

Hands-On 6

O seventh além de apresentar suporte ao endereçamento Ipv6, também está preparado para gerar arquivos de suporte ao Gnuplot.

A primeira coisa que é adicionada é uma opção de fazer o endereçamento por Ipv6, sendo selecionado por um argumento na linha de comando:

```
bool useV6 = false;

CommandLine cmd ( __FILE__ );
cmd.AddValue ( "useIpv6", "Use Ipv6", useV6 );
cmd.Parse ( argc, argv );
```

Aqui podemos observar que o usuário pode especificar a opção useIpv6 para trocar o valor da variável useV6 e consequentemente trocar o tipo de endereçamento.

Esse boolean é utilizado depois em um trecho de código que efetivamente faz seleção e configuração do método de endereçamento:

```
if (useV6 == false)
{
    Ipv4AddressHelper address;
    address.SetBase ("10.1.1.0", "255.255.255.0");
    Ipv4InterfaceContainer interfaces = address.Assign (devices);
    sinkAddress = InetSocketAddress (interfaces.GetAddress (1), sinkPort);
    anyAddress = InetSocketAddress (Ipv4Address::GetAny (), sinkPort);
    probeType = "ns3::Ipv4PacketProbe";
    tracePath = "/NodeList/*/ns3::Ipv4L3Protocol/Tx";
}
else
{
    Ipv6AddressHelper address;
    address.SetBase ("2001:0000:f00d:cafe::", Ipv6Prefix (64));
    Ipv6InterfaceContainer interfaces = address.Assign (devices);
    sinkAddress = Inet6SocketAddress (interfaces.GetAddress (1,1), sinkPort);
    anyAddress = Inet6SocketAddress (Ipv6Address::GetAny (), sinkPort);
    probeType = "ns3::Ipv6PacketProbe";
    tracePath = "/NodeList/*/ns3::Ipv6L3Protocol/Tx";
}
```

Outra funcionalidade que é implementada é a de suporte a geração automática de gráficos com o Gnuplot, a qual utiliza um helper para configurar os parâmetros do gráfico:

```
GnuplotHelper plotHelper;

plotHelper.ConfigurePlot ("seventh-packet-byte-count",
                          "Packet Byte Count vs. Time",
                          "Time (Seconds)",
                          "Packet Byte Count");
```

Na função configurePlot é determinado respectivamente o nome do arquivo, o título do gráfico e as legendas do sentido x e y.

Feito isso, agora temos que configurar a fonte e tipo de dados que deve ser plotado:

```
plotHelper.PlotProbe (probeType,
                      tracePath,
                      "OutputBytes",
                      "Packet Byte Count",
                      GnuplotAggregator::KEY_BELOW);
```

Na função `PlotProbe` determinamos o tipo de probe e o caminho dos dados, ambos determinados em variáveis externas, também determinamos o endereço da configuração que determina como os dados devem ser coletados, o nome desse conjunto de dados e a posição dos rótulos dos gráficos.

Por fim temos que configurar o arquivo que vai receber os dados exportados da simulação:

```
FileHelper fileHelper;  
  
fileHelper.ConfigureFile ("seventh-packet-byte-count",  
                          FileAggregator::FORMATTED);
```