

PERCOBAAN 7

ANALISA MOBILITY DENGAN MANET

6.1 Tujuan :

Setelah melaksanakan praktikum ini mahasiswa diharapkan mampu :

- Memahami integrasi mobility dengan manet
- Memahami analisa unjuk kerja routing protokol pada manet dengan berbagai mobility

6.2 Peralatan :

- 1 PC dilengkapi dengan OS Ubuntu 18.04
- Software NS-3 versi 3.29
- Software bonnmotion

6.3 Teori :

MOBILE AD-HOC NETWORK (MANET)

Mobile Ad-Hoc Network(MANET) adalah salah satu jenis Ad-Hoc Network yang terdiri atas node-node yang bersifat mobile dan terhubung melalui koneksi nirkabel. Komunikasi antar node dalam MANET berjalan secara independen tanpa intervensi dari akses poin yang terpusat. MANET memiliki sifat self-configuring dan infrastructure-less. Setiap node yang terhubung dalam MANET dapat bertindak sebagai node sumber, node tujuan, ataupun node perantara yang dapat menghubungkan satu node dengan node lain yang tidak dapat berkomunikasi secara langsung. Gambar 1 menunjukkan beberapa contoh perangkat yang dapat digunakan sebagai contoh node pada MANET yaitu laptop, smartphone, dan notebook, yang terhubung satu sama lain.



Gambar 1. Topologi MANET

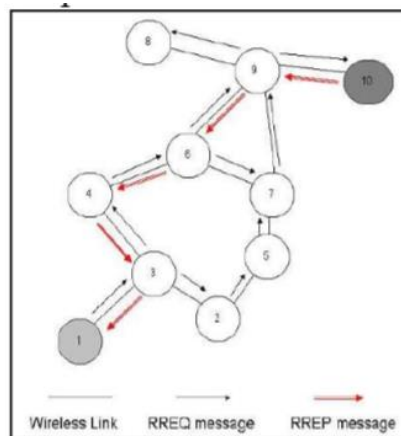
ROUTING PROTOCOL

AODV merupakan salah satu protokol routing reaktif yang digunakan pada Ad-Hoc Network. Pada AODV, tabel routing dibuat saat ada permintaan pengiriman paket data dan belum ada jalur menuju node tujuan. AODV menggunakan sequence number yang diterima dari node tujuan. Penggunaan sequence number bertujuan untuk menjaga informasi pada tabel routing tetap up-to-date dan menghindari perulangan routing yang mungkin dapat terjadi.

Terdapat dua fase dalam operasi AODV, yaitu:

1. Proses Route Discovery

Proses route discovery dilakukan ketika suatu node ingin mengirim paket data ke node tujuan dan tidak memiliki jalur yang valid dalam tabel routing. Maka node sumber akan melakukan broadcast pesan route request (RREQ) kepada node-node tetangganya. Setiap node memiliki sequence number dan broadcast ID yang nilainya akan bertambah setiap kali mengirimkan pesan RREQ. Pesan RREQ berisi sequence number, broadcast ID, dan sequence number baru yang dimiliki RREQ untuk node tujuan. Broadcast pesan RREQ dilakukan sampai ada node tetangga yang memiliki informasi tentang jalur menuju node tujuan. Kemudian node tetangga akan mengirim pesan route reply (RREP). Pengiriman pesan RREP ke node sumber dilakukan bersamaan dengan pembuatan reverse path. Ketika pesan RREP telah mencapai nodesumber dan reverse path telah dibentuk, maka jalur routing telah selesai dibentuk. Proses route discovery ditunjukkan pada Gambar 2.

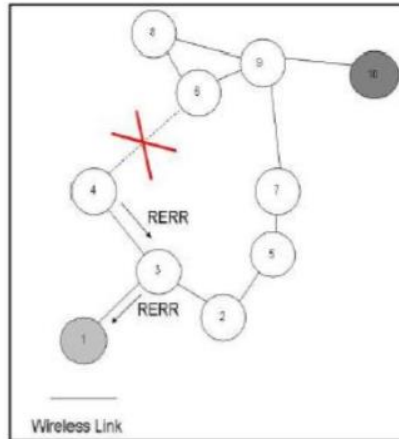


Gambar 2. Proses Route Discovery pada AODV

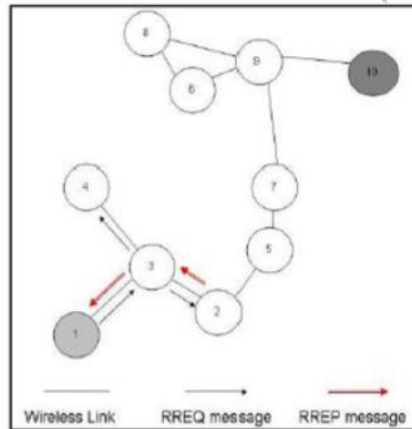
2. Proses Route Maintenance

Jalur routing yang telah terbentuk bertahan selama jangka waktu tertentu. Ketika sebuah jalur yang menghubungkan nodesumber dan node tujuan terputus, maka node

tetangga terdekat dengan lokasi terputusnya jalur akan mengirimkan pesan route error (RERR) kepada nodesumber. Setelah menerima pesan RERR, maka node sumber akan segera melakukan broadcast RREQ baru. Setelah node tetangga mengirim pesan RREP kepada nodesumber, maka reverse path baru telah selesai dibentuk dan siap digunakan untuk mengirim pesan ke node tujuan. Proses route maintenance ditunjukkan pada Gambar 3 dan 4.



Gambar 3. Pengiriman pesan RERR pada AODV

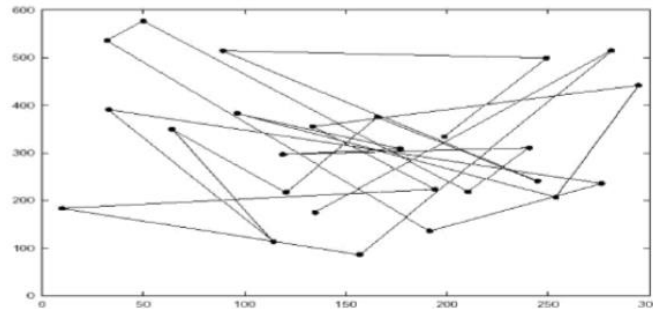


Gambar 4. Proses penemuan rute baru pada AODV

MODEL MOBILITAS

Pada model mobilitas random waypoint (RWP), node didistribusikan secara acak dalam jaringan. Setiap node menyebar secara independen dari node yang lain. Prosedur yang ditempuh oleh RWP dalam memulai pergerakan yaitu setiap node memilih tujuan secara acak. Selain itu, setiap node juga memilih kecepatan pergerakan secara acak dari interval $[Speedmin, \dots, Speedmax]$. Setelah node sampai di tujuan, setiap node akan berhenti sejenak sesuai dengan pause time

yang telah ditentukan sebelum kembali bergerak menggunakan prosedur yang sama (Laqtib, et al., 2016). Pola pergerakan dari setiap node yang menggunakan model mobilitas RWP cenderung berkumpul pada satu area simulasi. Yang dapat menyebabkan terjadinya density wave atau penumpukan node di satu area simulasi. Pergerakan node menggunakan RWP ditunjukkan pada Gambar 5.



Gambar 5. Pergerakan Random Waypoint

6.4 Prosedur Percobaan

A. Simulasi dengan 2D environment

1. Buat movement di bonnmotion berupa Random Waypoint Mobility (RWP)

```
# cd /home/zenhadi/bonnmotion/bonnmotion-3.0.1/bin
# ./bm -f rwp1 RandomWaypoint -n 100 -d 1000 -x 100 -y 100 -h 2 -p 0 -R 0
# ./bm NSFile -f rwp1
```

Akan terdapat 2 file:

- a. rwp1.ns_params
- b. rwp1.ns_movements

2. Copy-kan kedua file tersebut di folder berikut:

```
std::string traceFile = "src/mobility/examples/bonnmotion.ns_movements";
# cp rwp1* /home/zenhadi/ns-allinone-3.29/ns-3.29/src/mobility/examples/
```

3. Modifikasi file *manet-routing-compare.cc* sebagaimana pada praktikum 4 MANET pada direktori *ns-allinone-3.25/ns-3.25/scratch* melalui terminal

- a. Modifikasi pada bagian pembuatan node, sesuaikan jumlah node dengan jumlah node waktu dibuat di bonnmotion.

```

void
RoutingExperiment::Run (int nSinks, double txp, std::string CSVfileName)
{
    Packet::EnablePrinting ();
    m_nSinks = nSinks;
    m_txp = txp;
    m_CSVfileName = CSVfileName;

    int nWifis = 50;

    //double TotalTime = 200.0; => original
    double TotalTime = 120.0;
    std::string rate ("2048bps");
    std::string phyMode ("DsssRate11Mbps");
    std::string tr_name ("manet-routing-compare");
    int nodeSpeed = 20; //in m/s
    int nodePause = 0; //in s
    m_protocolName = "protocol";

    Config::SetDefault ("ns3::OnOffApplication::PacketSize",StringValue ("64"));
    Config::SetDefault ("ns3::OnOffApplication::DataRate", StringValue (rate));

    //Set Non-unicastMode rate to unicast mode
    Config::SetDefault ("ns3::WifiRemoteStationManager::NonUnicastMode",StringValue (phyMode));

    //NodeContainer adhocNodes;
    //adhocNodes.Create (nWifis); ✓

    //change parameter
    NodeContainer adhocNodes;
    adhocNodes.Create (100); ✓

```

- b. Modifikasi pada bagian mobility dimana file hasil bonnmotion diintegrasikan ke file *manet-routing-compare.cc*

```

wifiMac.SetType ("ns3::AdhocWifiMac");|
NetDeviceContainer adhocDevices = wifi.Install (wifiPhy, wifiMac, adhocNodes);

//MobilityHelper mobilityAdhoc;
//int64_t streamIndex = 0; // used to get consistent mobility across scenarios

//ObjectFactory pos;
//pos.SetTypeId ("ns3::RandomRectanglePositionAllocator");
//pos.Set ("X", StringValue ("ns3::UniformRandomVariable[Min=0.0|Max=300.0]"));
//pos.Set ("Y", StringValue ("ns3::UniformRandomVariable[Min=0.0|Max=1500.0]"));

//Ptr<PositionAllocator> taPositionAlloc = pos.Create ()->GetObject<PositionAllocator> ();
//streamIndex += taPositionAlloc->AssignStreams (streamIndex);

//std::stringstream ssSpeed;
//ssSpeed << "ns3::UniformRandomVariable[Min=0.0|Max=" << nodeSpeed << "]";
//std::stringstream ssPause;
//ssPause << "ns3::ConstantRandomVariable[Constant=" << nodePause << "]";
//mobilityAdhoc.SetMobilityModel ("ns3::RandomWaypointMobilityModel",
//                                "Speed", StringValue (ssSpeed.str ()),
//                                "Pause", StringValue (ssPause.str ()),
//                                "PositionAllocator", PointerValue (taPositionAlloc));
//mobilityAdhoc.SetPositionAllocator (taPositionAlloc);

//mobilityAdhoc.Install (adhocNodes);
//streamIndex += mobilityAdhoc.AssignStreams (adhocNodes, streamIndex);
//NS_UNUSED (streamIndex); // From this point, streamIndex is unused

std::string traceFile = "src/mobility/examples/rwpl.ns_movements";
Ns2MobilityHelper mobilityAdhoc = Ns2MobilityHelper (traceFile); ✓
mobilityAdhoc.Install ();

```

- c. Protokol routing yang digunakan adalah: AODV, DSDV, OLSR.
 - d. Analisa QoS melalui file: bacaCSV.py dan flowmon-parse-results.py untuk mengetahui throughput, delay, packet loss dan packet delivery ratio (PDR).
 - e. Lakukan perubahan di kecepatan node di point 1 dengan 5 m/s dan 10 m/s.
 - f. Ulangi langkah 3.c.
4. Lakukan hal yang sama pada tipe mobility lainnya yaitu Gauss Markov dan RPGM
- a. Buat dengan 3 kecepatan yang berbeda: 2 m/s, 5 m/s, dan 10 m/s.
 - b. Parameter yang lainnya sama:
 - i. Jumlah node : 100
 - ii. Area : 100m x 100m
 - iii. Durasi : 1000 second
 - iv. Pause time : 0 second
 - v. Jumlah koneksi : 10 (antara adhoc node dengan sink node).
5. Buat grafik perbandingan pada ketiga protokol yaitu AODV, DSDV dan OLSR dengan 3 perubahan kecepatan.

COMMAND:

RWP: ./bm -f rwp1 RandomWaypoint -n 100 -d 1000 -x 100 -y 100 -h 5 -p 0 -R 0
dimana

h maxspeed
p pause
R to get the same random

GaussMarkov : ./bm -f gmv1 GaussMarkov -n 100 -d 1000 -x 100 -y 100 -h 2 -p 0

RPGM : ./bm -f rpgm1 RPGM -n 100 -d 1000 -x 100 -y 100 -a 10 -h 2 -p 0 -c 0 -r 4

B. Simulasi dengan 3D Environment => Flying Adhoc Networks

1. Sumbu X, Y dan Z
Sumbu X: (-100m, 100m)
Sumbu Y: (-100m, 100m)
Sumbu Z: (0m, 100m)

2. Buat file fanet.py sebagai berikut:

```
#include "ns3/point-to-point-module.h"  
#include "ns3/ipv4-global-routing-helper.h"  
#include <fstream>  
#include <string>
```

```

#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/applications-module.h"
#include "ns3/mobility-module.h"
#include "ns3/config-store-module.h"
#include "ns3/wifi-module.h"

#include "ns3/aodv-helper.h"
#include "ns3/internet-module.h"
#include "ns3/netanim-module.h"

using namespace ns3;
NS_LOG_COMPONENT_DEFINE ("Mob");

int main (int argc, char *argv[])
{
    CommandLine cmd;
    cmd.Parse (argc, argv);

    NodeContainer c;
    c.Create (20); //20 wireless nodes

    WifiHelper wifi;
    wifi.SetStandard (WIFI_PHY_STANDARD_80211b);
    //80211a, 80211b, 80211n, 2.4g and 5G, 80211ac, 80211ax is also
    supported.80211p (VANETs, WAVE)

    WifiMacHelper mac;
    mac.SetType ("ns3::AdhocWifiMac");
    //AdhocWifiMac, StaWifiMac, ApWifiMac
    wifi.SetRemoteStationManager ("ns3::ConstantRateWifiManager",
                                  "DataMode", StringValue ("OfdmRate54Mbps"));
    YansWifiPhyHelper wifiPhy = YansWifiPhyHelper::Default ();
    YansWifiChannelHelper wifiChannel = YansWifiChannelHelper::Default ();
    wifiChannel.SetPropagationDelay
    ("ns3::ConstantSpeedPropagationDelayModel");
    wifiChannel.AddPropagationLoss ("ns3::FriisPropagationLossModel");
    wifiPhy.SetChannel (wifiChannel.Create ());
    NetDeviceContainer cDevices = wifi.Install (wifiPhy, mac, c);
    //
    NS_LOG_INFO ("Enabling AODV routing on all backbone nodes");
    AodvHelper aodv;
    //AODV protocol is being using FANETs.
    InternetStackHelper internet;
    internet.SetRoutingHelper (aodv); // has effect on the next Install ()
    internet.Install (c);

    //
    // Assign IPv4 addresses to the device drivers (actually to the associated
    // IPv4 interfaces) we just created.
    //
    Ipv4AddressHelper ipAddr;
    ipAddr.SetBase ("192.168.0.0", "255.255.255.0");
    Ipv4InterfaceContainer cInterfaces;
    cInterfaces=ipAddr.Assign (cDevices);

    /*

```

```

//Mobility Model - 2D
MobilityHelper mobility;

mobility.SetPositionAllocator ("ns3::GridPositionAllocator",
    "MinX", DoubleValue (0.0),
    "MinY", DoubleValue (0.0),
    "DeltaX", DoubleValue (5.0),
    "DeltaY", DoubleValue (10.0),
    "GridWidth", UIntegerValue (3),
    "LayoutType", StringValue ("RowFirst"));

mobility.SetMobilityModel ("ns3::RandomWalk2dMobilityModel", "Bounds",
    RectangleValue (Rectangle (-100, 100, -100, 100)));
mobility.Install (c);
*/

//Mobility Model -3D
MobilityHelper mobility;
mobility.SetMobilityModel ("ns3::GaussMarkovMobilityModel",
    "Bounds", BoxValue (Box (0, 100, 0, 100, 0, 100)),
    "TimeStep", TimeValue (Seconds (0.5)),
    "Alpha", DoubleValue (0.85),
    "MeanVelocity", StringValue
("ns3::UniformRandomVariable [Min=800|Max=1200]"),
    "MeanDirection", StringValue
("ns3::UniformRandomVariable [Min=0|Max=6.283185307]"),
    "MeanPitch", StringValue ("ns3::UniformRandomVariable [Min=0.05|Max=0.05]"),
    "NormalVelocity", StringValue
("ns3::NormalRandomVariable [Mean=0.0|Variance=0.0|Bound=0.0]"),
    "NormalDirection", StringValue
("ns3::NormalRandomVariable [Mean=0.0|Variance=0.2|Bound=0.4]"),
    "NormalPitch", StringValue
("ns3::NormalRandomVariable [Mean=0.0|Variance=0.02|Bound=0.04]");
mobility.SetPositionAllocator ("ns3::RandomBoxPositionAllocator",
    "X", StringValue ("ns3::UniformRandomVariable [Min=0|Max=100]"),
    "Y", StringValue ("ns3::UniformRandomVariable [Min=0|Max=100]"),
    "Z", StringValue ("ns3::UniformRandomVariable [Min=0|Max=100]");
mobility.Install (c);

UdpEchoServerHelper echoServer (9);

ApplicationContainer serverApps = echoServer.Install (c.Get(0));
serverApps.Start (Seconds (1.0));
serverApps.Stop (Seconds (10.0));

UdpEchoClientHelper echoClient (cInterfaces.GetAddress(0), 9);
echoClient.SetAttribute ("MaxPackets", UIntegerValue (1));
echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
echoClient.SetAttribute ("PacketSize", UIntegerValue (1024));

ApplicationContainer clientApps = echoClient.Install (c.Get(1));
clientApps.Start (Seconds (2.0));
clientApps.Stop (Seconds (10.0));

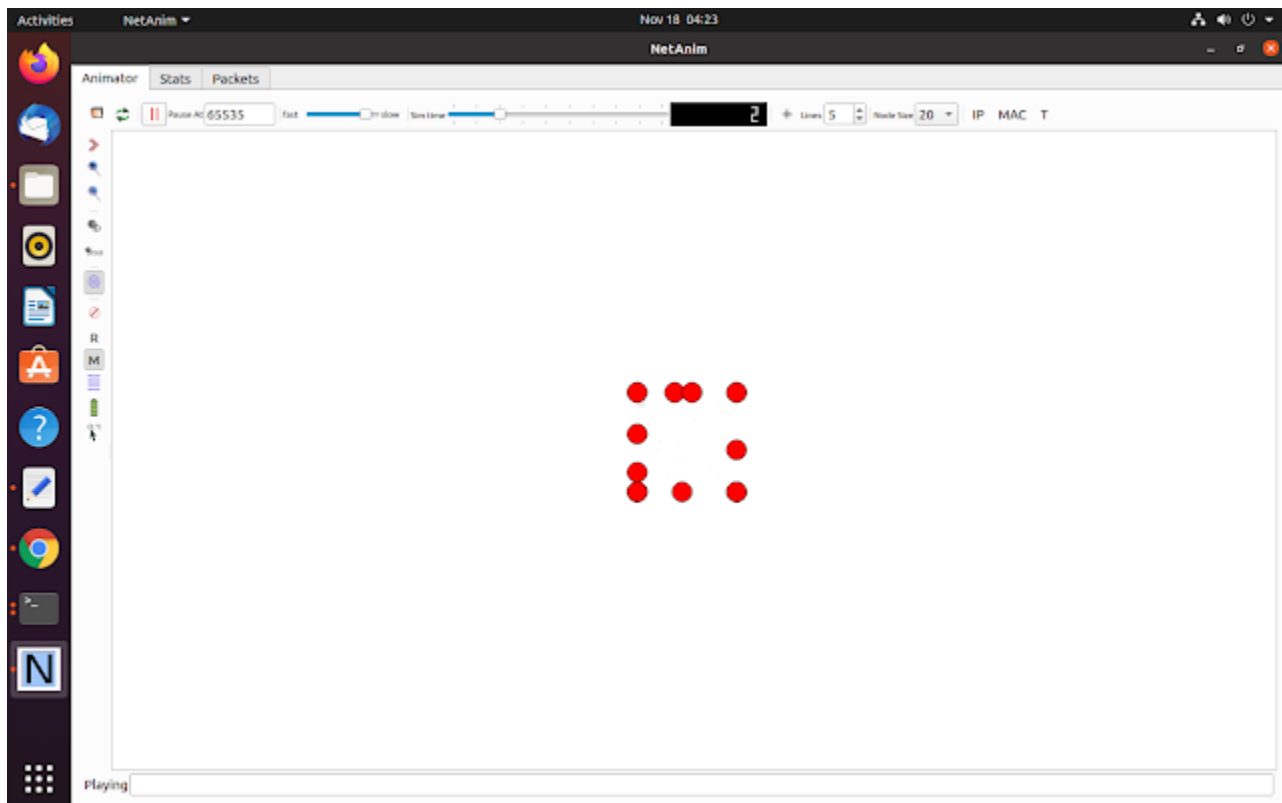
wifiPhy.EnablePcapAll ("Fanet3D"); //Packet Capture.
//Network Animation using NetAnim.
AnimationInterface anim("Fanet3D.xml");

```



```
//Ascii Trace Metrics can be processed using Tracemetrics Software.  
AsciiTraceHelper ascii;  
wifiPhy.EnableAsciiAll(ascii.CreateFileStream("Fanet3D.tr"));  
  
    Simulator::Stop (Seconds (10.0));  
    Simulator::Run ();  
    Simulator::Destroy ();  
    return 0;  
}
```

3. Salin file tersebut di folder `~ns-3.29/scratch`
4. Jalankan file tersebut
`# ./waf --run scratch/fanet`
5. Untuk melihat animasi
`# cd ../netanim-3.108/`
`# ./NetAnim`





Referensi:

1. Laisa RPM, Rakhmadhany P, Achmad B, Pengaruh Model Mobilitas Node pada Protokol Routing AODV dalam MANET, Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Vol. 3, No. 1, Januari 2019, hal. 563-572
2. <https://www.nsnam.com/2020/11/flying-adhoc-network-simulation-fanet.html>