Визуализация данных в распределенных системах сбора

В.Н. Черненков

Special Astrophysical Observatory of the Russian AS, Nizhnij Arkhyz 357147, Russia

Аннотация. Изложен способ передачи графической информации на основе UDP, предназначенный для управления системами сбора данных по сетям Internet. Описаны алгоритмы работы графических программ сервера и клиента, позволяющие работать в режиме мультидоступа. Представлена реализация указанного способа на РАТАН-600.

1 Введение

Наиболее гибкими стандартизованными средствами визуализации графических данных является оконный интерфейс X11. В распределенных системах сбора данных, базирующихся на технологии интернет, X11 позволяет проводить удаленный вывод и отображение этих данных в реальном времени. При удаленной работе возникает необходимость визуального контроля одновременно с нескольких рабочих мест. Однако, как известно, протокол передачи данных в стандарте X11 требует значительных вычислительных ресурсов и высокопроизводительных сетевых средств (более 100 кбайт в секунду). Он базируется на сравнительно интенсивном универсальном протоколе обмена данными, транспортный уровень которого TCP [1] ориентирован на жесткое соединение между процессами, порождающим графические данные, с одной стороны, и выводящим эти данные на экран графической станции, с другой. При увеличении количества пользователей графических данных необходимые ресурсные затраты растут пропорционально числу пользователей, что может оказаться серьезным препятствием для реализации системы реального времени с множественным доступом. Ведущие зарубежные фирмы – производители программного обеспечения для графических рабочих станций, одна из которых SUN Soft, в настоящее время активно разрабатывают более экономичный протокол передачи графических данных. В частности, разработки на базе гипертекстовых объектно-ориентированных языковых средств типа Java позволят решить проблему передачи динамических изображений по “медленным” сетям передачи данных. Однако внедрение их требует больших аппаратных затрат и до введения стандартов еще далеко.

В данной работе предложен альтернативный способ передачи графических данных для систем сбора, позволяющий сократить необходимые ресурсные затраты для мультидоступа. Способ дает возможность использовать как графические средства X11, так и обходиться без них при сохранении базовой идеологии ”клиент – сервер”.

[1] Transport Control Protocol

2 UDP как средство мультидоступа к данным

В современных операционных системах и сетевых средствах, базирующихся на сокетном механизме сетевых приложений [2] (BSD UNIX, SUN-OS, Linux, MS-Windows/WinSock и др.) передача данных между сервером и клиентами осуществляется через так называемые "гнезда" (sockets) – сокеты, открываемые с двух сторон с получением дескрипторов анологично файловым (Stevens, 1990). Здесь под сервером понимается программа, с одной стороны, принимающая поток данных из драйвера аппаратуры системы сбора (например КАМАК) и, с другой стороны, выдающего его в сеть через сокет по запросам от клиентов. Клиент, в свою очередь, – это программа, которая выдает в свой сокет запросы на получение данных из сети, получает их оттуда и выводит на внешнее графическое устройство. Графическим устройством могут быть как средства SVGA-графики (через функции SVGA-библиотеки), так и окружение X11.

2.1 Широковещательная передача

В отличие от TCP, протокол UDP [3] не ориентирован на соединение и, следовательно, к одному процессу – источнику данных, то есть к сокету, открытому этим процессом, могут подсоединиться сразу несколько потребителей этих данных. Обычный способ параллельного соединения осуществляется посредством выделения для сокета фиксированного номера порта и формирования группового (broadcast) адреса, охватывающего адресное пространство всех потенциальных потребителей, – клиентов. Очевидное преимущество – всего один сокет и всего один поток графических данных независимо от числа клиентов. Однако такой способ может быть использован только внутри одного сегмента локальной сети, поскольку широковещательная адресация, реализуемая на аппаратном уровне в широкополосных сетях (ARCnet, Ethernet и т.п.), обычно не транслируется в другие сети, даже если они того же типа.

Для передачи графических данных в удаленные пункты лучше использовать конкретную адресацию, иначе мы были бы вынуждены передавать данные для всей сети Internet.

2.2 Адресная передача точка – точка

При организации адресной передачи по UDP обмен строится по принципу запрос – ответ. Программа клиент посылает порцию данных – дейтаграмму запроса серверу, а тот высылает ответную дейтаграмму по адресу запрашивающего. Адрес клиента сервер берет из своего сокета. До следующего запроса сервер про этого клиента забывает.

Безусловный плюс по сравнению с TCP – не надо следить за соединением и для следующего клиента сокет всегда свободен. Если по каким-либо причинам связь рвется, а затем восстанавливается, то никаких заторов не возникает.

Недостаток – поток прямо пропорционален количеству клиентов. К тому же, из-за необходимости двухстороннего обмена данными поток еще и удваивается. При появлении запроса от нового клиента программа-сервер обычно создает свою копию с вновь открываемым сокетом для обслуживания этого клиента, что ведет к росту затрат на необходимую память.

Учитывая это и тот недостаток, что UDP не содержит встроенных средств контроля гарантированной доставки дейтаграмм, такая передача данных используется только в специальных случаях, когда перечисленные недостатки скомпенсированы или не имеют принципиального значения.

---

[2] В операционных системах с потоковой (streams) реализацией сетевых API, например, UNIX System V, работа с сокетами как правило полностью эмулируется при использовании соответствующих системных библиотек, и отличия в написании программ несущественны.

[3] User Datagram Protocol

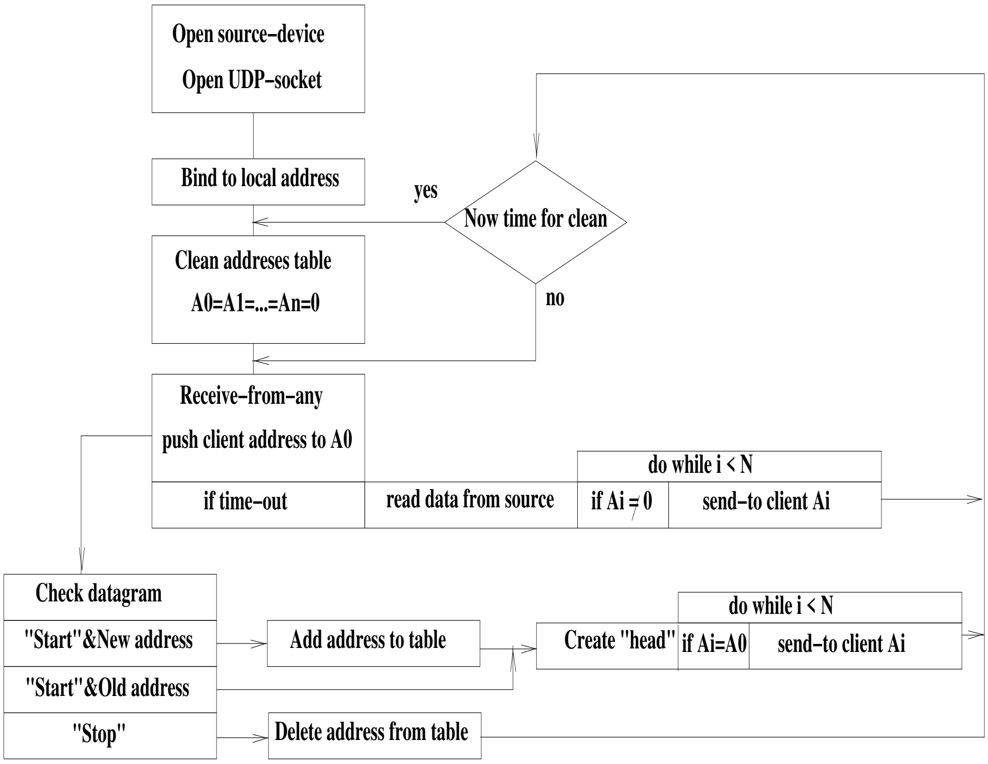

Рис. 1: Алгоритм работы сервера

2.3 Адресная передача точка – группа

Предлагается построить процедуру передачи, используя следующие свойства данных:

1. Графические данные в основном однотипны, то есть набор дейтаграмм ограничен фиксированным числом заранее известных структур.

2. Структуры графических данных привязаны к средствам их отображения с ограниченными полями, поэтому циклы вывода данных в реальном времени имеют периодический характер, связанный с заполнением экрана.

3. Количество потребителей графических данных ограничено сравнительно небольшим числом рабочих станций, с которых производится управление наблюдательным процессом.

4. Цена потерь графических данных в системе сбора невысока, поскольку средства визуального контроля вторичны по отношению к записи исходных данных на носители.

Предлагается следующий алгоритм взаимодействия между множеством клиентов и сервером. Этот алгоритм позволяет минимизировать поток дейтаграмм в сети и количество используемых сетевых ресурсов.

Основная идея алгоритма в том, что сервер открывает всего один UDP-сокет, но резервирует область для массива адресов фиксированного[4] множества клиентов. Номер порта также один и известен клиентам вместе с адресом сервера. Клиенты при старте также открывают у себя по одному сокету.

[4] Фиксация размера массива не принципиальна, и может быть заменена динамическим выделением памяти по каждому запросу от нового клиента

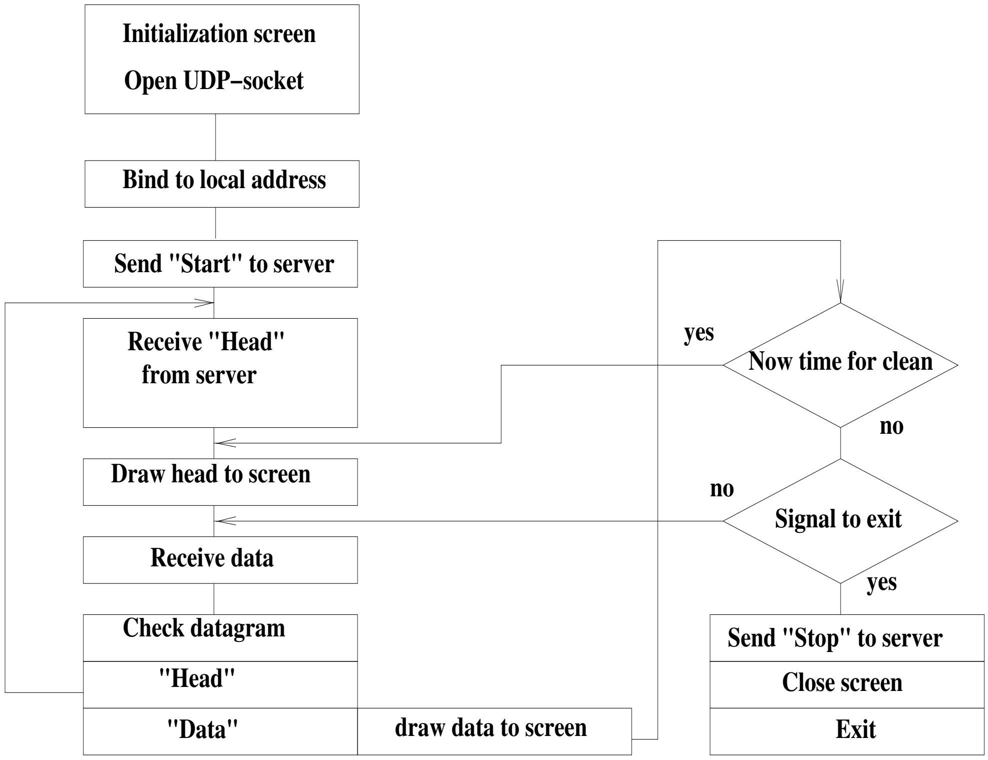

Рис. 2: Алгоритм работы клиента

Рассмотрим работу сервера. Сервер после старта программы создает собственную копию в виде “потомка”, который отсоединяется от управляющего терминала и становится самостоятельным “демоном”, а “родитель” свою работу завершает. После этого происходит открытие драйвера аппаратуры системы регистрации для приема “шапочки” параметров и собственно потока данных для визуализации. “Шапочка” запоминается в виде структуры данных для передачи в виде дейтаграммы, назовем ее № 1. Затем открывается сокет для приема запросов и выдачи данных. После открытия сокета программа попадает в бесконечный цикл ожидания запросов из сети и чтения данных из драйвера.

Исходно массив адресов клиентов очищен. При получении дейтаграммы она анализируется на содержание, и если соответствует стартовой, то из сокета извлекается адрес первого клиента и заносится в первую свободную ячейку адресного массива. Затем этому клиенту посылается дейтаграмма “шапочки” и начинается передача графических данных – дейтаграмм № 2. Данные передаются по мере их поступления от аппаратуры сбора, при этом производятся также попытки чтения дейтаграмм запросов от других клиентов.

Поступление дейтаграммы “Start” от клиента приводит к добавлению соответствующего адреса в массив адресов, передачи ”шапочки” и включению этого клиента в цикл передачи дейтаграмм № 2. Получение же сервером дейтаграммы “Stop” вызывает процедуру обнуления соответствующего адреса клиента в массиве адресов и прекращению передачи данных для него. Поэтому дополнительной обязанностью клиентов является периодическое подтверждение собственной “жизнеспособности” посылкой стартовых дейтаграмм. Эту процедуру логично связать с заполнением экрана клиента и его очисткой.

Поскольку клиенты могут “забывать” посылать завершающие дейтаграммы, сервер принудительно через некоторое время очищает адресный массив. Алгоритм работы сервера изображен на рис.1.

Алгоритм работы клиента изображен на рис.2. Работа клиента начинается с процедуры инициализации графического экрана, при этом средствами X11 могут создаваться специальные окна, раскрашенные в соответствии с палитрой и ресурсами рабочей станции. Затем программа открывает дейтаграммный сокет и посылает запрос на прием "шапочки" для оформления графических данных. Целесообразно организовать цикл таких запросов с выдержкой времени. (Обычно достаточно десятка секунд). После получения дейтаграммы № 1 вызывается процедура графического оформления данных в виде рамочек, атрибутов, шкал, сетки времени и т.п. Затем программа попадает в цикл ожидания дейтаграмм № 2 и разрисовки отсчетов по полученным данным в реальном времени. По мере заполнения экрана, а также при выбранном тайм-ауте[5] на прием данных, программа посылает стартовый запрос на прием дейтаграмм № 1 и перерисовывает оформление экрана. Такие циклы повторяются до тех пор, пока работу программы не прервет событие (или сигнал) на ее окончание. Перед выходом из программы и закрытием сокета производится посылка дайтаграммы "Stop" серверу.

3 Визуализация данных системы сбора первого облучателя

Практическая реализация изложеннного алгоритма взаимодействия графического сервера и клиентов была реализована в комплексе сбора данных облучателя № 1 РАТАН-600 – "Continuous"[6]. Программа сервера графических данных vsd, позволяет одновременно обслуживать до восьми независимых рабочих станций[7]. С помощью программ клиентов visn и xvisn для вывода графики на SVGA-экран и X-терминал соответственно, можно контролировать наблюдательный процесс по нескольким каналам (на экран можно параллельно выводить до восьми графиков реального времени). За основу программ клиентов были взяты локальные программы vis и xvis, ранее считывающие данные из промежуточного файла. Частота съема графических отсчетов составляет 10 Гц, а вывод данных на экран – раз в секунду. Визуальный контроль наблюдений можно проводить практически на любой рабочей станции в САО или РАТАН-600 независимо от удаления. Так, на рис.3 изображен экран компьютера, расположенного на Нижней Научной площадке САО (удаление от РАТАН-600 – 25 км) и связанного с облучателем № 1 через сети ARCnet, Ethernet и выделенный телефонный канал.

Благодарности. Автор благодарен Шергину В.С. за ценные советы по проблемам сетевого взаимодействия программ.

Список литературы

Comer D.E.: 1991, Interworking with TCP/IP, vol. 1: Principles, Protocols, and Architecture, 2nd ed., Prentice-Hall, 550.

Stevens W.R.: 1990, UNIX Network Programming, Prentice-Hall, 783

RFC 768 - User Datagram Protocol

RFC 793 - Transmission Control Protocol

RFC 896 - Congestion Control in IP/TCP Internetworks

RFC 1013 - X Window System Protocol, version 11: Alpha update April 1987

Верходанов О.В., Черненков В.Н., Шергин В.С.: 1995, Идеология гибкой системы обработки астрономических данных. II. Визуализация. Препр. Спец. астрофиз. обсерв., **106**, 58-65

Доценко А.В., Исаков А.Б., Рябов А.Ю.: 1994, под ред. Рябова А.Ю., UNIX X WINDOW MOTIF, Основы программирования ч.1 - М., АО "Аналитик", 252

[5] Система контроля по тайм-ауту полезна, поскольку как правило невозможно гарантировать непрерывную работу сервера и сети.

[6] Модернизированный вариант многопользовательского территориально распределенного комплекса – МИТРАКРАТ (Черненков, Цыбулев, 1995).

[7] Объем массива адресов, ограничивающий число одновременно обслуживаемых клиентов, задается при компиляции программы и может быть увеличен до необходимого значения.

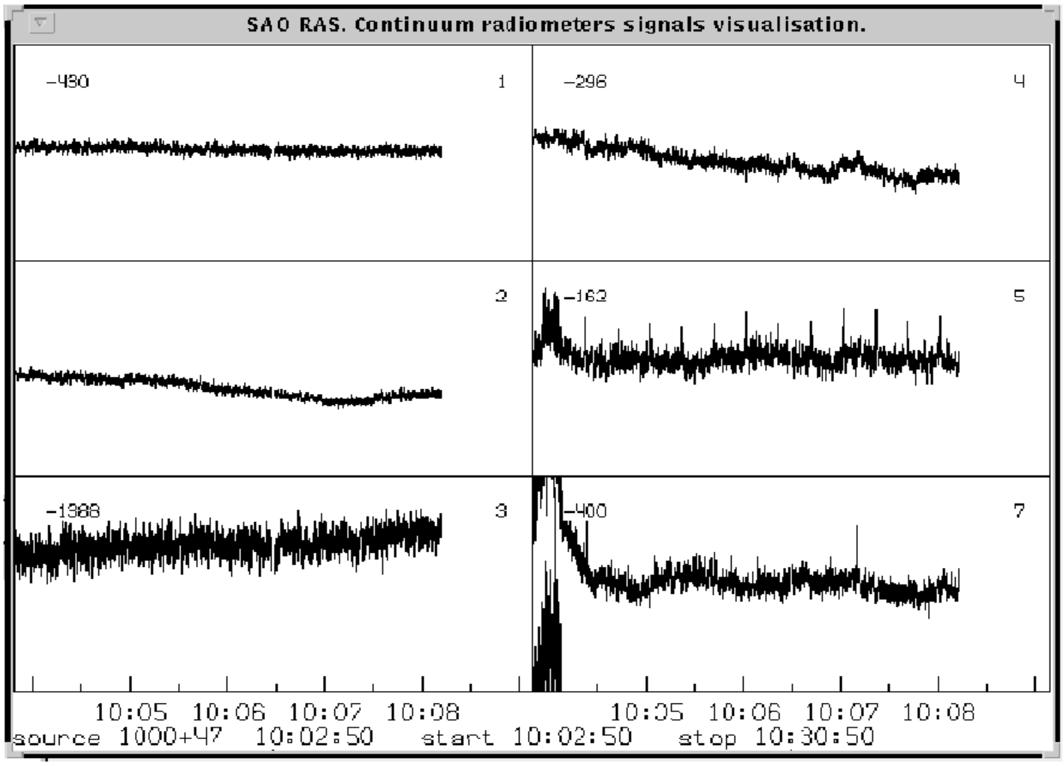

Рис. 3: Экран рабочей станции во время наблюдения

Корнильев К.Г., Шидяев Е.И.: 1992, под ред. Никитина В.Д., UNIX: Сетевые возможности, в сб. ТЭК т.18 - М., "ЭКО ТРЕНДЗ", 1-209

Черненков В.Н.: 1995a, Методика построения систем сбора данных на базе распределенной сетевой среды, Препр. Спец. астрофиз. обсерв., **109T**, 1-12

Черненков В.Н., Цыбулев П.Г.: 1995, Многопользовательская система сбора наблюдательных данных для радиометров непрерывного спектра на РАТАН-600. XXVI Радиоастрономическая конференция. Тезисы докл. 18-22 сент. С-Петербург, 389-390

Черненков В.Н.: 1995b, Методика подготовки и проведения наблюдений на РАТАН-600 в режиме теледоступа. XXVI Радиоастрономическая конференция. Тезисы докл. 18-22 сент. С-Петербург, 398