РАСПРЕДЕЛЕННАЯ ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНАЯ СИСТЕМА ДЛЯ СПЕКТРОСКОПИЧЕСКОГО АНАЛИЗА ПУЧКОВЫХ И ПЛАЗМЕННЫХ ОБЪЕКТОВ

С. Е. Гаврилов, С. А. Кипрушкин, Н. А. Королев, С. Ю. Курсков
Петрозаводский государственный университет

В работе рассмотрена созданная авторами распределенная информационно-измерительная система поддержки научно-образовательного процесса с обеспечением удаленного доступа к информационным и техническим ресурсам в сетях Интранет/Интернет. Отличительной особенностью этой системы является то, что она обеспечивает сетевую интеграцию автоматизированных исследовательских установок в естественнонаучных областях знаний и предоставляет коллективный доступ к их ресурсам в сетях, функционирующих на базе стека протоколов TCP/IP. Доступ к физическому оборудованию осуществляется с помощью серверов стандартных приборных интерфейсов (КАМАК, КОП), сервера доступа к микроконтроллерам MCS-96, а также коммуникационного сервера, интегрирующего серверы оборудования в единую информационную систему. Информационно-измерительная система предназначена для решения исследовательских задач в области оптической спектроскопии и поддержки образовательного процесса на физико-техническом факультете Петрозаводского государственного университета.

Введение

К современным системам автоматизации физического эксперимента предъявляются следующие требования:

1. Доступность уникального научного оборудования. Во-первых, это техническая возможность получить доступ к оборудованию, которое расположено на определенном расстоянии от исследователя, причем в удобное для исследователя время. Наиболее часто для этих целей используется управление научной аппаратурой через сеть Интернет посредством Web-сервера, работающего на компьютере, сопряженном с установкой; пользователь получает доступ к оборудованию с помощью стандартного Web-браузера. Во-вторых, это минимизация необходимого объема работ для организации доступа экспериментатора к измерительным и исполнительным устройствам установки.

2 Коллаборация. Современные системы автоматизации должны обеспечивать объединение усилий многих ученых при работе над проектом. Модульность построения, высокий уровень описания интерфейсов и простота реализации клиентского программного обеспечения позволяют объединять усилия исследователей для наращивания и развития системы при проведении каждого последующего эксперимента.

3. Развитие. В архитектуре системы должны быть заложены принципы развития в соответствии с развитием научного и методического обеспечения исследований. Автоматизированная система должна стимулировать исследователей включать новые методические наработки, реализованные в программных модулях, в состав системы для их дальнейшего общего использования.

4. Интеграция научных и образовательных функций. Возможность использования автоматизированной системы для образовательных целей обеспечивает непрерывность образовательного процесса (студенты, аспиранты, инженеры и научные сотрудники), когда студенты обучаются с использованием тех же самых инструментов, которые используются в научных исследованиях.

Таким образом, современная система автоматизации физического эксперимента должна не только обеспечивать управление конкретным экспериментом, но также, по возможности, должна удовлетворять и перечисленным требованиям.

Для разработки информационно-измерительных систем существуют различные программные средства. Распространенные инструментальные пакеты (например, пакеты фирмы National Instruments – LabWindows/CVI, LabVIEW, BridgeVIEW, а также системы визуализации измерительной информации (SCADA-системы), написанные с использованием инструментальных пакетов) обеспечивают в той или иной степени удаленное взаимодействие с физической аппаратурой, однако в этих системах оборудование подключено к тому компьютеру, на котором запущен инструментальный пакет. Это затрудняет использование различных приборных интерфейсов, связанных с отдельными подсистемами экспериментального комплекса и подключенных к разным компьютерам. Кроме того, хотя подобные пакеты имеют дружественный интерфейс и обычно обладают средствами визуального программирования, они не обеспечивают гибкости в построении системы и ее расширяемости. Поэтому использование подобных пакетов не всегда оправдано для создания систем сбора данных, контроля и управления экспериментом.

В качестве примера близкой по назначению к разработанной информационно-измерительной системе можно привести систему накопления данных MIDAS [1], созданную в Paul Scherrer Institute (Швейцария). Данная система предназначена для удаленного сбора экспериментальных данных в ядерной физике и физике элементарных частиц. Однако удаленный доступ в системе MIDAS построен на базе HTTP-протокола в комбинации с механизмом дистанционного вызова процедур. Недостатками этой системы является отсутствие мониторинга системы и невозможность зарезервировать какой-либо объект для работы, что необходимо в многопользовательской системе.

Структура системы

Распределенная информационно-измерительная система строится как централизованная система, ее структурная схема приведена на рис. 1.

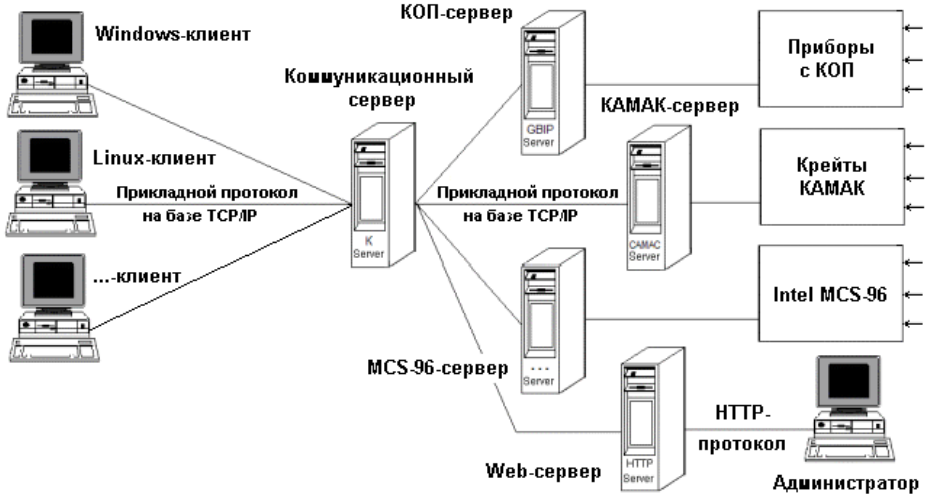


Рис. 1. Структура централизованной распределенной информационно-измерительной системы

Система состоит из следующих элементов: коммуникационного сервера, серверов оборудования (КАМАК-сервера [2], КОП-сервера [3] и сервера микроконтроллера семейства MCS-96 i80C196KC); программ-клиентов, осуществляющих сбор, накопление и обработку информации, а также управляющих ходом эксперимента; универсального протокола связи коммуникационного сервера с серверами оборудования и расширенного протокола связи коммуникационного сервера с клиентскими программами.

Программные модули информационной системы реализованы на языке программирования Java с использованием сокетов TCP. В отличие от сокетов UDP, TCP-сокеты обеспечивают надежное двунаправленное соединение между узлами Интернет. Для серверов используется класс ServerSocket, для клиентов – класс Socket из пакета java.net. Методы записи/чтения портов ввода/вывода для доступа к контроллерам приборных интерфейсов написаны на Си.

Кадр протокола обмена данными между коммуникационным сервером и серверами оборудования и кадр протокола обмена данными между коммуникационным сервером и клиентами приведены на рис. 2.

Рис. 2. Кадр протокола обмена данными между коммуникационным сервером и серверами оборудования (*а*) и кадр протокола обмена данными между коммуникационным сервером и клиентами (*б*)

В кадр запроса к серверу оборудования входят следующие поля: адрес ресурса – 8 байт, функция – 4 байта, тип данных – 1 байт, ключ – 4 байта, данные – 4 байта. Если значение поля «тип данныхо» равно нулю, то в поле «данные» находятся сами данные. Если «тип данных» равен 1, то поле «данные» содержит длину передаваемых данных в байтах (данные идут после основного кадра непрерывным потоком). Поле «ключ» зарезервировано для дальнейшего использования при шифровании кадров. Ответ сервера оборудования содержит в себе номер ошибки – 2 байта, тип данных – 1 байт, данные – 4 байта, информацию о состоянии системы – 4 байта. Поле «информация о состоянии системы» используется сервером оборудования для передачи содержимого регистров состояния приборного интерфейса пользователю. Кадр протокола обмена данными между коммуникационным сервером и клиентами содержит дополнительное поле (4 байта), в котором указывается псевдоним запрошенного сервера оборудования.

Сервер оборудования имеет типовую структуру и для разных приборных интерфейсов отличается лишь библиотеками методов, реализующих взаимодействие с конкретным при-

борным интерфейсом. Сервер оборудования представляет собой сервер последовательной обработки запросов. В его задачу входит определение допустимости для данного оборудования запрошенной функции и указанного адреса, передача запроса оборудованию, а также пересылка клиенту ответа или номера ошибки при возникновении исключительной ситуации. Таким путем достигается универсальность коммуникационного сервера, который работает с серверами оборудования по стандартному протоколу.

Структуру сервера оборудования рассмотрим на примере КАМАК-сервера. В состав КАМАК-сервера входят следующие классы:

- CamacS – основной класс сервера – реализует «прослушивание сети», подключение коммуникационного сервера, реализует алгоритм обслуживания клиента, в частности, обеспечивает проверку корректности значений входных параметров запроса клиента и выполнение команды управления аппаратурой;
- CserverProtocol – в данном интерфейсе определены коды операций, ошибок и другие константы коммуникационных протоколов (общий для системы);
- QueryToEServer – в данном классе определен объект «кадр запроса» к серверу и методы для работы с этим объектом (общий для системы);
- ReplyFromEServer – в классе определен объект «кадр ответа» и методы для работы с этим объектом (общий для системы);
- CamacLib – класс содержит библиотеку методов для работы с аппаратурой КАМАК. Для выполнения операций чтения-записи в порты ввода/вывода данный класс обращается к внешним методам, реализованным на Си. Основные методы класса: CmZ – безадресная КАМАК-команда ZERO, CmC – безадресная КАМАК-команда CLEAR, CmF – адресная КАМАК-команда управления, CmFWW – адресная КАМАК-команда записи, CmFRR – адресная КАМАК-команда чтения, CmQ – команда проверки L-запроса. Также в этом классе объявлены два внешних метода: outport – запись байта и inport – чтение байта из порта ввода/вывода.

В функции коммуникационного сервера входит обеспечение многопользовательского режима и распределение ресурсов, обеспечение безопасного доступа к аппаратуре экспериментального комплекса (создание условий работы, при которых пользователи не могут изменить данные других пользователей), мониторинг системы, обеспечение безопасности. Реализация многопользовательского режима достигается путем использования параллельных процессов с синхронизацией некоторых функций. Мониторинг системы включает хранение и предоставление по запросу администратора информации о пользователях, работающих в данный момент с аппаратурой. Безопасность системы может быть достигнута путем шифрования данных, которыми обмениваются коммуникационный сервер и сервер оборудования. Это представляется необходимым, так как существует вероятность подмены кадров коммуникационного сервера посторонним пользователем и, следовательно, получение несанкционированного доступа к информации всех клиентов, работающих с данным сервером оборудования. В кадрах обмена информацией для этих целей зарезервировано поле «ключ».

Коммуникационный сервер состоит из следующих основных классов:

- StartCServer – инициализация сервера. В этом классе осуществляется опрос всех серверов оборудования и установление связи с работающими серверами, после чего коммуникационный сервер переходит в режим ожидания связи с клиентами;
- ServerThread – класс, реализующий алгоритм обслуживания определенного клиента. В этом классе осуществляется обработка команд клиента, формируются запросы к серверам оборудования, обновляется информация об используемых модулях (приборах), подключенных к соответствующему приборному интерфейсу. Обращение к портам ввода/вывода осуществляется потоком из критической секции;

- CserverProtocol – интерфейс, который определяет коды операций, ошибок и другие константы протоколов;
- MainServInf – класс для хранения информации о серверах оборудования, входящих в информационную систему. Данные содержатся в виде набора записей, включающих в себя следующие поля: IP адрес сервера оборудования; номер порта, на котором сервер оборудования ожидает связи с клиентом (коммуникационным сервером), его псевдоним, сокет (если связь установлена) и состояние сервера оборудования;
- MainClientInfo – класс, отвечающий за мониторинг системы. Используется для хранения информации о пользователях, работающих в данный момент с исследовательской аппаратурой. Информация включает в себя IP адрес клиента, идентификационный номер клиента и занимаемые им ресурсы (псевдоним сервера оборудования и адрес ресурса).

В классах ReplyFromCServer, QueryToCServer, ClientReply, ClientQuery определены «кадры запроса» и «кадры ответа» соответствующих протоколов для обмена с серверами оборудования и клиентами, а также методы для работы с ними.

Коммуникационный сервер работает следующим образом. После запуска коммуникационный сервер считывает информацию о доступных серверах оборудования из конфигурационного файла (IP-адрес, номер порта, псевдоним). Далее последовательно устанавливаются постоянные соединения с серверами оборудования. Сервер, с которым соединение не было установлено, помечается как недоступный в данный момент. Попытка установления связи с ним будет повторена при обращении к нему любого клиента. После инициализации серверов оборудования коммуникационный сервер переходит в режим ожидания связи с клиентами. При установлении связи сервер генерирует параллельный процесс обслуживания клиента и присваивает этому процессу уникальный номер CID – Client ID (не равный нулю). Обмен информацией с клиентом осуществляется в режиме запрос–ответ по расширенному протоколу. Кадры (как запрос, так и ответ) представляют собой кадры сервера оборудования плюс псевдоним сервера оборудования (4 байта). Для работы с оборудованием клиент должен зарезервировать ресурс командой CS_GETRESOURCE, указав адрес ресурса и псевдоним сервера оборудования. По запросу клиента коммуникационный сервер формирует запрос к серверу оборудования с функцией CS_CHECKRESOURCE для проверки корректности адреса ресурса. Если ответ не содержит ошибок, коммуникационный сервер выделяет ресурс клиенту. В системе доступа к оборудованию используется иерархическая система адресов (для аппаратуры КАМАК: номер крейта, адрес станции, субадрес); полный адрес ресурса составляет 8 байт. Клиент упаковывает адреса в эти 8 байт, а сервер оборудования выполняет обратную операцию. После того, как ресурс будет выделен, клиент может начать работу. Завершение сеанса связи происходит по команде CS_QUIT. Все функции, использующиеся в системе (служебные функции), имеют номера с FFFFFFFFh и ниже. Для работы с оборудованием используются функции, начиная с нуля. Коды ошибок лежат в диапазоне от 0 до FFFFh. Если во время работы связь с каким-либо из серверов оборудования будет разорвана, то при следующем обращении какого-либо клиента к этому серверу коммуникационный сервер попытается восстановить связь.

В системе также предполагается режим суперпользователя (администратора). Последний подключается к серверу как обычный клиент, но с паролем в поле «данные» и указанием длины пароля в поле «ключ». После проверки пароля данному клиенту присваивается CID равный нулю, по которому разрешается выполнение дополнительных функций, таких как просмотр информации о клиентах и используемых ресурсах, регистрация клиента в системе и его удаление, а также освобождение ресурса.

Криптографическая защита данных в системе

Защита информации в системе основана на криптографическом расширении JCE 1.2 пакета Java 2 Platform Standard Edition v1.4 и пакете Cryptix 3.2 [4]. Пакеты JCE 1.2 и Cryptix 3.2 используются для разработки алгоритмов шифрования, создания и согласования ключей, а также аутентификации.

Все участники межсетевого взаимодействия имеют в своем распоряжении следующие классы:

- генератор пары ключей (открытого и закрытого ключа). В системе используются ключи алгоритма RSA в кодировке RAW, размером 512 бит. Ключи хранятся в файлах;
- классы шифрования и дешифрования произвольной байт-последовательности по алгоритму RSA. Зашифрованная последовательность кратна 64 байтам;
- классы шифрования и дешифрования произвольной байт-последовательности по алгоритму Rijndael. Метод шифрования CFB (Cipher Feedback – обратная связь по шифру). Открытое сообщение шифруется порциями длиной до 64 бит. При этом предыдущая порция зашифрованного сообщения объединяется (путем выполнения операции «исключающего ИЛИ») со следующей порцией открытого сообщения. В системе шифрование идет однобайтовыми блоками;
- класс цифровой подписи (используется алгоритм RSA и алгоритм дайджеста MD5). Цифровая подпись предназначена для аутентификации источника сообщения, проверки целостности сообщения и обеспечения невозможности отказа от факта подписи конкретного сообщения. Цифровая подпись вычисляется для произвольной последовательности байт. Размер подписи – 64/65 байт;
- класс вычисления дайджеста сообщения (используется однонаправленная хэш-функция MD5), размер дайджеста – 128 бит. Дайджест сообщения гарантирует целостность данных;
- класс MAC-кода (Message Authentication Code) – дайджест сообщения, который вычисляется с помощью шифрования с секретным ключом. Применяется алгоритм HMAC-MD5; размер MAC-кода – 128 бит. MAC-код выполняет функцию цифровой подписи в процессе работы системы в защищенном режиме;
- класс шифрования и дешифрования файлов ключей на базе пароля, вводимого пользователем. В этом классе используется алгоритм PBEWithMD5AndDES (спецификация JCE 1.2). Метод шифрования CBC (Cipher Block Chaining – сцепление зашифрованных блоков); шифрование осуществляется 8-байтовыми блоками. Если блок меньше 8 байт то, он дополняется по стандарту PKCS#5.

Кроме этого, коммуникационный сервер имеет еще один дополнительный класс для генерации секретных ключей. В системе применяются 128-битные ключи на основе симметричного алгоритма Rijndael. Данные ключи используются как временные, срок действия – один месяц.

Каждая сторона межсетевого взаимодействия генерирует пару ключей. Все открытые ключи пересылаются на коммуникационный сервер, который, в свою очередь, предоставляет клиентам и серверам оборудования свой открытый ключ. Открытые ключи размещаются на сайте администратора системы для сверки с полученными ключами.

В системе безопасности предусмотрено три режима работы, которые зависят от степени значимости решаемых задач. Первый режим – это открытая работа системы, когда криптографическая защита полностью отключена. Данный режим предназначен для тестирования и отладки системы. Второй режим – это реальная работа, которая требует обеспечения целостности и подтверждения неаннулируемости данных, а также аутентификации прав доступа,

но не конфиденциальности. И последний, третий режим – это закрытая работа, когда все данные передаются только в зашифрованном виде с дайджестом или цифровой подписью. Здесь конфиденциальность информации ставится на первое место.

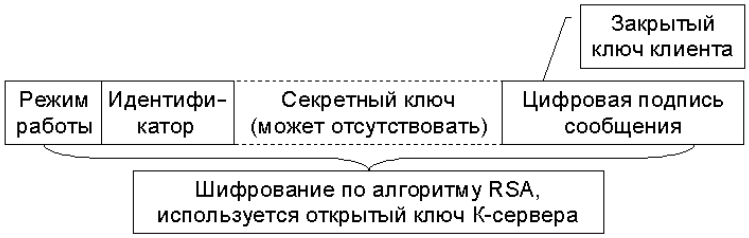
Рассмотрим более подробно два последних режима.

При инициализации коммуникационный сервер производит расшифровку хранилища ключей. Хранилище ключей содержит закрытые и секретные (временные) ключи. Расшифровка производится после ввода соответствующего пароля администратором. Коммуникационный сервер проверяет, не истек ли срок действия секретных ключей, и, если он истек, заменяет их. Для клиентов и серверов оборудования используются разные секретные ключи. Принадлежность ключа определяется уникальным идентификатором (номером), присваиваемым клиентам и серверам оборудования. Информация о разграничении прав доступа клиентов к серверам оборудования хранится в специальном файле, в котором содержится (для каждого сервера оборудования) список идентификаторов клиентов, имеющих право доступа к серверу. Аутентификация прав доступа выполняется коммуникационным сервером после фазы идентификации и сверки секретных ключей перед отправкой запроса к серверу оборудования.

После инициализации коммуникационный сервер осуществляет опрос готовности серверов оборудования к работе. Сервер оборудования при подключении к нему коммуникационного сервера отвечает зашифрованным сообщением, которое содержит идентификатор, секретный ключ или информацию об его отсутствии, подписанную цифровой подписью сервера оборудования. Шифрование осуществляется с использованием открытого ключа коммуникационного сервера. Далее коммуникационный сервер производит расшифровку и верификацию цифровой подписи. Если все в порядке, то проверятся секретный ключ. Если секретный ключ отсутствует или устарел, то серверу оборудования отправляется новый ключ. При успешном завершении проверки коммуникационный сервер посылает сообщение об отсутствии ошибок. Его ответ шифруется открытым ключом сервера оборудования с добавлением цифровой подписи коммуникационного сервера. Следующая попытка установления связи будет произведена при обращении клиента к серверу оборудования.

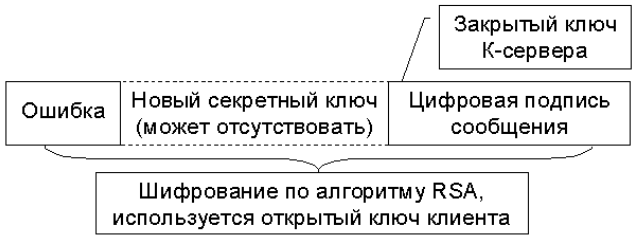
При запуске клиента или сервера оборудования пользователь или администратор должен ввести пароль для расшифровки закрытого и секретного ключей, хранящихся в файлах.

Когда клиент устанавливает соединение с коммуникационным сервером, он передает ему зашифрованное сообщение, содержащее режим работы, идентификатор и секретный ключ или информацию об его отсутствии, подписанную цифровой подписью клиента (рис. 3). Шифрование сообщения осуществляется с использованием открытого ключа коммуникационного сервера. Коммуникационный сервер производит расшифровку и верификацию цифровой подписи. При успешной верификации устанавливается режим работы, запрошенный клиентом. Если указан открытый режим, то далее все взаимодействие осуществляется без использования средств криптографической защиты. Если указан защищенный или закрытый режим, то проверяется секретный ключ клиента. При отсутствии ключа (или срок действия его истек), клиенту отправляется новый ключ. В противном случае клиенту посылается сообщение о том, что ошибок нет. Ответ коммуникационного сервера шифруется открытым ключом клиента с добавлением цифровой подписи коммуникационного сервера.



Режим работы – 1 байт;
Идентификатор – 4 байта;
Секретный ключ – размер файла ключа, 191 байт для алгоритма Rijndael;
Цифровая подпись – 64/65 байт.

а)



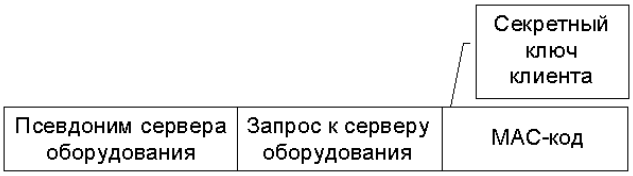
Ошибка – 4 байта;
Новый секретный ключ – размер файла ключа 191, байт для алгоритма Rijndael;
Цифровая подпись – 64/65 байт.

б)

Рис. 3. Первое сообщение клиента коммуникационному серверу (*а*), ответ коммуникационного сервера (*б*)

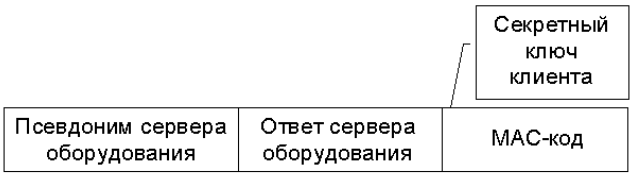
В защищенном режиме все запросы от клиента к коммуникационному серверу идут с MAC-кодом, то есть к обычному запросу добавляется 16-байтный код (рис. 4). Коммуникационный сервер проверяет MAC-код, используя секстетный ключ, присвоенный данному клиенту. Далее коммуникационный сервер пересылает запрос клиента серверу оборудования, определив нужный сервер по псевдониму, содержащемуся в запросе. К запросу вычисляется и добавляется MAC-код, при этом используется секретный ключ, присвоенный серверу оборудования. Ответ сервера оборудования проходит такую же процедуру, как и запрос от клиента, только в обратном порядке. Отметим, что режим работы сервер оборудования определяет динамически. Сначала запрос проверяется по алгоритму защищенного режима, при возникновении ошибки – по алгоритму закрытого режима и далее по алгоритму открытого режима. Успешная проверка позволяет определить режим работы системы.

В закрытом режиме все сообщения между клиентом, коммуникационным сервером и серверами оборудования передаются в зашифрованном виде. В этом случае к исходному сообщению добавляется 16-байтный дайджест, и весь запрос шифруется по алгоритму Rijndael (рис. 5).



Псевдоним сервера оборудования – 4 байта;
Запрос к серверу оборудования – 21 байт;
MAC-код – 16 байт.

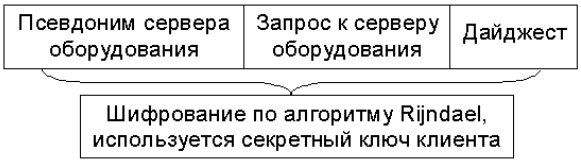
а)



Псевдоним сервера оборудования – 4 байта;
Ответ сервера оборудования – 13 байт;
MAC-код – 16 байт.

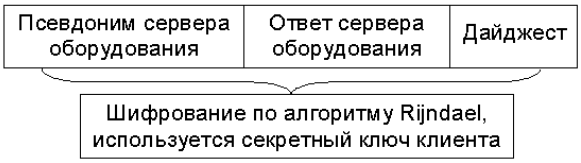
б)

Рис. 4. Запрос клиента к коммуникационному серверу в защищенном режиме (*а*), ответ коммуникационного сервера (*б*)



Псевдоним сервера оборудования – 4 байта;
Запрос к серверу оборудования – 21 байт;
Дайджест – 16 байт.

а)



Псевдоним сервера оборудования – 4 байта;
Ответ сервера оборудования – 13 байт;
Дайджест – 16 байт.

б)

Рис. 5. Запрос клиента к коммуникационному серверу в закрытом режиме (*а*), ответ коммуникационного сервера (*б*)

Опыт использования системы

Рассмотренная распределенная информационно-измерительная система используется для анализа пучковых и плазменных объектов методами оптической спектроскопии. В частности, с ее помощью ведутся исследования процессов возбуждения при атом-атомных столкновениях с участием атомов инертных газов [5].

На рис. 6 показан спектр плазменной струи микроплазмотрона, полученный с использованием распределенной информационно-измерительной системы. Струя плазмотрона направлена вдоль оптической оси монохроматора МДР-2; ширина аппаратной функции прибора на полувысоте – 0.2 нм. Буферный газ – He, избыточное давление – 1 атм; напряжение на разряде – 700 В, ток – 25 мА. Измерения проводились в локальной вычислительной сети ПетрГУ.

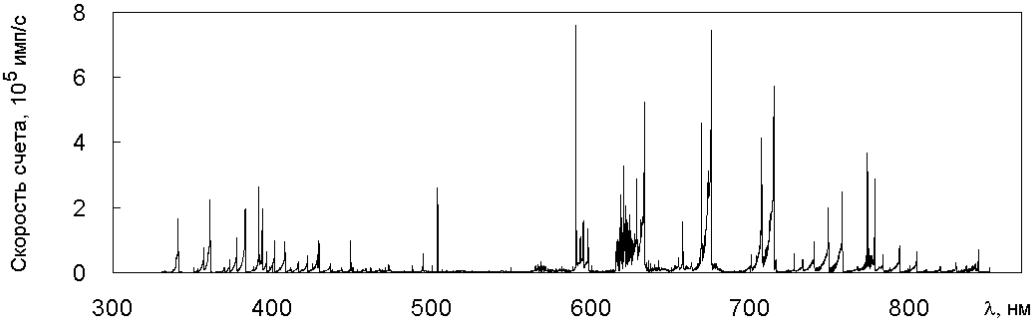


Рис. 6. Оптический спектр плазменной струи микроплазмотрона

Заключение

Отличительной особенностью разработанной распределенной информационной системы является то, что она позволяет объединить различные приборные интерфейсы с выделенными для них управляющими компьютерами в единую сеть, функционирующую на базе стека протоколов TCP/IP.

Также следует отметить, что в информационной системе значительно упрощено подключение нового исследовательского оборудования. В отличие от распространенных систем добавление к системе нового приборного интерфейса с подсоединенной к нему экспериментальной аппаратурой сводится к регистрации соответствующего сервера оборудования в коммуникационном сервере, после чего клиентские программы получают доступ к этой аппаратуре. Кроме того, перенос функций взаимодействия с клиентскими программами с серверов оборудования на коммуникационный сервер позволил значительно упростить структуру сервера оборудования и ускорить его разработку.

К достоинствам системы следует отнести и то, что программа, управляющая экспериментом, выполняется не на удаленном компьютере (как при использовании Web-технологий), а на пользовательском, который связан с системой через глобальную сеть. Такая организация взаимодействия элементов многопользовательской системы существенно повышает ее надежность.

Кроме того, структура коммуникационного сервера обеспечивает одновременный доступ нескольких пользователей к исследовательским комплексам или их подсистемам; при этом устройства, используемые одним клиентом, на время обмена данными защищаются от других клиентов.

Достоинства предложенной архитектуры особенно отчетливо проявляются при использовании распределенной системы в образовательных целях. Во-первых, упрощенная процедура создания сервера оборудования обеспечивает легкость включения в учебный процесс уникальной научной аппаратуры. Во-вторых, поскольку программы, управляющие ходом эксперимента, выполняются на компьютере пользователя, они могут быть модифицированы обучаемым в соответствии с поставленной задачей. В-третьих, возможна организация не только лабораторных работ с жестко заданными алгоритмом выполнения, но и полноценных научных экспериментов.

В заключение отметим, что использование криптографических средств защиты информации в системе гарантирует целостность, неаннулируемость и конфиденциальность данных в условиях многопользовательского сетевого доступа к ресурсам распределенной информационно-измерительной системы.

Благодарности

Мы благодарим профессора А. Д. Хахаева и И. П. Шибаева за содействие при выполнении данной работы, а также студентов А. С. Кашубу, О. А. Мурсалимова, Н. Г. Носович и В. В. Семина за участие в создании распределенной информационно-измерительной системы. Микроплазмотрон для измерений предоставлен В. А. Гостевым.

Работа выполнена при поддержке Российского фонда фундаментальных исследований (грант № 02–07–97503), а также Американского фонда гражданских исследований и развития (CRDF) и Министерства образования РФ (проект PZ–013–02).

Литература:

1. http://midas.psi.ch/
2. Zhiganov E. D., Kiprushkin S. A., Kurskov S. Yu, Khakhaev A. D. CAMAC Server for Remote Access to Physical Equipment // Learning and Teaching Science and Mathematics in Secondary and Higher Education. Joensuu, Joensuu University Press, 2000. P. 170–173.
3. Кашуба А. С., Кипрушкин С. А., Курсков С. Ю. Сервер канала общего пользования распределенной информационной системы поддержки научных исследований в области оптической спектроскопии // Технологии информационного общества – Интернет и современное общество 2002: Материалы V Всерос. Объединенной конф. СПб. / С.-ПбУ. С-Пб., 2002. С.104–105.
4. http://java.sun.com/; http://www.cryptix.org/
5. Kurskov S. Yu., Khakhaev A. D. Excitation of Ar I atoms into $3p^5np$ states ($4 \le n \le 6$) in binary Ar–Ar collisions // Northern optics 2003 (16–18 June 2003, Espoo, Finland). Helsinki: Helsinki University of Technology, 2003. P. P012.