

Laboratorijska vežbe iz predmeta Principi modernih telekomunikacija

Vežba: Simulacija telekomunikacionih mreža korišćenjem programa ns-3

Ova vežba se sastoji od formiranja IP mreže koja se sastoji od bežičnog i žičnog dela, kao i generisanja i praćenja saobraćaja u takvoj mreži. Mreža se sastoji od tri uređaja povezana pomoću dve veze, pri čemu je jedna veza bežična a druga žična.

Deo 1: Formiranje bežične mreže sa dva uređaja

Osnova za rad na ovoj vežbi je primer simulacije bežične mreže sa jednom stanicom i jednom pristupnom tačkom.

```
#include "ns3/core-module.h"
#include "ns3/internet-module.h"
#include "ns3/network-module.h"
#include "ns3/applications-module.h"
#include "ns3/wifi-module.h"
#include "ns3/mobility-module.h"
#include "ns3/netanim-module.h"

using namespace ns3;

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

    NodeContainer nodes;
    nodes.Create(2);

    Ptr<Node> sta = nodes.Get(0);
    Ptr<Node> ap = nodes.Get(1);

    YansWifiChannelHelper channel = YansWifiChannelHelper::Default ();
    YansWifiPhyHelper phy = YansWifiPhyHelper::Default ();
    phy.SetChannel (channel.Create ());

    WifiHelper wifi;
    wifi.SetStandard (WIFI_PHY_STANDARD_80211a);
    WifiMacHelper mac;

    Ssid ssid ("wifi_mreza");

    mac.SetType ("ns3::StaWifiMac",
                "Ssid", SsidValue (ssid),
                "ActiveProbing", BooleanValue (false));

    NetDeviceContainer staDevice = wifi.Install (phy, mac, sta);

    mac.SetType ("ns3::ApWifiMac",
                "Ssid", SsidValue (ssid));
    NetDeviceContainer apDevice = wifi.Install (phy, mac, ap);
```

```

MobilityHelper mobility;

mobility.SetPositionAllocator ("ns3::GridPositionAllocator",
    "MinX", DoubleValue (50.0),
    "MinY", DoubleValue (100.0),
    "DeltaX", DoubleValue (15.0),
    "DeltaY", DoubleValue (30.0),
    "GridWidth", UIntegerValue (3),
    "LayoutType", StringValue ("RowFirst"));

mobility.SetMobilityModel ("ns3::RandomWalk2dMobilityModel",
    "Bounds", RectangleValue (Rectangle (-50, 200, -50, 200)));
mobility.Install (sta);

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

InternetStackHelper stack;
stack.Install (nodes);

Ipv4AddressHelper address;

address.SetBase ("192.168.0.0", "255.255.255.0");
Ipv4InterfaceContainer staIf;
Ipv4InterfaceContainer apIf;
staIf = address.Assign (staDevice);
apIf = address.Assign (apDevice);

UdpClientHelper client(apIf.GetAddress(0), 10001);
client.SetAttribute("MaxPackets", UIntegerValue(30));
client.SetAttribute("Interval", TimeValue(Seconds(0.2)));
client.SetAttribute("PacketSize", UIntegerValue(512));

ApplicationContainer clientApps = client.Install(sta);
clientApps.Start(Seconds(1.0));
clientApps.Stop(Seconds(7.0));

UdpServerHelper server(10001);

ApplicationContainer serverApps = server.Install(ap);
serverApps.Start(Seconds(0.0));
serverApps.Stop(Seconds(8.0));

AsciiTraceHelper ascii;
phy.EnableAsciiAll(ascii.CreateFileStream("primer_wifi.tr"));
phy.EnablePcapAll("primer_wifi");

AnimationInterface anim("primer_wifi.xml");

Simulator::Stop (Seconds (10.0));
Simulator::Run ();
Simulator::Destroy ();

return 0;
}

```

Potrebno je formirati fajl wifi_dva_cvora.cc koji sadrži navedeni kod.

U okviru simulacije se prenosi UDP tok između WiFi stanice i pristupne tačke.

U snimljenom pcap fajlu prepoznati UDP tok i zabeležiti vreme slanja prvog UDP paketa i poslednjeg UDP paketa. Nacrtati grafik protoka UDP paketa u Wireshatk-u i priložiti ga u izveštaj.

Potrebno je pronaći zapise o prenosu prvog i poslednjeg paketa iz tekstualnog trace fajla i kopirati ih u izveštaj.

Potrebno je pregledati animaciju simulacije korišćenjem NetAnim vizuelizacionog alata kao i korišćenjem PyVis programa i u izveštaj upisati zapažanja.

Deo 2: Dodavanje žične mreže

Potrebno je dodati novi uređaj i WiFi pristupnu tačku žičnom vezom povezati sa tim uređajem. To je moguće uraditi sledećim programskim instrukcijama:

- dodati include:

```
#include "ns3/point-to-point-module.h"
```

- red:

```
nodes.Create(2);
```

promeniti u:

```
nodes.Create(3);
```

- dodatne instrukcije, pre instanciranja aplikacije:

```
Ipv4AddressHelper addressWired;  
addressWired.SetBase ("192.170.0.0", "255.255.255.0");  
NodeContainer wiredNodes = NodeContainer( nodes.Get(1), nodes.Get(2));  
PointToPointHelper p2p;  
// zadajemo brzinu linka 5Mb/s  
p2p.SetDeviceAttribute("DataRate", StringValue("5Mbps"));  
// zadajemo vreme propagacije po linku 5ms  
p2p.SetChannelAttribute("Delay", StringValue("5ms"));  
// instaliramo link  
NetDeviceContainer wiredDevices = p2p.Install(wiredNodes);  
Ipv4InterfaceContainer wiredInterfaces = addressWired.Assign(wiredDevices);  
// interfejs novog uređaja (wired device)  
Ptr<Node> wd = nodes.Get(2);  
mobility.Install (wd);  
Ipv4GlobalRoutingHelper::PopulateRoutingTables();
```

- ispod deklaracije AsciiTraceHelpera dodati linije:

```
p2p.EnableAsciiAll(ascii.CreateFileStream("primer_wired.tr"));  
p2p.EnablePcapAll("primer_wired");
```

Zadac u ovom delu vežbi je da se ove instrukcije uključe u simulacioni kod. Zatim je potrebno promeniti UDP tok u simulaciji, tako da krajnji uređaji toka budu novi čvor i WiFi stanica. Programski kod tako dobijene simulacije je potrebno dodati u izveštaj. Takođe, potrebno je u tekstualnom trace fajlu pronaći linije koje opisuju prvi i poslednji paket UDP toka i uneti ih u izveštaj.

Deo 3: Generisanje TCP saobraćaja

UDP aplikaciju zameniti OnOff aplikacijom koja generiše TCP saobraćaj. Nacrtati u Wireshark-u protok TCP paketa i priložiti ga u izveštaj.