网络论坛

NETWORK SIMULATOR(NS)概述

网络论坛
www.netforum.com.cn

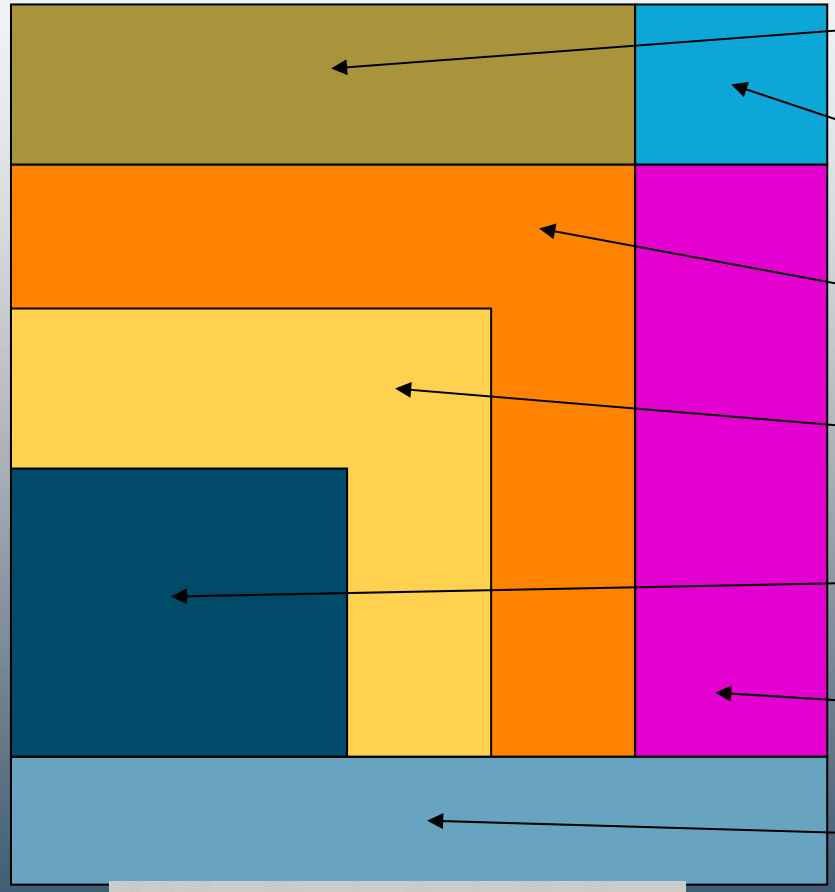
NS软件概述

网络论坛

- NS是一个由UC Berkeley开发的用于仿真各种IP网络的为主的优秀的仿真软件。
- 该软件的开发最初是针对基于UNIX系统下的网络设计和仿真而进行的。

NS的结构

网络论坛



OTcl：object-oriented Tcl

TclCL：C++ and OTcl linkage

Tcl与Tk编程环境

网络论坛

- Tcl与Tk是安装在Unix/Linux环境下的两个包，它们一起构成了一套开发系统应用程序和图形用户界面接口（GUI）应用程序的环境。
- Tcl的全称是Tool Command Language。
- Tk是Tcl在X Window环境下的扩展，它包含了Tcl的全部的C库函数，以及支持X Window的窗口、控件等C库函数，为用户开发图形用户界面提供了方便。
- Tcl是解释执行的脚本语言。它的实现依赖于Tcl内部的C函数库。添加新的C函数就可以扩充Tcl的命令和功能，是扩展性非常强的脚本程序设计语言。
- Tcl解释器把用户输入的命令和程序语句进行初步分析，然后调用C函数库里的相应函数来执行，输出结果。
- Tcl是无强制类型的脚本语言，一切变量，不论整型，浮点型等，都以字符串的形式存储。

Tcl的面向对象扩展——OTcl

网络论坛

- **Otcl**是**Tcl**的面向对象（**Object Oriented**）的扩展，在**Otcl**中加入了类的概念。
- 对象是类的实例，它有自己的属性（成员变量，**InstVar**）和自己的内部操作（成员函数，**InstProc**）。
- 对象具有继承、封装、多态性和动态绑定等性质。
- 面向对象机制的加入使得原始的**Tcl**变得更加强大，更加方便使用。
- 虽然和**C++**中对象和类以及其他面向对象程序设计语言中的概念相同，但是具体实现和语法却存在很大的差别。

C++和Otcl两种编程语言来实现NS

网络论坛

- 模拟器有两方面的事情需要做
- 一方面，具体协议的模拟和实现：需要一种程序设计语言，它需要很有效率的处理字节（**Byte**），报头（**Packet Header**）等信息，需要应用合适的算法在大量的数据集合上进行操作。为了实现这个任务，程序内部模块的运行速度（**run-time speed**）是非常重要的，而运行模拟环境的时间、寻找和修复**bug**的时间，重新编译和运行的时间（**run-around time**）就显得不是很重要了。
- 另一方面，许多网络中的研究工作都围绕着网络组件和环境的具体参数的设置和改变而进行的，需要在短时间内快速的开发和模拟出所需要的网络环境（**scenarios**），并且方便修改和发现、修复程序中的**Bug**。在这种任务中，**run-around time**就显得很重要了，因为模拟环境的建立和参数信息的配置只需要运行一次。

C++和Otcl两种编程语言来实现NS

网络论坛

- 为了满足以上两种不同任务的需要，NS的设计实现使用了两种程序设计语言，C++和Otcl。
- 这两种程序设计语言都是面向对象（object oriented）的程序设计语言 。
- C++程序模块的运行速度非常快，是强制类型的程序设计语言（变量严格定义整型，浮点型和字符、字符串类型），容易实现精确的、复杂的算法，但是修改和发现、修正bug所花费的时间要长一些，因为它比较复杂。该特性正好用于解决第一个方面的问题。
- Otcl是脚本程序编写语言，是无强制类型的，比较简单，容易实现和修改，容易发现和修正bug，虽然它的运行速度和C++的模块相比要慢很多。该特性正好用于解决第二方面的问题。

C++和Otcl两种编程语言来实现NS

网络论坛

- 最基本的，用**OTcl**来实现：

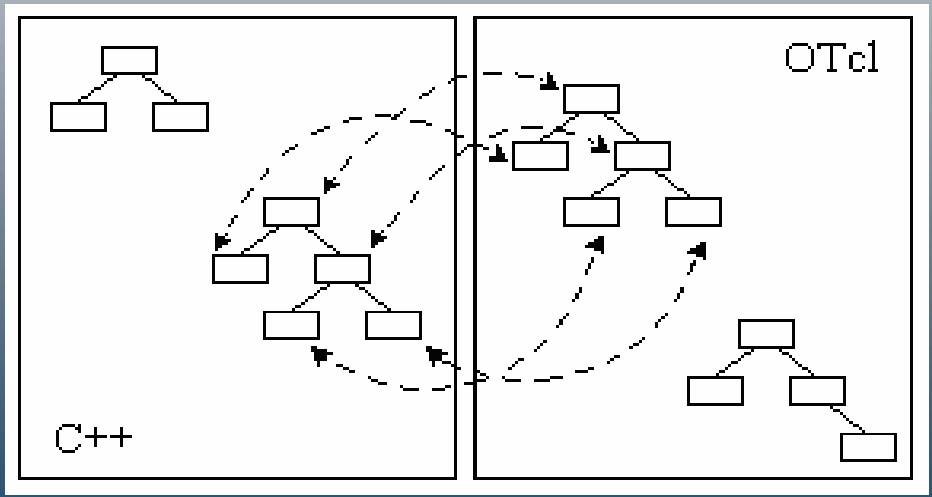
  对模拟环境的配置、建立和在模拟中只运行一次的地方。

  如果可能，用**Otcl**脚本来随意操作已经存在的**C++**对象。

- 最基本的，用**C++**来实现：

  如果你需要在模拟中对每一或大量数据包进行处理。

  如果你想修改已存在的**C++**对象的属性和功能。



C++和Otcl的连接 -- 类成员变量的绑定

网络论坛

- 变量绑定的目的，是用户通过修改和设置脚本中的Otcl对象中的成员变量时，该对象对应的“影子”C++对象中的成员变量也相应的变化，保持一直。变量绑定是TclObjet类中的成员函数。变量绑定在构造函数中完成。
- **ASRMAgent::ASRMAgent()**

```
{
        bind(“pdistance_”,&pdistance_);
        bind(“requestor_”,&requestor_);
        bind_time(“lastSent_”,&lastSessSent_”);
        bind_bw(“ctrlLimit_”,&ctrlBWLimit_);
        bind_bool(“running_”,&running_);
}
```

C++和Otcl的连接 --两种对象成员函数的对应

- 类似于变量绑定，Otcl对象中也要创建相应的成员函数，来达到调用对应的C++对象中的成员函数的目的。
- 这种让成员函数之间一一对应的机制，是通过在该对象中的command()函数来实现的。由于command()是TclObjet的成员函数，所以它的派生类每个对象都可以重写该函数。
- **Int ASRMAgent::command(int argc,const char* const* argv) {**

```
Tcl& tcl=Tcl::instance();
if(argc==3) {
        if(strcmp(argv[1],"distance?")==0) {……}
}
return (SRMAgent::command(argc,argv);    }
```

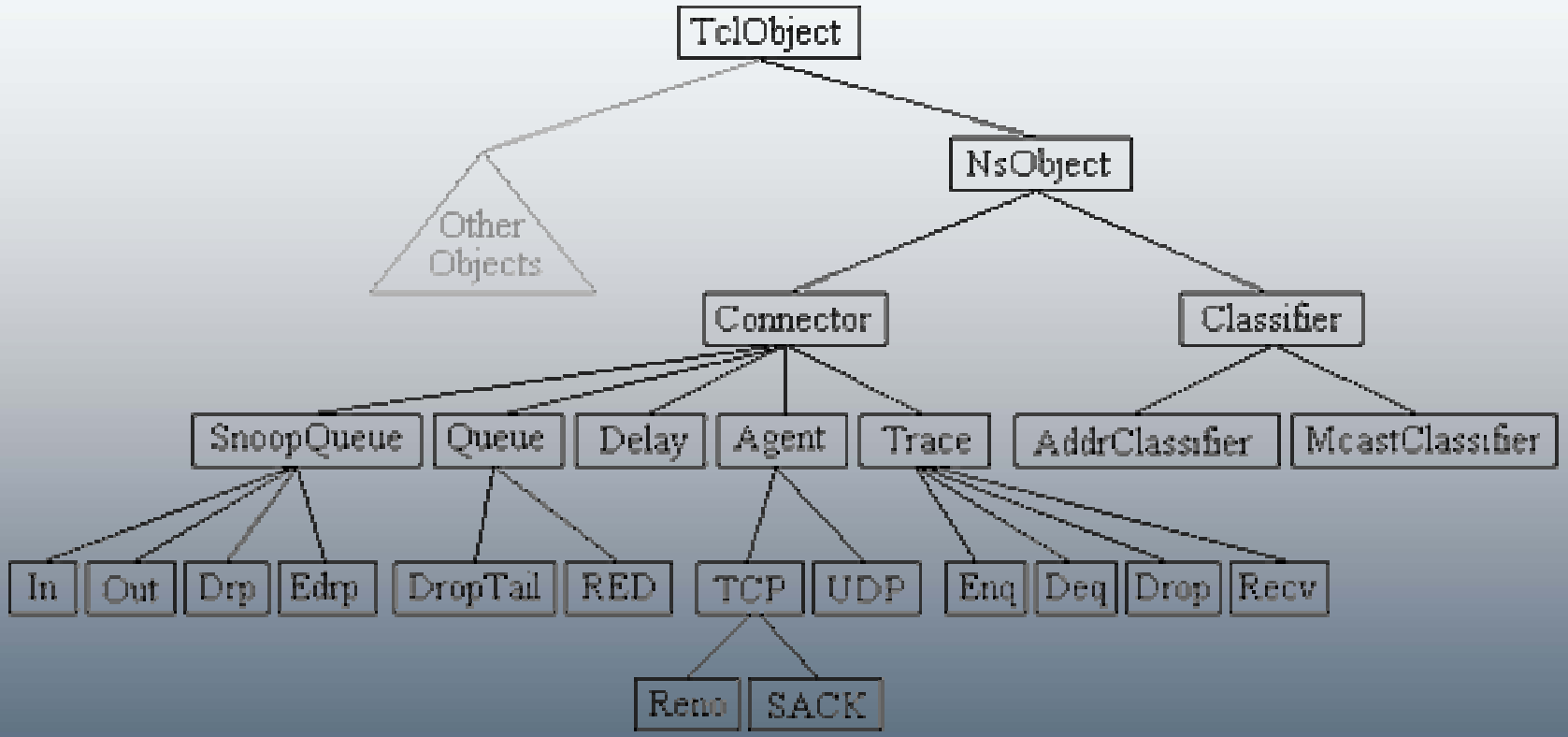
NS仿真原理--网络组件

网络论坛

- 网络组件分为简单网络组件和复合网络组件。
- **NsObject**是所有基本网络组件的父类，而它本身的父类是**TclObject**类。这个类的对象有一个基本的功能，就是处理数据包（**Packet**）。
- 所有的基本网络组件可以划分为**2**类，分类器（**Classifier**）和连接器（**Connector**）。它们都是**NsObject**的直接子类，也是所有基本网络组建的父类。
- 分类器（**Classifier**）的派生类组件对象包括地址分类器（**AddrClassifier**）和多播分类器（**McastClassifier**）等。
- 连接器（**Connector**）的派生类组件对象包括队列（**Queue**），延迟（**Delay**），各种代理（**Agent**），和追踪对象类（**Trace**）。

NS仿真原理-- 网络组件

网络论坛

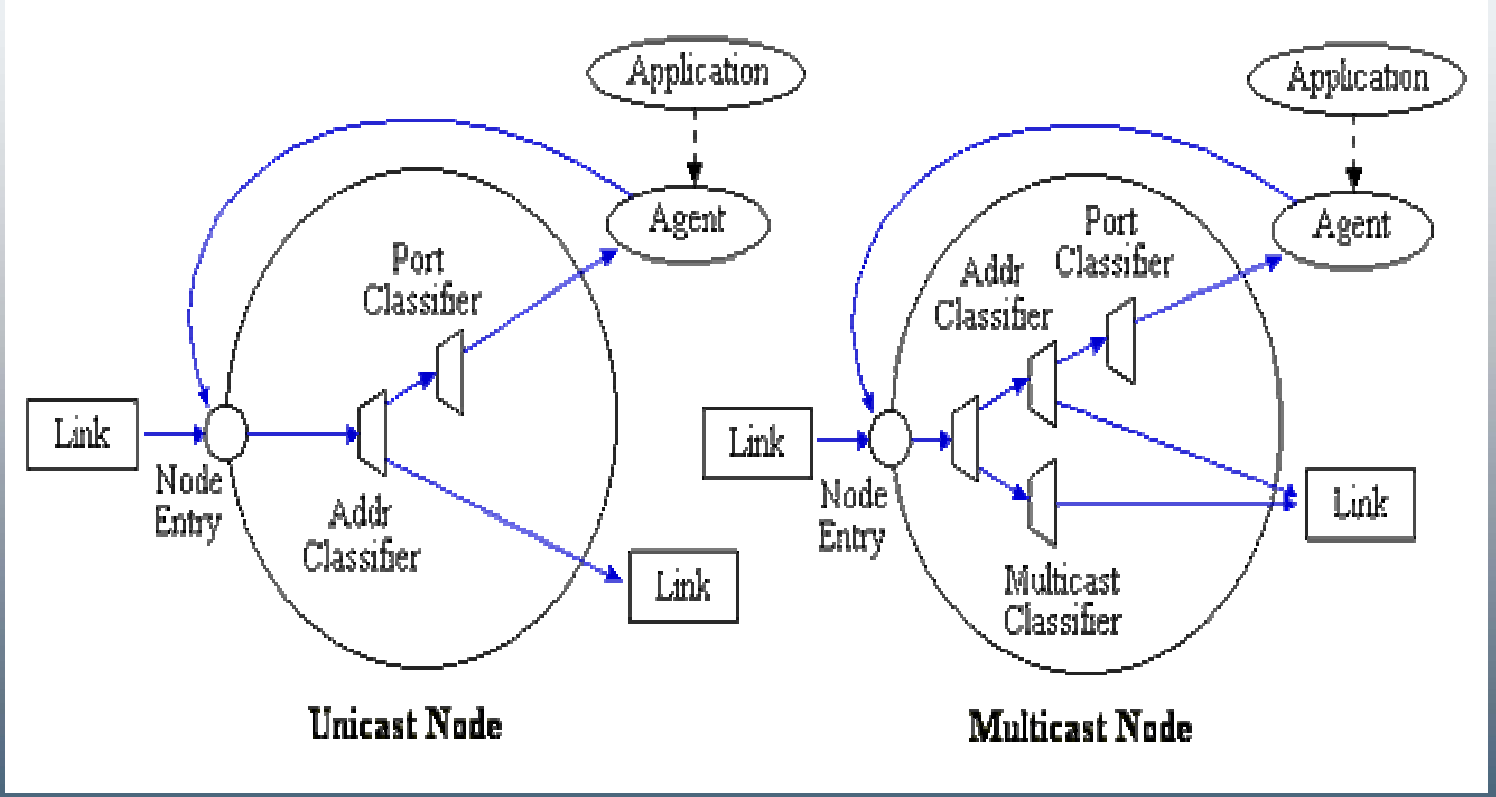


NS仿真原理--网络组件　分类器 (classifier)

- 分类器——**classifier**是**Ns-2**基本网络组件的一个大类。它的基本派生类有地址分类器（**AddrClassifier**）和多播分类器（**McastClassifier**）等。
- 分类器的特点，是基于分类器的基本网络组件具有**1**个或多个可能的数据输出路径，属于交换（**Switch**）设备（对应来说，连接器**Connector**只有一个数据的输出路径）。
- 复合组建（比如**Node**和**Link**），它们不是**TclObject**类的派生类，而是在**ns-2**中独立的类。它们属于复合组件
- 拓扑结点（**Node**）是由一个结点入口对象和若干个分类器（**Classifier**）组成的一个符合对象。

NS仿真原理--网络组件　分类器 (classifier)

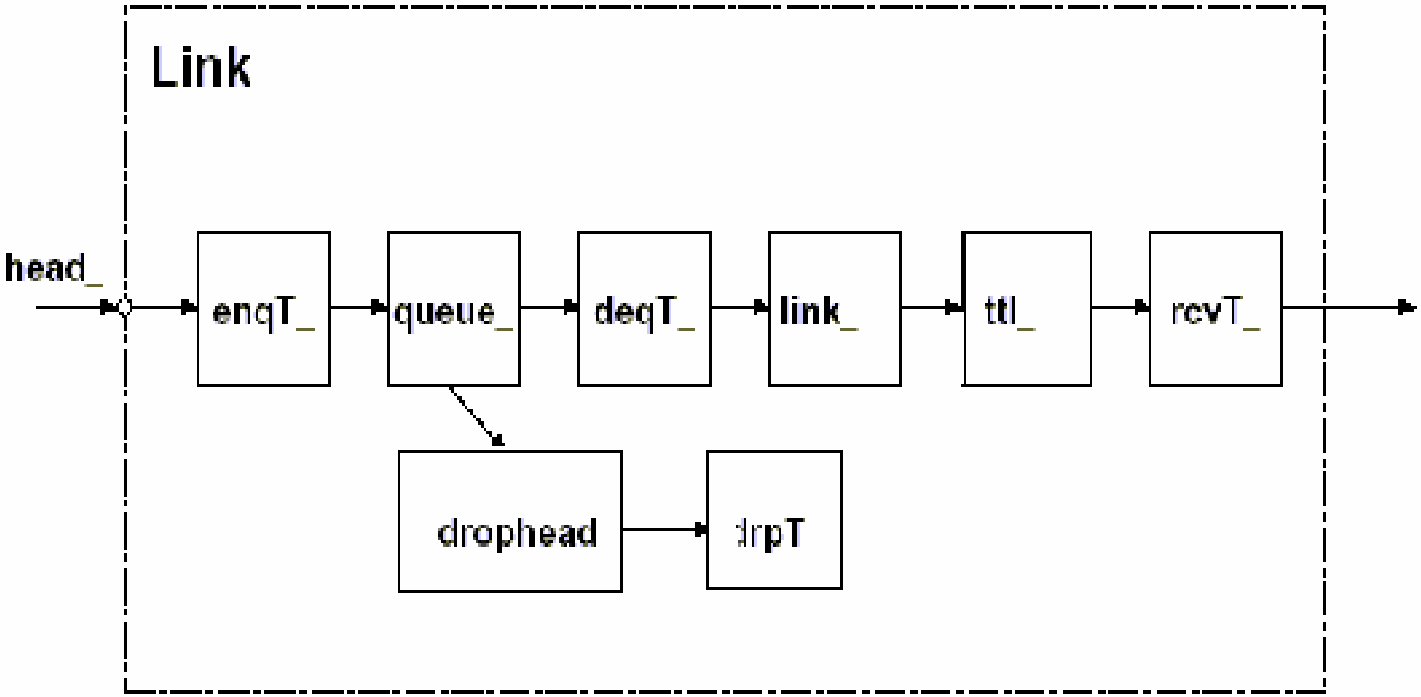
网络论坛



NS仿真原理--网络组件　连接器 (connector) 网络论坛

- 连接器——Connector是Ns-2基本网络组件中的另一个大类。它的基本派生类对象包括缓冲队列（Queue）、延迟（Delay）、各种产生和处理数据包的代理（Agent）和对象的跟踪器（Trace）。
- 连接器的特点，是基于连接器的基本网络组件只有一个可能数据输出的路径，即（单进单出的管道，所以叫Connector），和分类器有1个或多个可能的数据输出路径（Switch）是不同的。
- 拓扑结点连接类（Link）是Ns-2中另一个主要的复合组件对象。一个结点和另一个结点之间的简单连接（simplex-link）是单向的。一个最基本的简单连接由一个连接入口、包缓冲队列、延迟处理对象、废弃处理对象和时间处理对象（TTL）组成。
- 一个结点的输出队列，是通过和这个结点相连的Link中的缓冲队列来实现的。

NS仿真原理--网络组件　连接器 (connector) 网络论坛



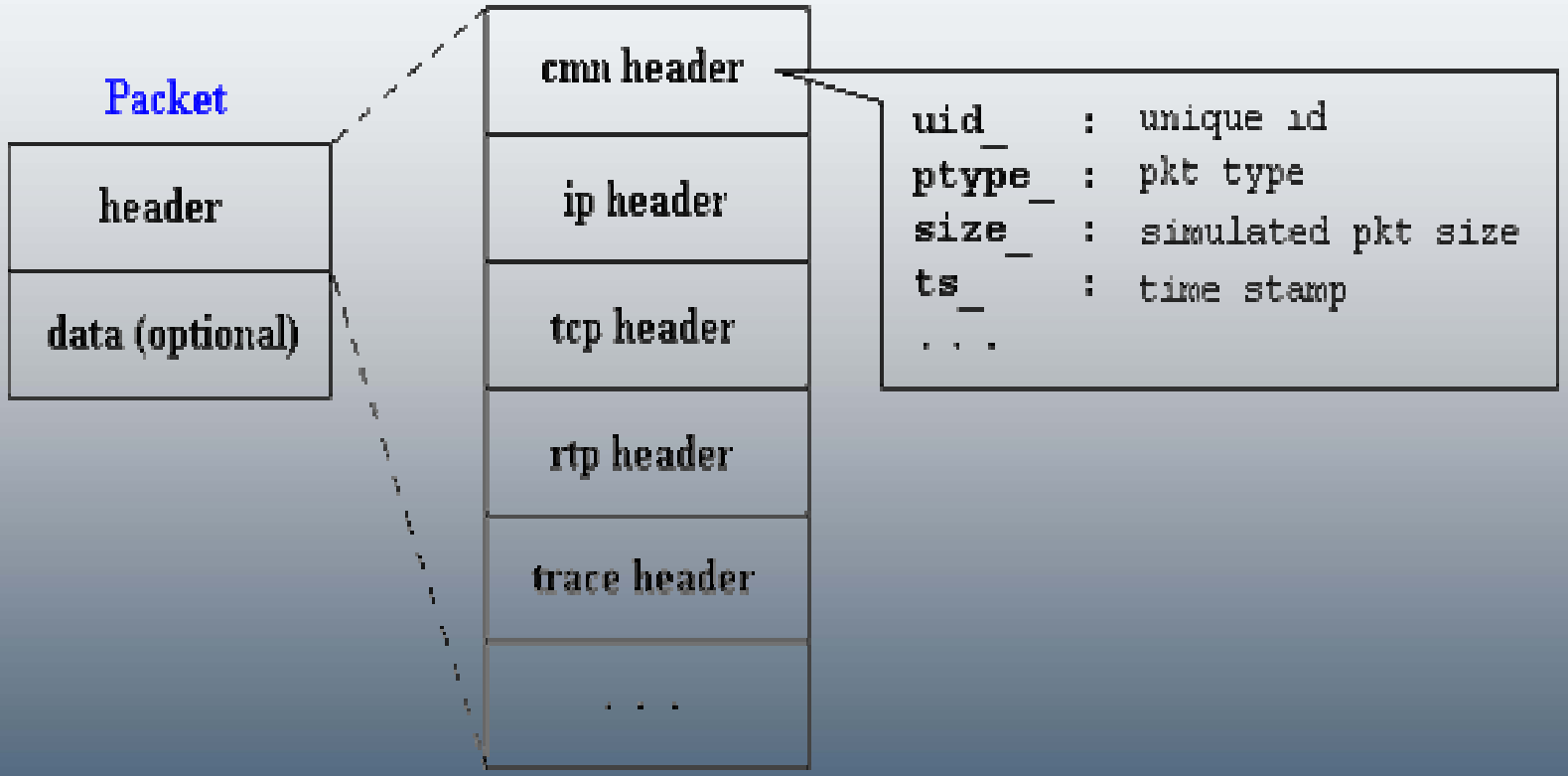
NS仿真原理--网络组件　包 (packet)

网络论坛

- 在**Ns-2**中，一个“包”是由一个报头（**Header**）堆栈（**Stack**）和一个可选择的数据空间构成。
- 报头的格式，在模拟器对象（**Simulator**）创建的过程中创建好，同时，所有注册过的报头，比如一般的属性说明、**IP**报头，**Tcp**报头，**RTP**报头，都在包的报头堆栈中存储。
- 通过不同报头在包的报头堆栈中的偏移量（**Offset**），网络组件可以根据需要，来访问包中的不同报头。这就符合上面的机制，在报头格式创建的时候，无论这个注册过的报头是否需要，都被加到报头堆栈中。
- 通常，一个包只含有报头堆栈部分，而没有数据空间。这是因为，在模拟的环境中，传输实际的数据是没有意义的。

NS仿真原理--网络组件　包 (packet)

网络论坛



NS仿真原理--网络组件　代理 (Agent)

网络论坛

- 代理网络组件代表了在网络层中数据包的产生地和运输源头，同时也是各层网络中各种协议的实现。
- **Agent/TCP**
- **Agent/TCP/Newreno**
- **Agent/TCP/Vegas**
- **Agent/TCP/Reno**
- **Agent/TCP/Sack1**
- **Agent /TCP/Fack**

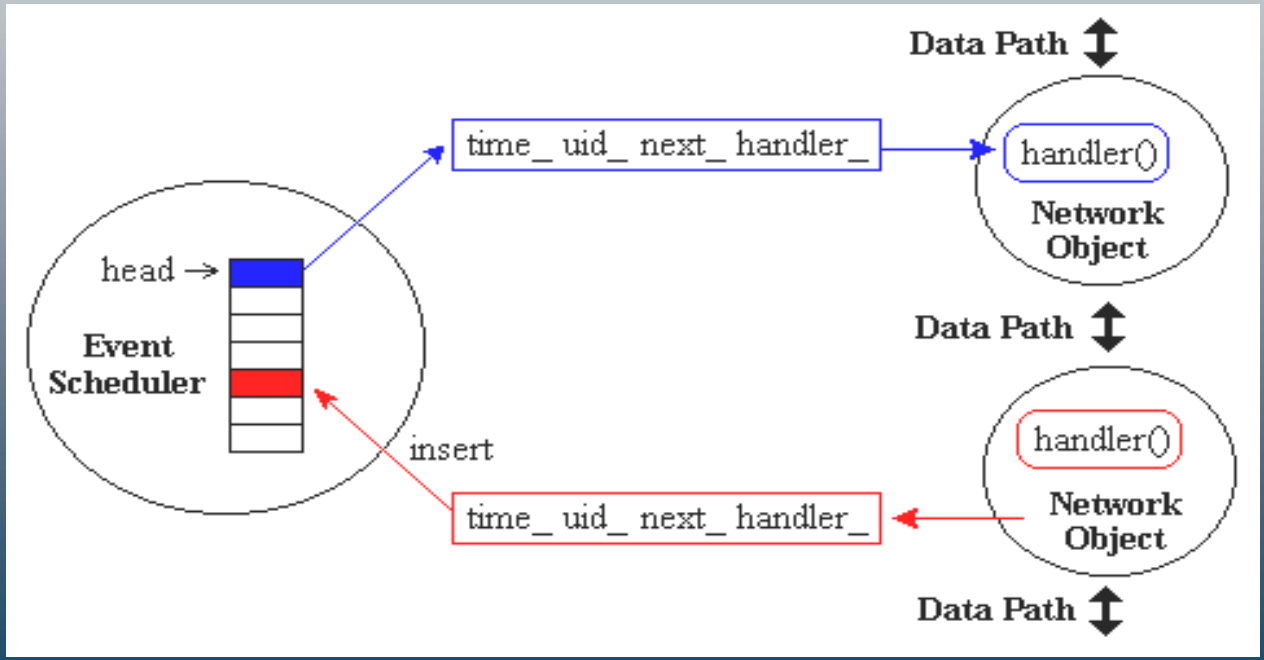
NS仿真原理--网络组件　应用层(Application) 网络论坛

- 应用程序层是建立在传输代理上的应用程序的模拟。
- **Ns-2**中有两种类型的**“**应用程序**”**，数据源发生器（**Traffic Generator**）和模拟的应用程序（**Simulated applications**）。

NS仿真原理--事件调度器

网络论坛

- NS是离散事件驱动的网络仿真器。它使用Event Scheduler对所有组件希望完成的工作和计划该工作发生的时间进行列表和维护。如下图所示:Event Scheduler主要是一个具有优先级的队列，它按照事件发生的时间对其中的工作排序，并遵循这样的顺序执行工作。而各个组件之间的通信是依靠传递数据包的方式来实现的



NS运行环境和使用方式

网络论坛

- 平台：**Windows、Linux、Unix、machitosh**，还要求系统装有**C++**编译器
- 两种语言：**C++、OTcl**

- 使用方式**--**命令行方式：敲入一个命令，返回一个结果

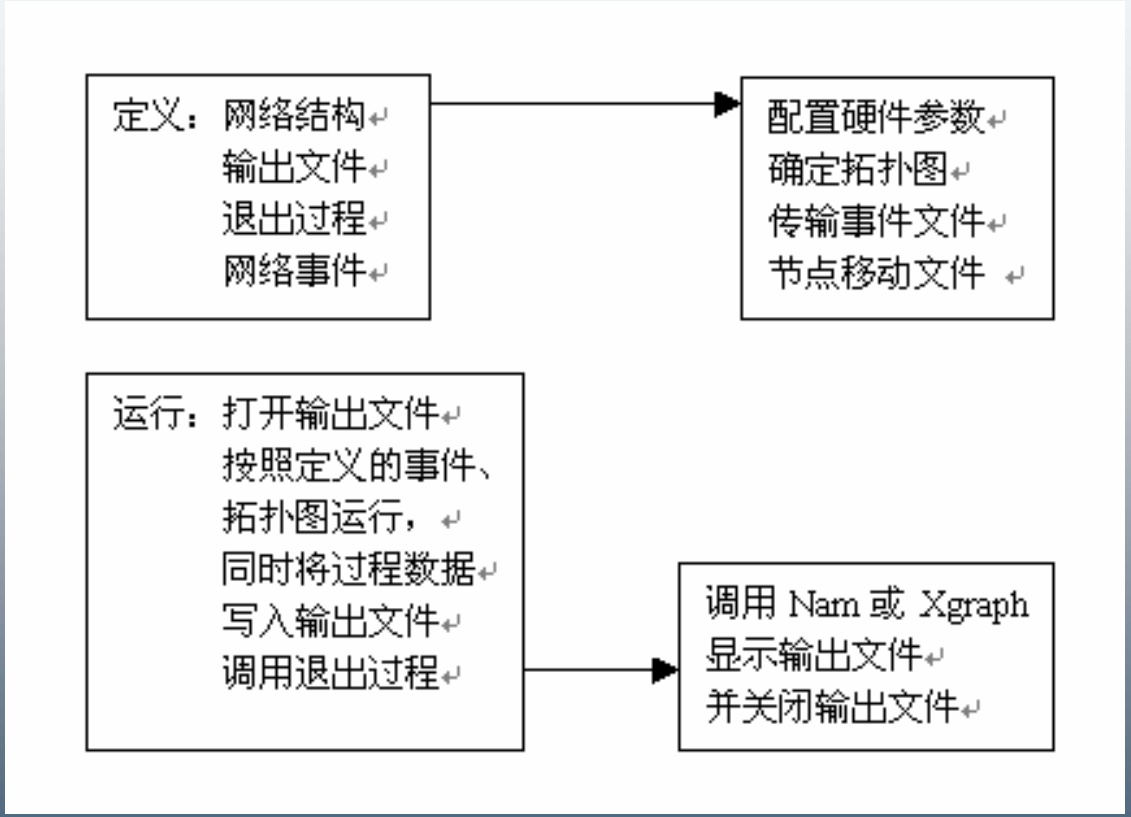
  输入命令：　　**%/ns-version**

  返回版本号：　**ns /2.0a12**
- 使用方式**--**脚本方式：指定一个脚本文件（***.tcl**文件），让**NS**执行

  输入命令：　　**%/ns <example.tcl>**

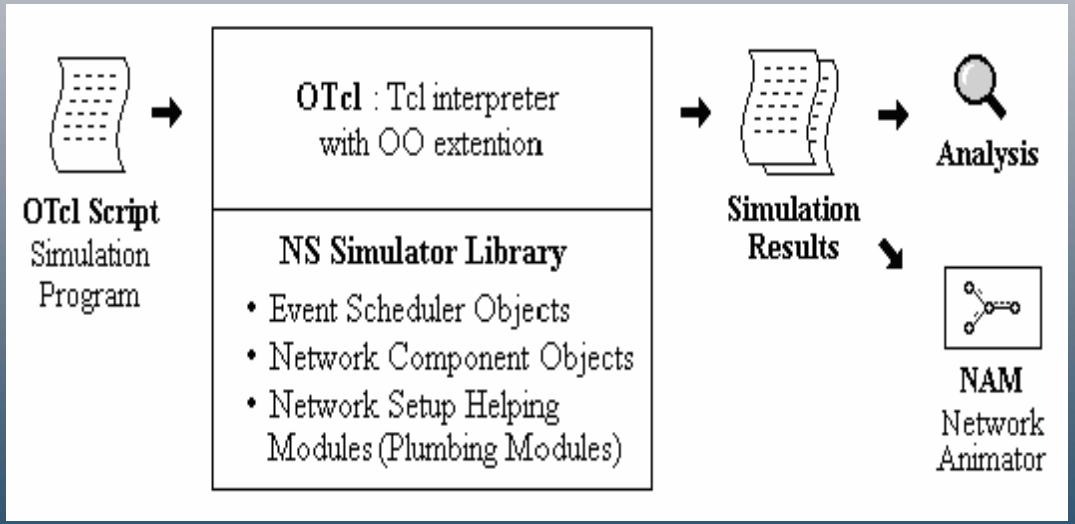
NS的脚本写法

网络论坛



NS的工作流程

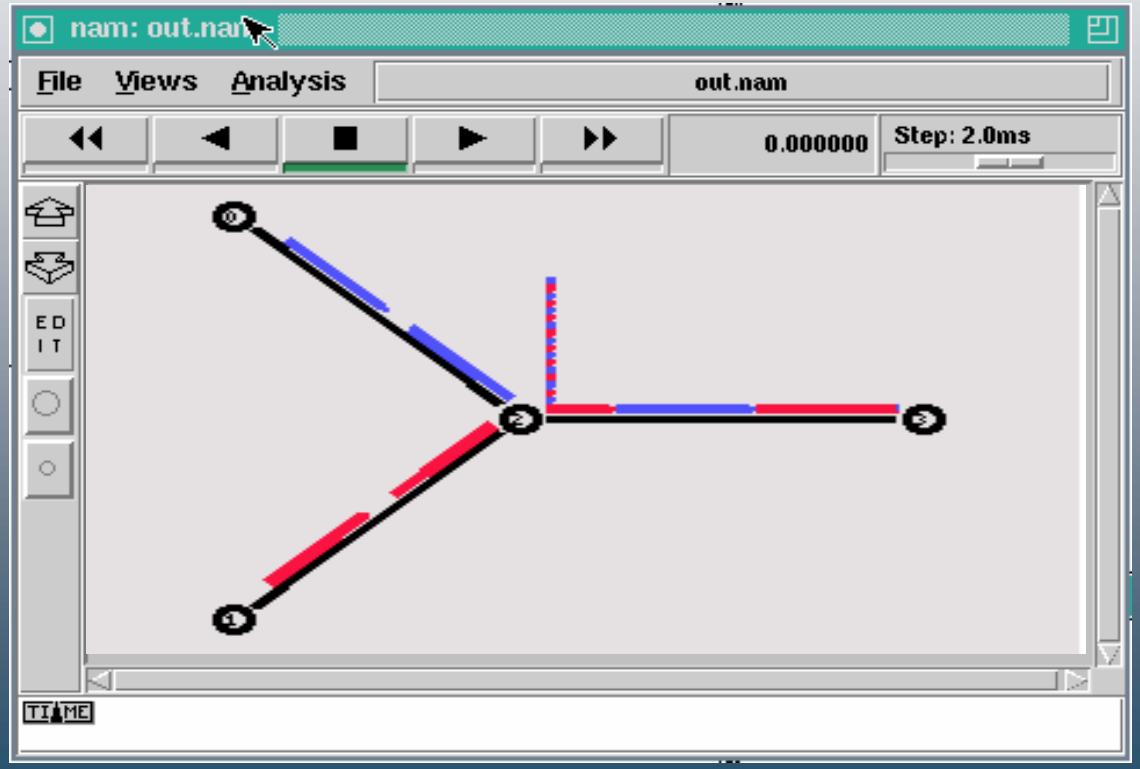
- NS代码使用OTCL语言编写，通过OTCL语言解释器解释，使用NS仿真库（包含Event Scheduler，网络组件对象和模块）进行编译和仿真，输出仿真的结果。
- 根据仿真的结果记录，可进行进一步的相关内容的分析，生成网络拓扑图，或者得到数据的可视化的图表。



NS的工作界面

网络论坛

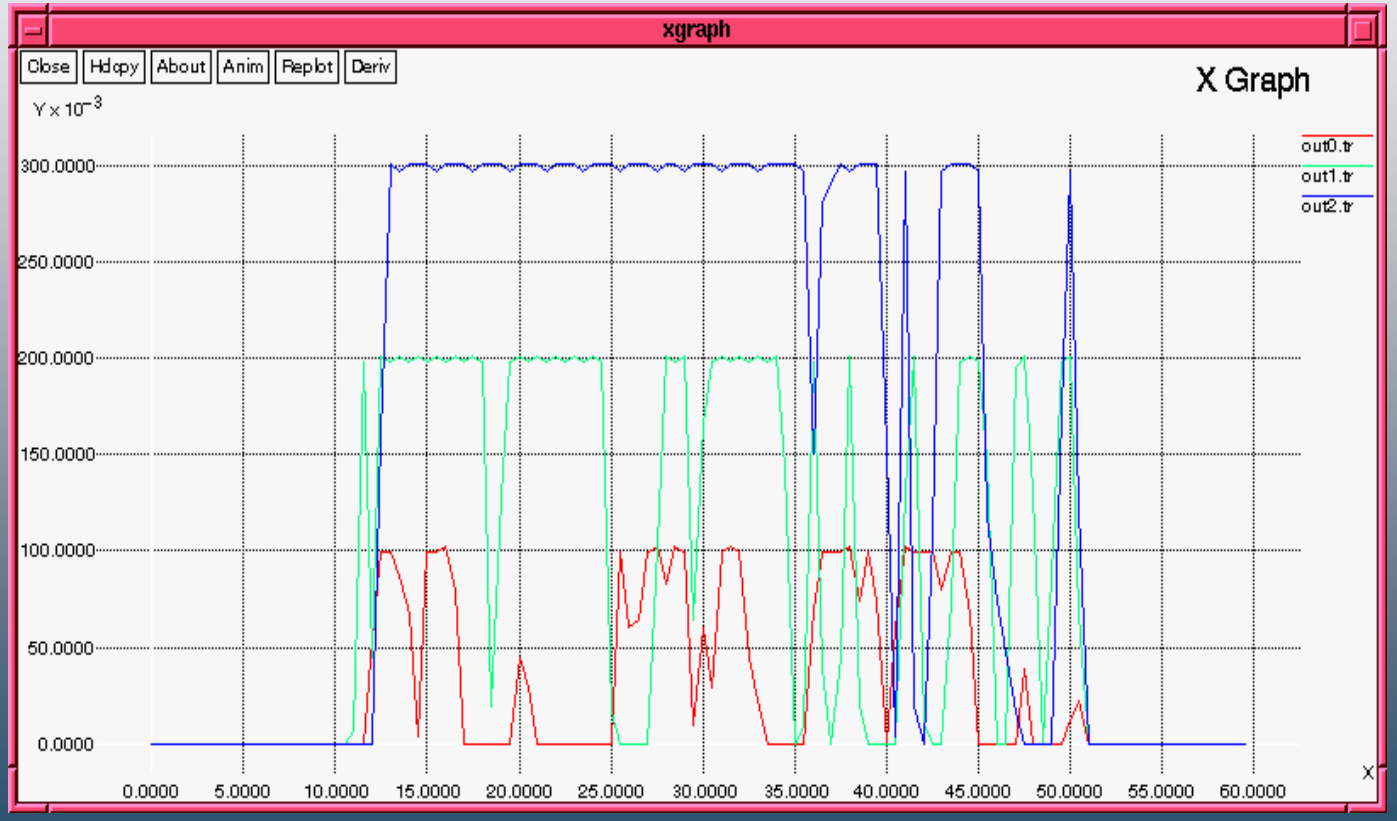
- 使用辅助的Network Animator (NAM) 工具，在NS中可以很清晰的显示网络的拓扑结构，这对整个网络的仿真的研究都有一定的帮助。(如下图)



NS的工作界面

网络论坛

- 使用**X Gragh**工具，可以将**NS**的仿真结果用图表形式生动形象的表示出来。



NS的应用范围

网络论坛

- NS设计的出发点是基于网络仿真，它集成了多种网络协议（如**TCP&UDP**），业务类型（如**FTP**，**Telnet**，**Web**，**CBR&VBR**等)，路由排队管理机制（如**Drop Tail**，**RED&CBQ**等），路由算法（如**Dijkstra**算法等）。
- 此外，**NS**还集成了组播业务和应用于局域网仿真有关的部分**MAC**层协议。其仿真主要针对路由层，传输层，数据链路层展开。
- 因此NS可以进行对固定，无线，卫星以及混和等多种网络的仿真。

NS的特点

网络论坛

- 较之众多的网络仿真软件，NS具有以下的突出特点：
  - 源代码公开；
  - 可扩展形强；
  - 速度和效率优势明显；
- 但同时也存在界面不够友好，可视化效果不甚理想，建模层次不够清晰的一些问题。