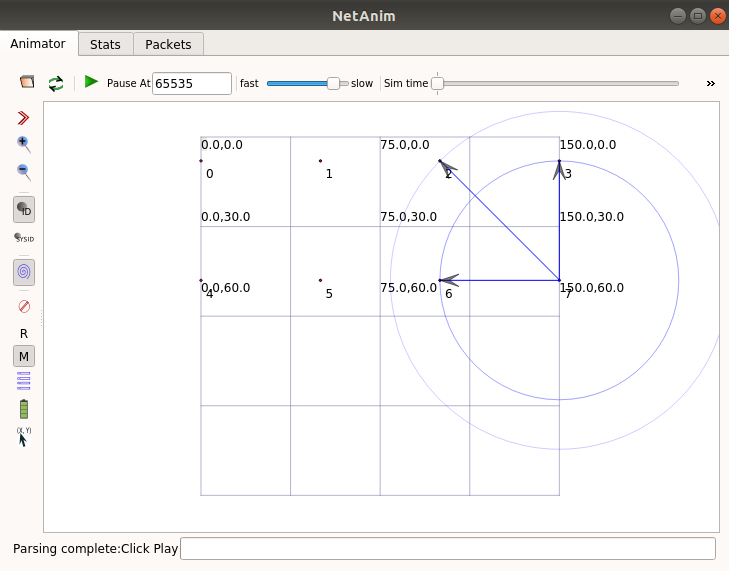
4 -> 1

2 -> 1

2 -> 6

2 -> 3

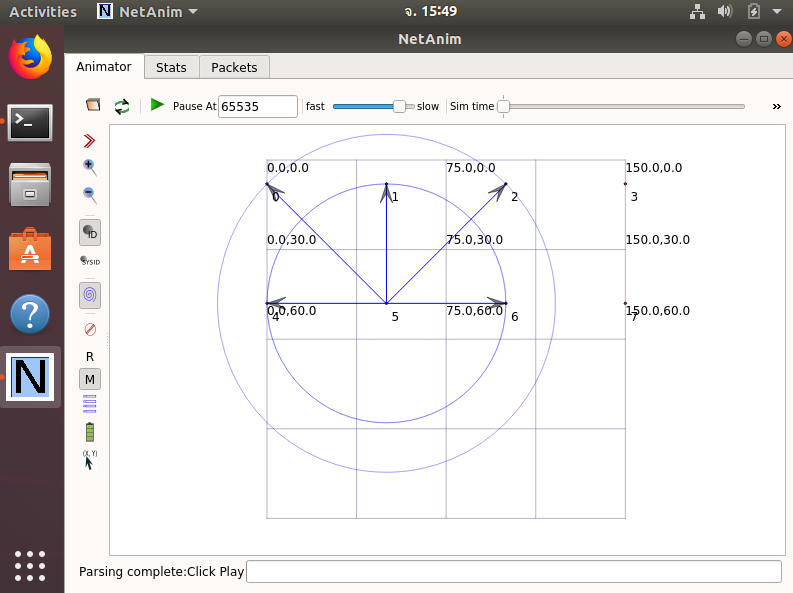
2 -> 7



Step 3 : จาก Step นี้ มีสองหัวข้อหลักๆที่เราต้องดู

Node 5 ไม่ทำการ rebroadcast RREQ ต่อ เนื่องจากได้รับ sequence number ซ้ำนั่นเอง

Node 7 ทำการตอบ RREQ ด้วย RREP กลับไปยัง node 2 ซึ่งเป็น node ที่ทำการส่ง RREQ มาหา node 7



Step 4 : Node 2 ทำการ rebroadcast RREP กลับไปหา source node ผ่าน node 5 (ถึงแม้ตามจริงควรจะส่งกลับไปตามที่ได้รับ RREQ มาก็ตาม)