ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В КРИТИЧЕСКИХ ИНФРАСТРУКТУРАХ

УДК 004.75

РЕАЛИЗАЦИЯ ИМИТАЦИОННОЙ МОДЕЛИ ГИБРИДНОГО ПРОТОКОЛА МАРШРУТИЗАЦИИ БЕСПРОВОДНОЙ СЕТИ В NETWORK SIMULATOR (NS-2)

Молокович Игорь Аркадьевич
Публичное акционерное общество «Информационные телекоммуникационные технологии»
Кантемировская ул., 8, Санкт-Петербург, 197342, Россия
e-mail: igor-molokovich@yandex.ru

Аннотация. Рассматривается реализация имитационной модели предложенного гибридного протокола маршрутизации беспроводной сети в симуляторе Network Simulator (NS-2).

Ключевые слова: протокол маршрутизации; имитационная модель; network simulator (ns2).

IMPLEMENTATION OF THE HYBRID WIRELESS ROUTING PROTOCOL SIMULATION MODEL IN NETWORK SIMULATOR (NS-2)

Molokovich Igor
Public joint stock company «Information telecommunication technologies»
8 Kantemirovskay Av., St. Petersburg, 197342, Russia
e-mail: igor-molokovich@yandex.ru

Abstract. The implementation of the simulation model of the proposed hybrid wireless routing Protocol in the network Simulator (NS-2) is considered.

Keywords: routing Protocol; simulation model; network simulator (ns2).

Введение.

Рассмотрим беспроводную сеть, которая состоит из нескольких ad hoc компонентов (сетей доступа) и стационарных маршрутизаторов области ретрансляторов, которые работают в качестве ядра сети. Каждый ad hoc компонент рассматривается как отдельный регион. Маршрутизатор, подключенный к этому региону, несет ответственность за выдачу адресов узлам, маршруты к узлам других регионов и сетей и управление в этом регионе.

Предлагаемый гибридный протокол маршрутизации (HRP - Hybrid Routing Protocol) состоит из трех компонентов маршрутизации:

- протокола маршрутизации внутри сети доступа (Access Network Routing Protocol) – ANRP;
- протокола маршрутизации стационарных маршрутизаторов области ретрансляторов (Router Infrastructure Routing Protocol) – RIRP;
- протокола маршрутизации между сетями доступа (Access network Gateway Routing Protocol) – AGRP.

Маршрутизаторы инфраструктуры используют Router Infrastructure Routing Protocol. Поскольку маршрутизаторы инфраструктуры статические маршрутизаторы, RIRP относится к семейству беспроводных активных протоколов маршрутизации. RIRP работает на каждом статическом маршрутизаторе и обеспечивает маршруты в регионы, связанные с маршрутизаторами. Таблицы маршрутизации всегда актуальны, чтобы обеспечить немедленную маршрутизацию. Это позволяет существенно сократить задержки при определении маршрута.

В сети доступа маршруты поддерживаются с помощью протокола Access Network Routing Protocol – ANRP. ANRP – это реактивный протокол маршрутизации, который поддерживает расширенный поиск маршрутов и услуги технического обслуживания маршрута на основе локальных подключений к региону в режиме ad hoc.

Access network Gateway Routing Protocol используется, когда требуется определить маршрут между двумя ad hoc узлами или сетями доступа. Он получает информацию о маршрутах от протоколов RIRP и ANRP и создает полный маршрут от источника к получателю и предоставляет его в исходный узел. Когда узел требует маршрут, AGRP получает информацию от RIRP и ANRP обоих сетей доступа, строит маршрут и отправляет его ANRP и RIRP.

Имитационная модель протокола маршрутизации в рассмотренной сети строится с помощью Network Simulator (NS-2). Network Simulator (NS-2) - один из программных симуляторов моделирования процессов в телекоммуникационных сетях [1]. NS-2 позволяет описать топологию сети, конфигурацию источников и приёмников трафика, параметры соединений (полосу пропускания, задержку, вероятность потерь пакетов и т.д.) и множество других параметров моделируемой системы. Данные о динамике трафика, состоянии соединений и объектов сети, а также информация о работе протоколов фиксируются в генерируемом trace-файле.

NS-2 является объектно-ориентированным программным обеспечением. Его ядро реализовано на языке C++. В качестве интерпретатора используется язык скриптов (сценариев) OTcl (Object oriented Tool Command Language). NS-2 полностью поддерживает иерархию классов C++ и подобную иерархию классов интерпретатора OTcl. Обе иерархии обладают идентичной структурой, т.е. существует однозначное соответствие между классом одной иерархии и таким же классом другой. Объединение для совместного функционирования C++ и OTcl производится при помощи TclCl (Classes Tcl). В случае, если необходимо реализовать какую-либо специфическую функцию, не реализованную в NS-2 на уровне ядра, для этого используется код на C++.

Для моделирования используется программный симулятор NS-2 версии 2.34 на базе операционной системы Linux Xubuntu версии 18.04, установленной на виртуальной машине VMware Workstation 12 Player.

Абстракцией сетевого уровня в системе NS-2 является узел. Узел может принимать пакеты и классифицировать их по адресу и порту, являющимся полями IP-заголовка. Как было отмечено выше, узел является составным объектом. Простейшая структура узла изображена на рис. 1.

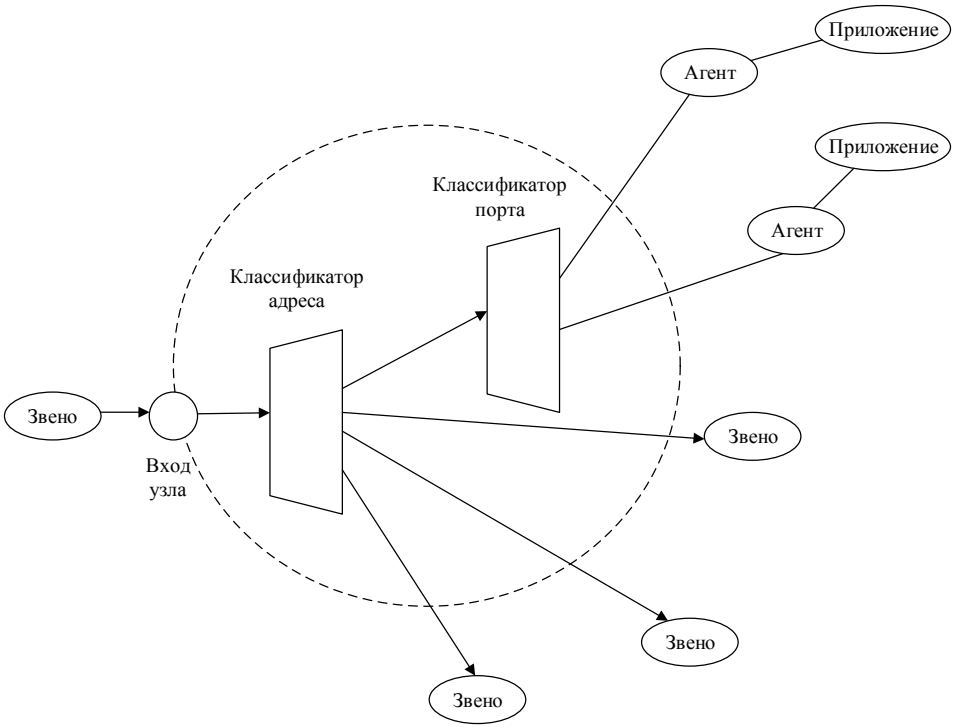

Рис. 1. Структура узла в NS-2

Прибывающие в узел пакеты принимаются входной точкой (Entry Point) узла и классифицируются классификатором адреса (Address Classifier), представляющим собой демультиплексор. Пакеты, адресованные другим узлам, передаются на одну из линий связи, подключенных к узлу, в соответствии с маршрутной таблицей, принадлежащей классификатору адреса. Пакеты же, адресованные данному узлу, передаются классификатору порта (Port Classifier), также являющемуся демультиплексором. В зависимости от номера порта, классификатор передает пакет одному из прикрепленных к узлу агентов (Agents), которые являются сетевыми объектами, не входящими в состав узла и отвечающими за реализацию протоколов более высокого уровня (например, транспортных протоколов UDP и различных реализаций TCP, протоколов маршрутизации и пр.).

Узел может иметь практически любое количество связей, а также портов с прикрепленными к ним агентами. Существуют также более сложные multicast-узлы, способные рассылать прибывающие пакеты нескольким адресатам.

Маршрутная таблица узла может быть, как статической, так и динамической. Во втором случае она формируется агентом маршрутизации, подключенным к узлу и получающим информацию о топологии сети путем обмена специальными пакетами с аналогичными агентами на других узлах (как это и делается при динамической маршрутизации в реальных сетях). Для создания TCP-соединения необходимо создать два агента: TCP-передатчик и TCP-приемник. TCP-приемник лишь принимает пакеты и посылает назад уведомления о получении (ACK). Поэтому такой агент в системе один: TCPSink. Передатчик же может вести себя различным образом в зависимости от порядка и моментов получения им уведомлений от адресата о получении отправленных им пакетов. Соответственно, в системе NS-2 имеется несколько вариантов TCP-передатчика, моделирующих различные реализации протокола TCP (TCP, TCP/Tahoe, TCP/Reno, TCP/Vegas), использующие разные комбинации алгоритмов TCP (Slow Start, Congestion Avoidance, Fast Recovery, Fast Retransmit). С протоколом UDP все обстоит гораздо проще, потому что уведомления о получении там не предусмотрены. Для передачи используется агент UDP, а для приема агент Null, просто принимающий и отбрасывающий все приходящие пакеты.

За прикладной уровень в NS-2 отвечают приложения (Applications). Приложение прикрепляется к агенту и служит для создания трафика. В NS-2 имеются приложения, моделирующие трафик, характерный для реальных протоколов прикладного уровня (FTP и Telnet), а также абстрактные генераторы трафика различного типа (например, CBR простейший генератор трафика с постоянным темпом выдачи пакетов). Приложения запускаются и останавливаются пользовательскими at-событиями.

Дуплексные (двусторонние) линии связи в NS-2 являются составными объектами, состоящими из двух противоположно направленных симплексных (односторонних) линий связи. Симплексная линия связи также является составным объектом. Она состоит из очереди, Null-агента, в который отправляются отброшенные из очереди пакеты, линии задержки и объекта, проверяющего и уменьшающего на единицу поле времени жизни (Time to Live, TTL) пакета. Кроме того, могут быть еще объекты-мониторы, следящие за состоянием очереди (постановка в очередь, извлечение, отброс пакетов).

Для реализации предлагаемого гибридного протокола маршрутизации в симуляторе NS-2 необходимо выполнить следующие действия [2].

В каталоге ns-2.34 создать каталог hrp с файлами:

hrp_pkt.h – определяет структуру пакета;

hrp_rtable.cc – содержит реализацию таблицы маршрутов;

hrp_rtable.h – объявлена таблица маршрутов;

Hrp.cc – содержит реализацию всех таймеров и агента маршрутизации;

hrp.h – определены все необходимые таймеры и агент маршрутизации, который выполняет функции протокола.

Пакет состоит из набора заголовков и необязательного поля данных. Формат заголовка пакета задается, когда создается объект *Simulator* (модель), где описан набор всех существующих (или используемых) заголовков: общий заголовок, который обычно используется всеми объектами, IP-заголовок, TCP-заголовок, RTP-заголовок (UDP в NS-2 использует RTP-заголовок) и трейс-заголовок. Также записаны сдвиги каждого заголовка в пакете. Это значит, что, неважно, используется или нет тот или иной заголовок, стек заголовков, создаваемый агентом, состоит из всех существующих заголовков, а сетевой объект может получить доступ к любому заголовку в стеке, используя соответствующую величину сдвига.

В качестве протокола маршрутизации внутри сети доступа (ANRP) в модели используется протокол AODV (Ad hoc On-Demand Distance Vector), относящийся к реактивным протоколам маршрутизации.

В качестве протокола маршрутизации стационарных маршрутизаторов области ретрансляторов (RIRP) в модели используется протокол DSDV (Destination Sequenced Distance Vector), который относится к проактивным протоколам маршрутизации.

Теперь необходимо внести изменения в некоторые файлы ns-2.34. В файле ns-2.34/common/packet.h, который определяет класс пакета, добавить строку:

// insert new packet types here

static const packet_t PT_HRP = 62;

В этом же файле находим функцию static packetClass classify(packet_t type) { и добавляем строку:

type == PT_HRP)

Далее находим функцию static void initName() и добавляем строку:

name_[PT_HRP] = «HRP»;

В файле ns-2.34/trace/cmu-trace.h добавить строку (trace определяет класс трассировки):

*void format_hrp(Packet *p, int offset);*

В файле ns-2.34/trace/cmu-trace.cc включить следующий файл заголовка в начало файла:

#include <hrp/hrp_pkt.h>

Далее в конце файла добавить код:

void

CMUTrace::format_hrp(Packet p, int offset){struct hdr_hrp_pkt* ph = HDR_HRP_PKT(p);*

if(pt_->tagged()) {sprintf(pt_->buffer() + offset, «-HRP:o %d -HRP:s %d -HRP:l %d», ph->pkt_src(), ph->pkt_seq_num(), ph->pkt_len());}else if (newtrace_) {sprintf(pt_->buffer() + offset, «-P HRP -Po %d -Ps %d -Pl %d «, ph->pkt_src(), ph->pkt_seq_num(), ph->pkt_len());}else {sprintf(pt_->buffer() + offset, «[HRP %d %d %d] «, ph->pkt_src(), ph->pkt_seq_num(), ph->pkt_len());}}

В этом же файле в функции *void CMUTrace::format(Packet* p, const char *why)* добавить строки:

case PT_HRP:

format_hrp(p, offset);

break;

В файле *ns-2.34/tcl/lib/ns-packet.tcl* после строки *foreach prot {* вставить:

HRP.

В конце файла *ns-2.34/tcl/lib/ns-default.tcl* добавить строку:

Agent/HRP set accessible_var_ true

В файле *ns-2.34/tcl/lib/ns-lib.tcl* в функции *Simulator instproc create-wireless-node args {* после *switch -exact $routingAgent_ {* вставить строки:

HRP {

set ragent [$self create-hrp-agent $node]

}

В конце этого же файла добавляются следующие строки:

Simulator instproc create-hrp-agent { node } {
 # Create Protoname Routing Agent
set ragent [new Agent/HRP [$node node-addr]]
$self at 0.0 «$ragent start»
$node set ragent_ $ragent
return $ragent
}

В файле *ns-2.34/queue/priqueue.cc* в функции *PriQueue::recv(Packet *p, Handler *h)* добавляем строку:

case PT_HRP:

В файле *Makefile.in* в разделе *INCLUDES* добавьте следующий каталог:

-I./hrp

В разделе OBJ_CC этого же файла добавить следующую строку в конце:

*hrp/Hrp.o hrp/hrp_rtable.o *

После внесения изменений в файлы ns-2.34 создается файл hrp.tcl, содержащий программу имитационного моделирования предлагаемого протокола гибридной маршрутизации в беспроводной сети. Следующим шагом является работа интерпретатора OTcl, использующего библиотеку NS-2, который обрабатывает написанную программу. Результатом его работы являются файлы трассировки, которые анализируются для получения результатов моделирования.

Заключение.

Таким образом, процесс добавления разработанного гибридного протокола маршрутизации в NS-2 включает создание в каталоге ns-2.34 собственного каталога с файлами, содержащими структуру пакета, определение и реализацию таблицы маршрутов, определение и реализацию всех таймеров и агента маршрутизации, который выполняет функции протокола; внесение необходимых изменений в файлы NS-2; написание программы имитационного моделирования, ее обработка и анализ полученных результатов.

СПИСОК ЛИТЕРАТУРЫ

1. Королькова А.В. Моделирование информационных процессов: учебное пособие / А.В. Королькова, Д.С. Кулябов. – М.: РУДН, 2014. – 191 с.
2. Francisco J. Ros, Pedro M. Ruiz. Implementing a New Manet Unicast Routing Protocol in NS2. *Dept. of Information and Communications Engineering University of Murcia.* December, 2004. - 35 с.

УДК 621.396.24

АНАЛИЗ ТЕХНОЛОГИЙ УПРАВЛЕНИЯ РЕЖИМАМИ УСТАНОВЛЕНИЯ СОЕДИНЕНИЙ В СЕТИ ДЕКАМЕТРОВОЙ РАДИОСВЯЗИ

Путилин Алексей Николаевич
Публичное акционерное общество «Информационные телекоммуникационные технологии»
Кантемировская ул., 8, Санкт-Петербург, 197342, Россия
e-mail: a.n.putilin@yandex.ru

Аннотация. Рассмотрено обобщенное представление совместного функционирования контуров управления автоматизированной радиосети (АРС) различных уровней: от управления сетью в целом до управления установлением канала связи между радиостанциями. Определена структура технологической и абонентской нагрузки в АРС. Предложенное описание позволяет найти характеристики распределения потоков технологической и абонентской нагрузки в АРС при использовании централизованного, распределённого и децентрализованного управления.

Ключевые слова: сеть декаметровой радиосвязи; управление установлением соединения; распределение нагрузки в сети.

ANALYSIS OF THE TECHNOLOGY CONTROL REGIMES ESTABLISH CONNECTIONS IN THE NETWORK DECAMETER RADIO

Putilin Alexey
Public joint stock company «Information telecommunication technologies»
8 Kantemirovskay Av., St. Petersburg, 197342, Russia
e-mail: a.n.putilin@yandex.ru

Abstract. The generalized representation of joint functioning of control circuits of the automated radio network (ARS) of various levels is considered: from control of a network as a whole to control of establishment of the communication channel between radio stations. The structure of the technological and user traffic in the ARS is determined. The proposed description makes it possible to find the characteristics of the flow distribution of technological and subscriber load in the ARS using centralized, distributed and decentralized control.

Keywords: decameter radio network; control of connection establishment; traffic distribution in the network.