on-off-helper.cc

Go to the documentation of this file.

```
/* -*- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; -*- */
/*
 * Copyright (c) 2008 INRIA
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License version 2 as
 * published by the Free Software Foundation;
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
 *
 * Author: Mathieu Lacage <mathieu.lacage@sophia.inria.fr>
 */
#include "on-off-helper.h"
#include "ns3/inet-socket-address.h"
#include "ns3/packet-socket-address.h"
#include "ns3/string.h"
#include "ns3/data-rate.h"
#include "ns3/uinteger.h"
#include "ns3/names.h"
#include "ns3/random-variable-stream.h"
#include "ns3/onoff-application.h"

namespace ns3 {

OnOffHelper::OnOffHelper (std::string protocol, Address address)
{
  m_factory.SetTypeId ("ns3::OnOffApplication");
  m_factory.Set ("Protocol", StringValue (protocol));
  m_factory.Set ("Remote", AddressValue (address));
}

void
OnOffHelper::SetAttribute (std::string name, const AttributeValue &value)
{
  m_factory.Set (name, value);
}

ApplicationContainer
OnOffHelper::Install (Ptr<Node> node) const
{
  return ApplicationContainer (InstallPriv (node));
}

ApplicationContainer
OnOffHelper::Install (std::string nodeName) const
{
  Ptr<Node> node = Names::Find<Node> (nodeName);
  return ApplicationContainer (InstallPriv (node));
}

ApplicationContainer
OnOffHelper::Install (NodeContainer c) const
{
  ApplicationContainer apps;
  for (NodeContainer::Iterator i = c.Begin (); i != c.End (); ++i)
    {
      apps.Add (InstallPriv (*i));
    }

  return apps;
}

Ptr<Application>
OnOffHelper::InstallPriv (Ptr<Node> node) const
{
  Ptr<Application> app = m_factory.Create<Application> ();
  node->AddApplication (app);

  return app;
}

int64_t
OnOffHelper::AssignStreams (NodeContainer c, int64_t stream)
{
  int64_t currentStream = stream;
  Ptr<Node> node;
  for (NodeContainer::Iterator i = c.Begin (); i != c.End (); ++i)
    {
      node = (*i);
      for (uint32_t j = 0; j < node->GetNApplications (); j++)
        {
          Ptr<OnOffApplication> onoff = DynamicCast<OnOffApplication> (node->GetApplication (j));
          if (onoff)
            {
              currentStream += onoff->AssignStreams (currentStream);
            }
        }
    }
  return (currentStream - stream);
}

void
OnOffHelper::SetConstantRate (DataRate dataRate, uint32_t packetSize)
{
  m_factory.Set ("OnTime", StringValue ("ns3::ConstantRandomVariable[Constant=1000]"));
  m_factory.Set ("OffTime", StringValue ("ns3::ConstantRandomVariable[Constant=0]"));
  m_factory.Set ("DataRate", DataRateValue (dataRate));
  m_factory.Set ("PacketSize", UintegerValue (packetSize));
}

} // namespace ns3
```