Инж. П.В. Саранцев

ПРОГРАММЫ И ПРОТОКОЛЫ, ОСУЩЕСТВЛЯЮЩИЕ МЕЖСЕТЕВОЕ ВЗАИМОДЕЙСТВИЕ В СИСТЕМАХ УПРАВЛЕНИЯ И ЗАЩИТЫ АЭС

Требования по обеспечению внутреннего и внешнего взаимодействия комплекса электрооборудования в системах управления и защиты (КЭ СУЗ) диктуют необходимость разработки специализированных сетевых приложений, универсально работающих в сетевой среде комплекса и базирующихся на существующих протоколах передачи данных. Однако в связи с переходом на новую платформу LiNEM возникла необходимость замены механизма, программ и протоколов межсетевого взаимодействия. С этой целью была поставлена инновационная задача по созданию открытой ОС для КЭ СУЗ на базе семейства ОС Unix, отвечающей всем требованиям как к программным, так и к аппаратным сетевым элементам.

ОС Unix изначально была создана для работы с сетью в общем и с сетевыми протоколами в частности. Базовым протоколом для ОС является – IP, однако система также имеет возможность активации и других протоколов, самым известным из которых является IPX (этот протокол постепенно вытесняется, однако многие компании продолжают использовать его в своих сетях по причине простоты конфигурирования). Протокол IP в отличии от IPX имеет более развернутую архитектуру.

Основным протоколом, на котором базируется межсетевое взаимодействие в комплексе ПТК ИДС – является протокол IP. Он позволяет реализовать все стоящие перед системой задачи с требуемым уровнем надежности [1]. Устойчивость работы, надежность доставки потоков данных, возможность контроля CRC-сумм делает протокол IP совместно с системными и прикладными модулями ПО идеальным выбором для заданного класса систем. Надежность вышеуказанной связки подтверждается накопленным опытом разработки систем безопасности и мониторинга под ОС Unix [2]. Научная цель проведения этих работ – создание нового механизма межсетевого взаимодействия двух, четырех и более серверов, работающих как в режиме кластера, так и в режиме ускоренного синхронного обмена информацией диагностического характера.

В данном исследовании протокол IP представляет собой транспортную среду для передачи информации. Алгоритм же доставки в

рамках данного протокола предельно прост: низкие уровни работы открывают возможность потоковой передачи информации, контроль же за доставкой возлагается на более высокие уровни (TCP и UDP), при ошибке дейтограмма выбрасывается, а отправителю посылается соответствующее сообщение (ICMP) [3]. Новым подходом в разработке межсетевого взаимодействия можно также назвать создание подпрограммы оперативного контроля за профилем и качеством передачи потока информации, а также произведение записи этих параметров, получаемых из заголовков протокола ICMP (Internet Control Messaging Protocol) [4].

Поле "версия" в структуре заголовка пакетов характеризует версию IP-протокола (например 4 или 6). Формат пакета определяется создаваемой программой диагностики и может быть разным для разных значений поля “версия”. Значение поля “версия” во избежании непредсказуемых последствий должно контролироваться программой. HLEN - длина заголовка, измеряемая в 32-разрядных словах, обычно заголовок содержит 20 октетов (HLEN=5, без опций и заполнителя). Поле “полная длина” определяет полную длину IP-дейтограммы (до 65535 октетов), включая заголовок и данные. Однооктетное поле “тип сервиса” (TOS - type of service) характеризует то, как должна обрабатываться дейтограмма. Это поле делится на 6 субполей.

Субполе "приоритет" предоставляет возможность присвоить код приоритета каждой дейтограмме в виде: 0 – обычный уровень, 1 – приоритетный, 2 – медленный, 3 – срочный, 4 – экстренный, 5 – критический, 6 – межсетевое управление, 7 – сетевое управление.

Заметным нововведением, можно считать более плотную работу с полем "IP precedence". В ходе исследования было обнаружено, что ОС Unix в современном состоянии развития ядра способна корректно обрабатывать это поле срочности пакета. Получив эти данные, группе разработчиков были направлены соответствующие рекомендации по использованию этого поля в пакетах особой важности (для их ускоренной обработки как коммутатором, так и серверами следует устанавливать значение этого поля в значение "7").

Биты C, D, T и R характеризуют пожелание относительно способа доставки дейтограммы. Так D=1 требует минимальной задержки, T=1 – высокую пропускную способность, R=1 – высокую надежность, а C=1 – низкую стоимость. TOS играет важную роль в маршрутизации пакетов и многие маршрутизаторы учитывают эти запросы при выборе маршрута (протоколы OSPF и IGRP).

Одновременно во время исследования было выяснено, что только один бит из четырех в TOS может принимать значение 1. Значения по умолчанию равны нулю. Большинство из рекомендаций са-

моочевидны. Так при telnet наибольшую важность имеет время отклика, а для SNMP (управление сетью) - надежность. Исходя из отсутствия telnet-пакетов на реальном полигоне работы комплекса и присутствия большого количества пакетов snmp – необходимо отдавать все приоритетные очереди этим пакетам, направленным и исходящим с UDP порта 161.

Поля идентификатор, флаги (3 бита) и указатель фрагмента (fragment offset) управляют процессом фрагментации и последующей "сборки" дейтограммы. Идентификатор представляет собой уникальный код дейтограммы, позволяющий идентифицировать принадлежность фрагментов и исключить ошибки при "сборке" дейтограмм. Бит 0 поля флаги является резервным, бит 1 служит для управления фрагментацией пакетов (0 - фрагментация разрешена; 1 - запрещена), бит 2 определяет, является ли данный фрагмент последним (0 – последний фрагмент; 1 - следует ожидать продолжения). Поле “время жизни” (TTL - time to live) задает время жизни дейтограммы в секундах, т.е. предельно допустимое время пребывания дейтограммы в системе. При каждой обработке дейтограммы, например в маршрутизаторе, это время уменьшается в соответствии со временем пребывания в данном устройстве или согласно протоколу обработки. Если TTL=0, дейтограмма из системы удаляется. Во многих реализациях TTL измеряется в числе шагов, в этом случае каждый маршрутизатор выполняет операцию TTL=TTL-1. TTL помогает предотвратить зацикливание пакетов. Поле “протокол” аналогично полю тип в Ethernet-кадре и определяет структуру поля “данные”.

Поле “контрольная сумма” заголовка вычисляется с использованием операций сложения 16-разрядных слов заголовка по модулю 1. Сама контрольная сумма является дополнением по модулю 1 полученного результата сложения. Обратите внимание, здесь осуществляется контрольное суммирование заголовка, а не всей дейтограммы. Поле “опции” не обязательно присутствует в каждой дейтограмме. Размер поля опции зависит от того, какие опции применены. Если используется несколько опций, они записываются подряд без каких-либо разделителей. Каждая опция содержит один октет кода опции, за которым может следовать октет длины и серия октетов данных. Если место, занятое опциями, не кратно 4 октетам, используется заполнитель.

Поле “код” содержит номер опции. Поле “длина” определяет размер записи для опций, включая первые 3 октета. Указатель отмечает первую свободную позицию в списке IP-адресов (куда можно произвести запись очередного адреса). Интересную возможность представляет опция маршрут отправителя, которая открывает

возможность посылать дейтограммы по заданному отправителем маршруту. Это позволяет исследовать различные маршруты, в том числе те, которые недоступны через узловые маршрутизаторы. Существуют две формы такой маршрутизации: Свободная маршрутизация и Жесткая маршрутизация.

Жесткая маршрутизация означает, что адреса определяют точный маршрут дейтограммы. Проход от одного адреса к другому может включать только одну сеть. Свободная маршрутизация отличается от предшествующей возможностью прохода между двумя адресами списка более, чем через одну сеть. Поле “длина” задает размер списка адресов, а указатель отмечает адрес очередного маршрутизатора на пути дейтограммы.

IP-слой имеет маршрутные таблицы, которые просматриваются каждый раз, когда IP получает дейтограмму для отправки. Когда дейтограмма получается от сетевого интерфейса, IP первым делом проверяет, принадлежит ли IP-адрес места назначения к списку локальных адресов, или является широковещательным адресом. Если имеет место один из этих вариантов, дейтограмма передается программному модулю в соответствии с кодом в поле протокола. IP-процессор может быть сконфигурирован как маршрутизатор, в этом случае дейтограмма может быть переадресована в другой узел сети. Маршрутизация на IP-уровне носит пошаговый характер. IP не знает всего пути, он владеет лишь информацией – какому маршрутизатору послать дейтограмму с конкретным адресом места назначения.

Рассмотрим протоколы верхних уровней, использующие TCP в качестве транспортной среды более подробно.

Протокол пересылки файлов FTP

(File Transfer Protocol) реализует удаленный доступ к файлам на удаленных серверах. Для того, чтобы обеспечить надежную передачу, FTP использует в качестве транспорта протокол с установлением соединений - TCP. Кроме пересылки файлов протокол FTP предлагает и другие услуги. Так, пользователю предоставляется возможность интерактивной работы с удаленной машиной, например он может распечатать содержимое ее каталогов. Протокол FTP выполняет аутентификацию пользователей, а современные ее реализации даже обеспечивают режим безопасной аутентификации и шифрованного туннелирования информационного канала.

В стеке TCP/IP протокол FTP предлагает наиболее широкий набор услуг для работы с файлами, однако он является и самым сложным для программирования. Приложения, которым не требу-

ются все возможности FTP, могут использовать более экономичный протокол TFTP (Trivial File Transfer Protocol). Этот протокол реализует только передачу файлов, причем в качестве транспорта используется более простой, чем TCP, протокол без установления соединения - UDP.

Приведем несколько новых команд, которые были исследованы во время тестирования работы программы зеркалирования на стенде НПП ВНИИЭМ:

Выход в shell - интерпретатор на локальной системе.

dir [удаленная директория] [локальный файл]

ls [удаленная директория] [локальный файл]

Выводит список файлов в директории либо нестандартный вывод, либо, если указано имя локального файла, в этот файл.

get [удаленный файл] [локальный файл]

Вызывает передачу копии удаленного файла на ваш компьютер. В случае, если имя локального файла не было задано, то оно совпадает с именем удаленного файла.

mget [удаленные файлы]

Для получения нескольких файлов

hash

Служит переключателем для индикации каждого полученного блока данных в 1024 байта, повышает наглядность процедуры.

cd [удаленная директория]

Сменить директорию. Существуют также 'cdup' или 'cd' для возврата на один или выше

lcd

Меняет рабочую директорию на локальной машине (без аргумента - переход в домашнюю директорию пользователя)

bin (или binary)

Переключает в режим передачи двоичных файлов

ascii

Переключает в режим передачи текстовых файлов (обычно по умолчанию)

prompt

Переключает интерактивную подсказку. Часто при использовании команды 'mget' желательно предварительно набрать 'prompt', чтобы не давать многократные подтверждения

pwd

Выводит имя удаленной рабочей директории

mkdir [имя директории]

Создает директорию на удаленной машине

open хост [порт]

Устанавливает соединение с заданным FTP сервером

put [локальный файл] [удаленный файл]

Пересылает файл на удаленную систему. Если имя удаленного файла не указано, то оно совпадает с именем на локальной системе

quit

Синоним для 'bye'

recv [удаленный файл] [локальный файл]

Синоним для команды 'get'

reget [удаленный файл] [локальный файл]

"Дополучение" удаленного файла в том случае, когда часть его уже есть на локальной машине. Команда особенно полезна для получения больших файлов при возможных резервах соединения

delete [удаленный_файл]

Стирает удаленный файл

close

Обрывает FTP сеанс с удаленным сервером и возвращает к командному интерпретатору

bye

Оканчивает работу с FTP сервером и приводит к выходу и из интерпретатора.

Протокол TELNET

Протокол обеспечивает передачу потока байтов между процессами, а также между процессом и терминалом. Наиболее часто этот протокол используется для эмуляции терминала удаленного компьютера. При использовании сервиса telnet пользователь фактически управляет удаленным компьютером так же, как и локальный пользователь, поэтому такой вид доступа требует хорошей защиты. Поэтому серверы telnet всегда используют как минимум аутентификацию по паролю, а иногда и более мощные средства защиты, например систему Cerberos. Во время исследования пакета и протокола, реализующего данный сервис, было выявлено большое количество существенных недостатков, исправление которых было технически невозможно, по причине исторически созданной архитектуры самого протокола. Новым направлением работы в связи с этим стало предложение по отказу от использования протокола telnet, и в случае необходимости переход на более безопасный протокол ssh. Рассмотрение и изучение этого протокола будет проведено позже. Протокол, программная и серверная часть будет установлена на разработанном в НПП ВНИИЭМ дистрибутиве LiNEM.

Протокол SNMP

Simple Network Management Protocol используется для органи-

зации сетевого управления. Изначально протокол SNMP был разработан для удаленного контроля и управления маршрутизаторами Internet, которые традиционно часто называют также шлюзами. С ростом популярности протокол SNMP стали применять и для управления любым коммуникационным оборудованием - концентраторами, мостами, сетевыми адаптерами и т.д. Проблема управления в протоколе SNMP разделяется на две задачи.

Первая задача по изучению протокола была связана с передачей диагностической информации. Протоколы передачи управляющей информации определяют процедуру взаимодействия SNMP-агента, работающего в управляемом оборудовании, и SNMP-монитора, работающего на компьютере администратора, который часто называют также консолью управления. Протоколы передачи определяют форматы сообщений, которыми обмениваются агенты и монитор.

Вторая задача была связана с контролируемыми переменными, характеризующими состояние управляемого устройства. Стандарты регламентируют, какие данные должны сохраняться и накапливаться в устройствах, имена этих данных и синтаксис этих имен. В стандарте SNMP определена спецификация информационной базы данных управления сетью. Эта спецификация, известная как база данных MIB (Management Information Base) [5], определяет те элементы данных, которые управляемое устройство должно сохранять, и допустимые операции над ними. Для осуществления диагностики было предложено использовать собственный диагностический интерфейс со встроенной возможностью как визуализации состояния устройств и зондов, так и всей сети в целом.

В ходе исследования, однако, было обнаружено, что протокол SNMP не может использовать IPX транспорт [6].

Протокол NFS

Сетевая файловая система NFS (Network File System) впервые была разработана компанией Sun Microsystems Inc. NFS использует транспортные услуги UDP и позволяет монтировать в единое целое файловые системы нескольких машин с ОС Unix. Бездисковые рабочие станции получают доступ к дискам файл-сервера как к своим локальным дискам.

NFS значительно увеличивает нагрузку на сеть. Если пропускная способность сети позволяет NFS нормально работать, то пользователи получают большие преимущества. Поскольку сервер и клиент NFS реализуются в ядре ОС, все обычные несетевые программы получают возможность работать с удаленными файлами,

расположенными на подмонтированных NFS-дисках, точно также как с локальными файлами.

Система X-Window

Система использует протокол X-Windows, который работает на базе TCP, для многооконного отображения графики и текста на растровых дисплеях рабочих станций. По первым результатам тестирования система зарекомендовала себя как очень стабильная и надежная для работы как администратора системы, так и персонала на объекте, использующего программы сетевого мониторинга. В рамках создания дистрибутива LiNEM было решено создавать единую пользовательскую среду (ЕПС) на базе этого протокола. Как уже подчеркивалось выше, новым подходом в реализации этих функций стала именно ЕПС, разрабатывающаяся на базе X-Window.

Протокол IPX

Протокол IPX предназначен для передачи дейтограмм в системах, неориентированных на соединение, он обеспечивает связь между NetWare серверами и конечными станциями (так же может обеспечиваться межсетевое взаимодействие между ОС Unix и Novell Netware). Максимальный размер IPX-дейтограммы составляет 576 байт, из них 30 байт занимает заголовок. Предполагается, что сеть, через которую транспортируются эти дейтограммы, способна пересылать пакеты соответствующей длины. IPX-пакеты могут рассылаться широковещательно, для этого поле “тип” должно принять значение 0x14, адрес сети назначения должен соответствовать локальной сети, адрес узла назначения при этом принимает значение 0xFFFFFF.

IPX-пакеты, передаваемые по сети Ethernet, могут иметь несколько разных форматов. Старейший из них носит в Novell название “802.3” [7] и используется по умолчанию в версиях до 3.11. В последующих версиях форматом по умолчанию является “802.2”. Применим также и формат, называемый Ethernet II, который наиболее близок идеологии TCP/IP. Сеть в Netware - это логический канал, который используется совместно рядом узлов так, что они могут взаимодействовать друг с другом непосредственно. Так процессы, реализуемые на одном сервере, считаются подключенными к внутренней IPX-сети. Все пользователи сети типа Ethernet II образуют логическую сеть IPX. Все пользователи одной сети типа 802.3 рассматриваются как узлы различных сетей IPX.

Серверы Netware можно сконфигурировать так, чтобы они вос-

принимали пакеты разных типов, и поэтому могли иметь непосредственные связи с разными сетями. IPX-сервер может выполнять и функции маршрутизатора. За заголовком следуют данные, их объем определяется кодом поля “Длина пакета” (минус 30) и лежит в диапазоне от 0 до 546 байт.

Поле “Контрольная сумма” (2 байта) устанавливается ipx-драйвером равным 0xffff, это означает, что контрольного суммирования не производилось. Приложениям разрешено использовать поле “контрольная сумма” при работе с кадрами Ethernet II, ieee 802.2 и Ethernet SNAP и запрещено для работы с кадрами ieee 802.3. Контрольная сумма служит лишь для контроля правильности IPX-заголовка и не имеет никакого отношения к полю данных IPX-дейтограммы. Для того чтобы работать с контрольными суммами на NetWare-сервере, следует выдать команду set enable IPX checksum=n, где n указывает на то, что контрольная сумма использована.

Поле “Длина пакета” (2 байта) содержит число байт в пакете, включая заголовок, и может лежать в пределах от 30 (только заголовок) до 576. В действительности максимальная длина IPX-пакета равна 1518 байт, но при прохождении пакетов через маршрутизаторы, когда не используется протокол LIP(large internet packet, протокол межсетевой пересылки больших пакетов) максимальная длина может быть равной лишь 576 байт (что и принято по умолчанию). Следует также иметь в виду, что согласно регламентациям Novell длина пакета может принимать только четные значения. Поле “управление пересылкой” (1 байт) устанавливается IPX-драйвером равным нулю перед посылкой пакета. Каждый маршрутизатор увеличивает значение этого поля на 1. Если пакет прошел через 15 маршрутизаторов, очередной - удалит пакет из сети (в некотором смысле это аналог поля время жизни - TTL в протоколах TCP/IP). Поле “управление пересылкой” можно использовать для оптимизации маршрутов в локальной сети. Если станция общается только с серверами соседней субсети, ее следует переключить туда и снизить тем самым нагрузку маршрутизатора. Контроль за содержимым этого поля выполняется протоколом IPX. Поле “тип пакета” (1 байт) устанавливается прикладной программой. При использовании протокола ipx это поле должно содержать нуль, в случае использования протокола SPX - 5, а для протокола NCP(Netware core protocol) -17. Поля адрес узла назначения и адрес узла отправителя содержат 12-байтовые структуры ipxaddr_1. Эта структура включает в себя 4 байта адреса сети (присваивается администратором сети при установке сети Novell), 6 байт адреса узла (физический адрес, задается изготовителем сетевого интерфейса) и 2 байта дескрипто-

ра соединителя (socket, необходим для адресации программы, принимающей пакеты, заполняется приложением). Пакеты, адресованные серверу в NetWare 3.x или 4.x содержат в поле “адрес узла” получателя код 0x00 00 00 00 00 01 (аналогичный код будет записан в поле “адрес отправителя”, если им является сервер). Адрес же узла получателя на уровне Ethernet или Token Ring будет равен физическому сетевому адресу интерфейса или локального маршрутизатора, если сервер размещен в другой субсети. Соединители (socket) служат для управления обработкой пакетов. Широковещательный пакет будет получен ЭВМ, если она имеет открытый соединитель для процесса, которому он адресован. По этой причине должны приниматься специальные меры, чтобы предотвратить возможность посылки двумя программами пакетов различного типа на один и тот же соединитель. Ряд номеров соединителей зарезервировано IPX-протоколом для определенных целей:

Дескрипторы соединителей для рабочих станций задаются динамически и их коды лежат в диапазоне 0x4000 - 0x8000. В отличии от протокола IP, IPX не имеет фиксированных адресов для сетей или интерфейсов, которые следует конфигурировать. Вместо этого рабочие станции получают свои сетевые номера от маршрутизатора, к которому они подсоединены, и используют Ethernet-адрес в качестве номера узла.

Приложение должно устанавливать поля тип пакета и адрес узла назначения, а IPX-драйвер заполняет остальные поля.

Коды IPX-пакетов: 0 - обычный IPX-пакет, 1 - пакет с маршрутной информацией (RIP - routing information protocol), 2 - отклик, 3 - ошибка, 4 - информационный пакетный обмен (pep - packet exchange protocol), 5 - последовательный пакетный обмен (SPX - sequence packet exchange), 17 - протоколы ядра NetWare (NCP), 20 - именной пакет netbios (широковещательный).

Маршрутная информация передается между серверами и маршрутизаторами. Динамический маршрутный протокол RIP (routing information protocol, базируется на стандарте Xerox IP; см. также RFC-1058) [8-12] обеспечивает конечные станции информацией, которая необходима для динамического управления оптимизацией маршрутов. Рассылка маршрутной информации производится с помощью широковещательных пакетов. Сети Novell являются источником значительных потоков широковещательных пакетов. Аналогичным образом объекты сети оповещаются о других изменениях в сетевой среде, например рассылка информации о доступных услугах (SAP - service advertisement protocol). Протокол SAP позволяет узлам, которые предлагают определенные услуги (например файл-серверы или принт-серверы), сообщать о своих адре-

сах и видах доступных услуг. Администратор может регулировать поток таких пакетов, задавая постоянные времени для таймеров обновления информации.

Маршрутизация пакетов в сети достаточно проста. Каждому сетевому сегменту маршрутизатор присваивает номер в пределах от 1 до fffffffe. Каждой группе устройств присваивается “сетевой номер”, который представляет эту группу во всех маршрутизаторах сети. Пакеты, посылаемые от одного члена группы другому, посылаются непосредственно. Пакеты от одного члена группы к объекту из другой группы будут пересланы через маршрутизаторы. Для выбора маршрута в пределах локальной сети используется маршрутный протокол RIP.

ЛИТЕРАТУРА

1. Программное обеспечение компьютеров в системах безопасности атомных электростанций // Международный стандарт МЭК 60880, 1986.

2. Ядерное приборостроение. Обзор применения стандарта МЭК 60880 // Технический отчет МЭК 61940.

3. RFC№ 844. Who talks ICMP, too? - Survey of 18 February 1983. R. Clements. Feb-18-1983.

4. RFC№ 1256. ICMP Router Discovery Messages. S. Deering, Ed.. Sep-01-1991.

5. RFC № 1089. SNMP over Ethernet. M.L. Schoffstall, C. Davin, M. Fedor, J.D.

6. RFC № 1098. Simple Network Management Protocol (SNMP). J.D. Case, M. Fedor.

7. M.L. Schoffstall, C. Davin. April 1989.

8. Internet Engineering Task Force. Request for Commands #1234 http://www.ietf.org/rfc/rfc1234.txt?number=1234.

9. Internet Engineering Task Force. Request for Commands #1420 http://www.ietf.org/rfc/rfc1234.txt?number=1420.

10. Internet Engineering Task Force. Request for Commands #1553 http://www.ietf.org/rfc/rfc1234.txt?number=1553.

11. Internet Engineering Task Force. Request for Commands #1634 http://www.ietf.org/rfc/rfc1234.txt?number=1634.

12. Internet Engineering Task Force. Request for Commands #11792 http://www.ietf.org/rfc/rfc1234.txt?number=1792.