УДК 004.738
ББК 30

ОРГАНИЗАЦИЯ ЗАЩИЩЕННОГО HTTP-ВЗАИМОДЕЙСТВИЯ В МУЛЬТИ-СЕТЕВОЙ СРЕДЕ

Асратян Р. Э.[1], Лебедев В. Н.[2]
(*Учреждение Российской академии наук Институт проблем управления РАН, Москва*)

Рассматривается проблема организации информационного взаимодействия в территориально-распределенных информационно-управляющих системах, разрабатываемых на базе технологии .NET и ориентированных на работу в сложной мульти-сетевой среде, включающей множество частных сетей предприятий. Предлагается подход к решению этой проблемы, основанный на создании защищенных межсетевых каналов на базе системы прокси-серверов, оснащенных средствами маршрутизации HTTP-взаимодействий по символическим именам сетевых узлов и ресурсов.

Ключевые слова: распределенные системы, интернет-технологии, сетевые протоколы, веб-сервисы

1. Введение

В последние годы наблюдается все больший рост интереса к методам построения территориально-распределенных систем, обеспечивающих межведомственную и даже транснациональную интеграцию информационных ресурсов. Это, в свою очередь, привлекло внимание к методам организации информаци-

[1] *Рубен Эзрасович Асратян*, *кандидат технических наук*, *доцент* (*rea@ipu.ru*).
[2] *Виталий Николаевич Лебедев*, *кандидат технических наук*, *доцент* (*lebvini@ipu.ru*).

онного взаимодействия в сложной, мульти-сетевой среде, включающей множество частных сетей предприятий, соединенных ведомственными и интернациональными глобальными сетями.

Разработчики информационных систем, ориентированных на работу в таких средах, обычно сталкиваются с двумя противоречивыми требованиями, прямо связанными с обеспечением информационной безопасности:

- все информационные ресурсы системы (серверы приложений, серверы баз данных) должны быть «спрятаны» в частных сетях предприятий с исключением возможности прямого соединения с ними из глобальных сетей;
- должен быть обеспечен контролируемый доступ к каждому из этих ресурсов с клиентских рабочих мест, независимо от того, размещены ли они в разных частных сетях или в одной (и даже в том случае, если эти частные сети подсоединены к разным глобальным сетям).

Эти требования могут быть удовлетворены только путем решения двух взаимосвязанных проблем: проблемы маршрутизации данных в мульти-сетевой среде и проблемы их защиты от несанкционированного доступа. Данная работа главным образом посвящена методам решения проблемы маршрутизации информационных запросов в разработках распределенных мульти-сетевых систем на базе технологии *.NET* и сетевого протокола *HTTP*/*SOAP* [6, 9].

Долгое время казалось, что имеющиеся в Интернете средства *IP*-маршрутизации (вместе с такими мощными механизмами, как *DNS*, *NAT*, протоколы обмена маршрутной информацией *IGP*, *EGP* и т.п.) полностью решают проблему маршрутизации данных в сети вообще и в распределенных системах в частности. В самом деле, зачем нужно придумывать средства маршрутизации на транспортном уровне или уровне приложения, если они уже имеются на базисном сетевом уровне стека (иерархии) протоколов *TCP*/*IP* [4, 7]?

Возникшие в последние годы новые тенденции и приоритеты в подходах к построению распределенных систем показывают, что это не совсем так. Повышение внимания к структурам

множественных частных сетей все чаще приводит к ситуации, в которой затруднительно или даже невозможно обеспечить маршрутизацию и управление доступом на сетевом уровне – уровне управления *IP*-датаграмм. В такой ситуации могут оказаться востребованными средства маршрутизации по интернет-именам (или корпоративным именам) ресурсов, реализуемые на уровне приложения стека протоколов *TCP*/*IP*.

2. Средства поддержки взаимодействий между частными сетями

Как известно, Интернет является сетью с коммутацией пакетов и изначально построен как «сеть сетей» – его основу составляют локальные сети предприятий, соединенные с помощью межсетевых маршрутизаторов – выделенных компьютеров, присоединенных к двум или более сетям одновременно [4, 7]. Когда программа, работающая в одной локальной сети, открывает соединение и начинает информационный обмен с программой, работающей в другой локальной сети, возникает двунаправленный поток *IP*-датаграмм (пакетов), проходящий через один, два или целую цепочку маршрутизаторов. Хотя на пользовательском уровне используются символические (доменные) имена сетевых узлов и информационных ресурсов, обычная схема организации взаимодействия в Интернете включает трансляцию доменных имен в *IP*-адреса еще до открытия сетевого соединения. Собственно открытие соединения и маршрутизация данных целиком базируются на *IP*-адресах, которые заносятся в заголовок каждой датаграммы. Разумеется, уникальность *IP*-адреса любого узла сети в пределах всего Интернета является основой для процесса маршрутизации.

Разработчики стека (иерархии) протоколов *TCP*/*IP* придерживались твердого правила: любой сетевой механизм должен быть реализован на возможно более низком уровне стека для максимальной разгрузки вышестоящих уровней. Например, реализованные на сетевом уровне средства маршрутизации данных в сети (*IP*-датаграмм) стали наиболее фундаментальным

механизмом в Интернете и позволили избавить все протоколы транспортного и прикладного уровня от необходимости решения этой задачи. (Фактически только средства электронной почты сегодня имеют еще и свой собственный прикладной механизм маршрутизации данных, сохранившийся еще с «доинтернетовских» времен.) Тем не менее, данный подход порождает проблемы, связанные с так называемыми частными сетями предприятий

Главная особенность частных сетей заключается в их изолированности от всемирной сети, т.е. в невозможности прямого взаимодействия между программами, если одна из них работает в частной сети, а другая вне нее, без применения специальных средств (даже при наличии физического соединения через маршрутизатор). Сама постановка вопроса о маршрутизации *IP*-датаграмм между Интернетом и частной сетью или между различными частными сетями является в значительной степени бессмысленной, так как частные сети используют одинаковые диапазоны *IP*-адресов, т.е. один и тот же сетевой адрес может использоваться во множестве частных сетей (особенно если они администрируются независимо). Первоначально применение частных сетей объяснялось в основном дефицитом уникальных *IP*-адресов, но с течением времени на первый план вышли соображения информационной безопасности.

Рассмотрим в качестве примера мульти-сетевую среду, изображенную на рис. 1. Если сервер WS_2 в частной локальной сети LAN_2 имеет такой же *IP*-адрес, как сервер WS_3 в частной локальной сети LAN_3, то одновременное взаимодействие с ними из рабочей станции PC, размещенной в частной локальной сети LAN_1, попросту невозможно.

Для организации взаимодействия между частными сетями традиционно используются две основные сетевые технологии.

- *NAT* (*Network Address Translation*) [7, 8] – основная технология наведения «информационных мостов» между частными сетями и внешним миром. Технология основана на замене *IP*-адресов, содержащихся в заголовках *IP*-датаграмм, при прохождении ими сетевых маршрутизаторов. Чаще всего она использу-

ется для обеспечения возможности обращения клиентов из частной сети к внешним серверам. Однако эта же технология может быть использована и для решения обратной задачи – доступа извне к серверам, размещенным в частной сети. Эта задача решается путем присваивания таким серверам «внешних» уникальных *IP*-адресов в дополнение к «внутренним» частным *IP*-адресам, что делает их доступными для внешних клиентов (т.е. клиентов, размещенных вне частной сети). Главный недостаток технологии лежит в области информационной защиты: присваивание внешнего *IP*-адреса открывает сервер для внешних атак, а передаваемые данные остаются незащищенными от несанкционированного доступа. Применение дополнительных средств защиты типа межсетевых экранов [8] способно лишь частично исправить положение. (Все сказанное можно в полной мере отнести и к технологии *Port forwarding*, основанной на присваивании «внешних» номеров портов внутренним ресурсам частной сети.)

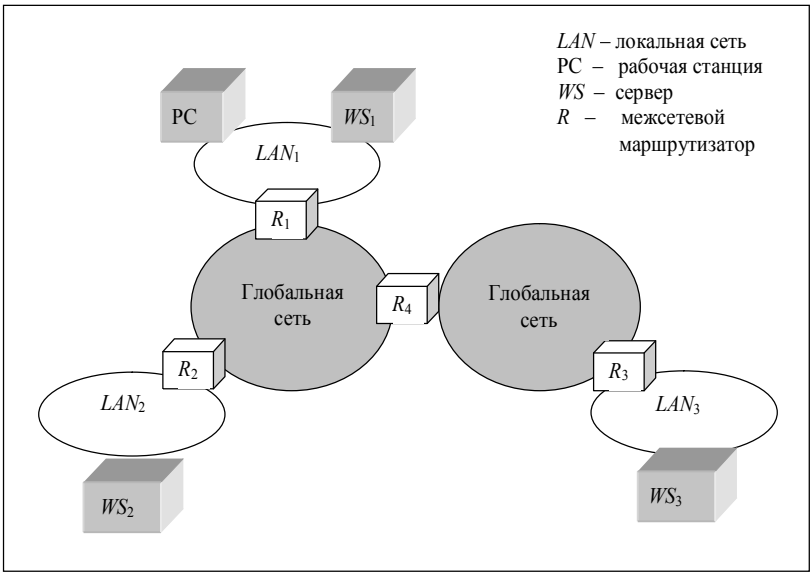

Рис. 1. Пример мульти-сетевой среды

- *VPN* (*Virtual Public Network*) [7, 8] – технология построения защищенных каналов взаимодействия через общедоступную глобальную сеть, позволяющая связать две или более удаленных локальных сетей в одну территориально-распределенную частную сеть с единым жестким администрированием, гарантирующем уникальность *IP*-адресов в пределах всей распределенной сети. Технология основана на передаче *IP*-датаграмм «поверх» другого протокола, оснащенного средствами информационной защиты (например, *PPP* или *IPSec*). В отличие от *NAT*, технология *VPN* содержит все необходимые средства защиты данных от несанкционированного доступа. Однако при наличии большого числа частных сетей, принадлежащих различным организациям, требование жесткого централизованного администрирования становится практически нереализуемым.

Можно констатировать, что вышеописанные технологии не вполне адекватны задаче организации информационного взаимодействия в мульти-сетевой среде. Поэтому на практике в рассматриваемой ситуации разработчики чаще всего прибегают к эвристическим приемам, привязанным к особенностям конкретной системы. Например, в разработках на базе технологии веб-сервисов [9] создают специальные межсетевые сервисы-ретрансляторы, приспособленные к конкретным спецификациям удаленных запросов.

Далее мы рассмотрим новое решение данной задачи, основанное на дополнении механизма маршрутизации *IP*-датаграмм по *IP*-адресу (реализуемого на сетевом уровне иерархии протоколов), новым механизмом маршрутизации, реализуемом на уровне протокола приложения (в данном случае – на уровне *HTTP*). Этот механизм маршрутизации основан не на *IP*-адресах, а на символических именах серверов и информационных ресурсов, что обеспечивает новые возможности при работе в мульти-сетевой среде. Разумеется, между двумя механизмами есть определенная аналогия, так как оба они используют адресную информацию, сопровождающую передаваемые данные: или IP-адрес, содержащийся в заголовке *IP*-датаграммы, или символическое имя сетевого информационного ресурса (*URL – Univer-*

sal Resource Locator), содержащееся в заголовке *HTTP*-запроса. Идея подхода заключается в создании специальных каналов межсетевого взаимодействия, основанных на маршрутизации запросов по символическим именам узлов и ресурсов.

3. Маршрутизация HTTP-взаимодействий по символическим именам

Описываемый подход к построению каналов межсетевого взаимодействия основан на технологии прокси-серверов. А эта технология, в свою очередь, основана на том, что в заголовке *HTTP*-запроса всегда сохраняется символическое наименование адресуемого сетевого информационного ресурса (*URL*).

Прокси-сервером называется сервер-посредник между *HTTP*-клиентом и *HTTP*-сервером, который позволяет осуществлять информационное взаимодействие в ситуации, когда прямое соединение между ними невозможно или нежелательно. Построение канала межсетевого взаимодействия основано на последовательном (каскадном) включении специализированных прокси-серверов и реализуется с использованием следующих принципов.

- Каждая частная сеть, могущая быть источником межсетевых *HTTP*-запросов (т.е. запросов, направленных в другие сети), оснащается так называемым выходным прокси-сервером, осуществляющим «перехват» запросов для передачи их во внешние сети. Каждая частная сеть, содержащая информационные ресурсы, доступные для внешних источников запросов, оснащается так называемым входным прокси-сервером, осуществляющим прием запросов из внешних сетей. Прокси-серверы размещаются на сетевом маршрутизаторе частной сети, причем выходной прокси-сервер «прослушивает» запросы на внутреннем сетевом интерфейсе маршрутизатора, а входной – на внешнем.
- Работа каждого прокси-сервера управляется соответствующей таблицей маршрутизации запросов. Таблица маршрутизации выходного прокси-сервера ставит в соответствие символическим именам удаленных узлов и/или информационных

ресурсов *IP*-адреса (или интернет-имена) входных прокси-серверов соответствующих частных сетей (точнее – адреса их внешних сетевых интерфейсов). Таблица маршрутизации входящего прокси-сервера связывает символические имена удаленных узлов и/или информационных ресурсов с *IP*-адресами или *URL* адресуемых ресурсов в частной сети.

- Обработка *HTTP*-запроса в межсетевом канале включает несколько этапов. Вначале каждый запрос попадает на вход выходного прокси-сервера (это обеспечивается или системной настройкой, общей для всех *HTTP*-клиентов в каждой рабочей станции, или с помощью соответствующей настройки маршрутизатора частной сети – технология «прозрачного» прокси). Дальнейшую «судьбу» запроса определяет анализ *URL* адресуемого информационного ресурса, содержащийся в его заголовке. Этот *URL* может быть основан или на анонсированном корпоративном, ведомственном или интернациональном имени узла или ресурса. Сопоставив это имя с таблицей маршрутизации, выходной сервер находит *IP*-адрес удаленного входного прокси-сервера, в котором должна быть продолжена обработка запроса. Продолжение обработки включает определение *IP*-адреса или *URL* адресуемого ресурса в частной сети, организацию обращения к ресурсу, получение *HTTP*-ответа и отправку его удаленному клиенту по тому же пути, но в противоположную сторону. Таким образом, маршрутизация всегда осуществляется, по крайней мере, в два этапа: сначала с точностью до удаленной частной сети (входного прокси-сервера), а затем – с точностью до конкретного ресурса в удаленной сети (если участвующие в обработке частные сети подсоединены к разным глобальным сетям, то могут понадобиться дополнительные прокси-серверы, размещенные на границах глобальных сетей).
- В течение всего времени обработки запроса вплоть до получения ответа соединение между клиентом и выходным прокси-сервером остается открытым, т.е. взаимодействие осуществляется без разрушения режима *on-line*.

Легко видеть, что поэтапная маршрутизация запроса оставляет значительную свободу в администрировании каждой част-

ной сети. Например, при перемещении информационного ресурса с одного узла на другой внутри сети администратору достаточно откорректировать таблицу маршрутизации собственного входного прокси-сервера (не ставя в известность об этой коррекции никого из удаленных клиентов).

Таблица маршрутизации подготавливается в форме текстового файла, каждая строка которого сопоставляет или имя удаленного узла или ресурса с его *IP*-адресом или же «внешнее» имя сетевого ресурса с альтернативным «внутренним» именем. Строку таблицы можно рассматривать как определение имени удаленного узла или ресурса. Поясним это на следующем коротком примере.

Рассмотрим ситуацию с тремя локальными частными сетями: LAN_1, LAN_2 и LAN_3 (см. рис. 2). Каждая из частных сетей оснащена выходным (*O*) и входным (*I*) прокси-серверами, обеспечивающими взаимодействие LAN_1 с LAN_2 и LAN_3.

Предположим, что в LAN_2 на узле с *IP*-адресом 192.168.0.1 установлен веб-сервер, в котором имеется веб-сервис с *URL* http://192.168.0.1/Vehicle/Service.asmx, а в таблице маршрутизации входного прокси-сервера имеется строка определения «внешнего» имени ресурса:

http://federal.vehicledb.ws
http://192.168.0.1/Vehicle/Service.asmx

Эта строка означает, что упомянутый веб-сервис анонсирован для внешних клиентов под именем http://federal.vehicledb.ws.

Предположим также, что в LAN_3 на узле с тем же *IP*-адресом 192.168.0.1 установлен веб-сервер, в котором имеется веб-сервис с *URL* http://192.168.0.1/Person/Service.asmx, а в таблице маршрутизации входного прокси-сервера имеется строка определения «внешнего» имени ресурса:

http://omsk.persondb.ws
http://192.168.0.1/Person/Service.asmx

Эта строка означает, что упомянутый веб-сервис анонсирован для внешних клиентов под именем http://omsk.persondb.ws.

Внесем в таблицу маршрутизации выходного прокси-сервера в LAN_1 две строки:

federal.vehicledb.ws 197.20.100.10
omsk.persondb.ws 194.88.200.11

где 197.20.100.10 и 194.88.200.11 – *IP*-адреса входных прокси-серверов LAN_2 и LAN_3 соответственно. С этого момента обращение клиентской программы из LAN_1 по *URL* **http://federal.vehicledb.ws** приведет к вызову веб-сервиса в LAN_2, а обращение по *URL* **http://omsk.persondb.ws** приведет к вызову веб-сервиса в LAN_3. Другими словами, совпадение *IP*-адресов обоих серверов в частных сетях в данном случае не является препятствием для одновременной работы с обоими веб-сервисами. Подчеркнем три важных обстоятельства.

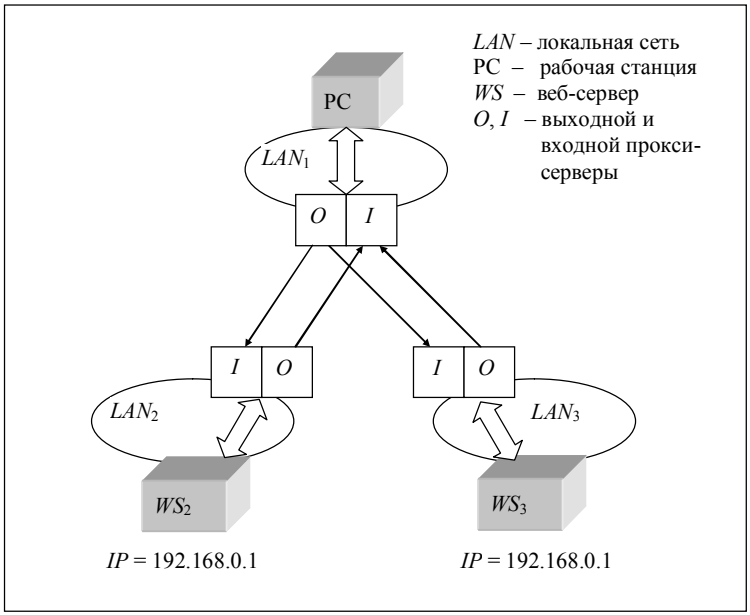

Рис. 2. Каналы межсетевого взаимодействия

- Обращение к удаленному ресурсу по имени, содержащему дополнительные спецификации, например, **http://federal.vehicledb.ws?wsdl,** в итоге приведет к обращению к ресурсу **http://192.168.0.1/Vehicle/Service.asmx?wsdl**. Другими словами, межсетевой канал вполне можно использовать не

только для обращения к методам удаленного веб-сервиса, но и для считывания метаданных и спецификаций функций в процессе проектирования клиентских приложений.

- В выходном прокси-сервере допускается определение маршрута для группы символических имен одной строкой. Если в таблице маршрутизации указан фрагмент имени, заканчивающийся символом «точка», то данная строка задает правило маршрутизации для всех имен, начинающихся с этого фрагмента. Например, если внести в таблицу маршрутизации выходного прокси-сервера в LAN_1 строку

tomsk. 195.88.210.21

то обращение клиентской программы по любому *URL*, начинающемуся со строки http://tomsk. (например, http://tomsk.persondb.ws или http://tomsk.docdb.ws и т.п.), будет «перенаправляться» на входной прокси-сервер по адресу 195.88.210.21. Данная возможность позволяет устранить необходимость корректировать таблицу маршрутизации выходного прокси-сервера при появлении каждого нового информационного ресурса в удаленных частных сетях

- Если бы почему-либо понадобилось переместить, например, веб-сервис **http://192.168.0.1/Person/Service.asmx** на другой узел в той же частной сети, то простой коррекции таблицы маршрутизации входного прокси-сервера было бы достаточно, чтобы никто из внешних клиентов не заметил этого изменения.

4. Структура прокси-сервера

И входной и выходной прокси-серверы представляет собой постоянно активные программы, реализованные в форме системного сервиса (в платформе *Win*32) [1] и в форме «демона» (в платформе *UNIX*) [5]. Оба способа реализации используют одну и ту же общую структуру прокси-сервера, изображенную на рис. 3 (и даже общий исходный код на языке *C*++).

Любая деятельность в прокси-сервере инициируется функцией «Приемник соединений», в которой реализован «вечный цикл» прослушивания входящих соединений на специальном

выделенном порте. Обнаружив запрос на входящее соединение, «Приемник соединений» порождает обрабатывающую программную нить для обслуживания этого соединения, организует двунаправленный *TCP*-канал, связывающий созданную программную нить с программой клиента, и возвращается к прослушиванию входящих соединений. Вся содержательная обработка выполняется в рамках программной нити. Эта обработка включает прием *HTTP*-запроса по сети, его анализ и извлечение *URL* адресуемого ресурса из заголовка запроса (функция «Обработчик первичных соединений») и обращение к таблице маршрутизации для определения направления дальнейшего продвижения запроса. Функция «Обработчик вторичных соединений» обеспечивает установление соединения со следующим сервером (для выходного прокси-сервера это может быть входной прокси-сервер удаленной частной сети, а для входного прокси-сервера – веб-сервер, поддерживающий адресуемый сетевой ресурс) с помощью функции «Коннектор».

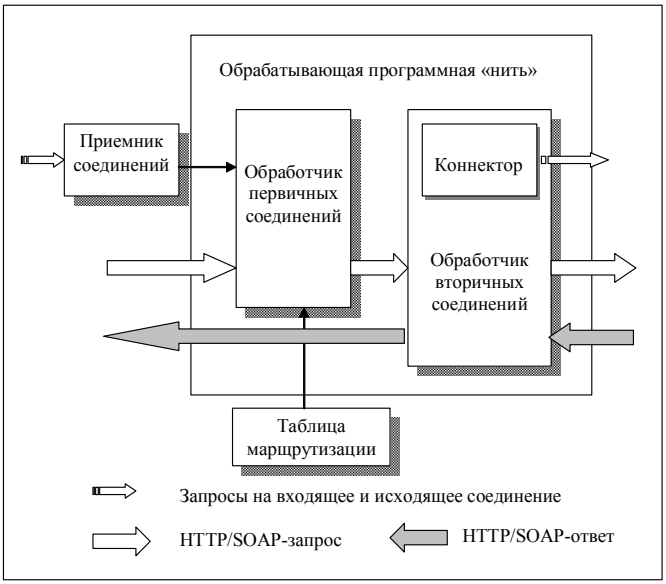

Рис. 3. Структура прокси-сервера

5. Взаимодействие между прокси-серверами

До сих пор мы намеренно не акцентировали внимание на вопросе о сетевом протоколе, используемом для взаимодействия между прокси-серверами. Вернее, предполагался простейший вариант: использование *HTTP* во всех «звеньях» межсетевого канала. Однако данный вариант не является ни единственно возможным, ни наиболее предпочтительным. Любой сетевой протокол, способный обеспечить передачу *HTTP*-запроса от одного сервера до другого и ответа в обратном направлении, может быть использован в этом звене без нарушения «прозрачности» канала для *HTTP*-клиента и *HTTP*-сервера.

Так как взаимодействие между прокси-серверами осуществляется через глобальную сеть, главным требованием к его организации является обеспечение защиты от несанкционированного доступа. С этой точки зрения данное взаимодействие может быть построено на базе целого ряда сетевых технологий.

- Протокол *HTTPS* представляет собой обычный протокол *HTTP*, организованный «поверх» защищенного *TCP*-соединения на базе технологии *SSL* (*Secure Socket Layer*). *HTTPS* оснащен необходимыми средствами авторизации и криптозащиты данных и представляет собой одно из наиболее эффективных решений для организации межсетевых каналов.
- Протоколы *PPP* и *IPSec* уже упоминались в данной статье в качестве базовых технологий организации виртуальных частных сетей. Оба протокола предполагают шифрование потока данных на уровне отдельных *IP*-датаграмм и строгую авторизацию клиента, открывающего сетевое соединение. В сущности, применение этих технологий в данном случае означает связывание всех выходных и входных прокси-серверов в одну виртуальную частную сеть.
- Протокол *RFPP* представляет собой сетевую технологию, специально ориентированную на поддержку удаленного взаимодействия в распределенных системах. Главными его особенностями являются устойчивость к сетевым сбоям, наличие собственных средств защиты от несанкционированного доступа

и межсерверной маршрутизации данных, а также возможность совмещения передачи данных с их обработкой на разных серверах. Эти свойства *RFPP* могут быть эффективно использованы при построении каналов межсетевого обмена данными [2, 3].

6. Заключение

Задача организации информационного взаимодействия в мульти-сетевой среде, включающей множество частных сетей предприятий, все чаще «выходит на первый план» в разработках распределенных систем. Несмотря на более чем двадцатилетнюю историю развития Интернета, следует признать, что решение этой задачи совершенно недостаточно поддержано современными сетевыми технологиями.

Если необходимо обеспечить защищенное взаимодействие между большим количеством (десятки и сотни) независимо администрируемых частных сетей, то требование уникальности *IP*-адресов информационных ресурсов может оказаться невыполнимым, а использование *VPN* – невозможным. В подобной ситуации разработчик обычно бывает вынужден тратить время и силы на создание собственных средств решения этой проблемы, приспособленных к особенностям конкретного проекта.

Рассмотренный в данной статье метод организации удаленных взаимодействий в мульти-сетевой среде не относится к числу международно-признанных интернет-технологий, прошедших соответствующую стандартизацию. К его недостаткам можно отнести жесткую ориентацию на протокол *HTTP*/*SOAP* и дополнительную задержку, вносимую серверами-посредниками (хотя эта задержка измеряется долями секунды, она может заметно снизить производительность системы с высокой интенсивностью межсетевых запросов). Тем не менее, он может рассматриваться, как одно из возможных «общих решений» вышеупомянутой задачи, избавляющих разработчиков от необходимости создания собственных средств организации взаимодействия в мульти-сетевой среде в каждом конкретном проекте.

Опыт реализации описываемого подхода (на платформах *Win*32 и *UNIX*/*Linux*) и его применения в разработках распределенных систем на базе технологии *.NET* показал его удобство для разработчика и достаточно высокую универсальность:

- подход может применяться в мульти-сетевых структурах самой различной размерности и сложности;
- однажды созданная система каналов межсетевого взаимодействия может быть использована различными автоматизированными системами, работающими в одной и той же сетевой среде.

Отметим, что межсетевой канал является абсолютно «прозрачным» и для *HTTP*-клиента и для *HTTP*-сервера (последние могут даже не подозревать о его существовании), а его работа совершенно не зависит от структуры и содержания передаваемых по нему информационных запросов (например, от набора веб-сервисов и спецификаций их методов).

Литература

1. АНДРЕЕВ А.Г., БЕЗЗУБОВ Е.Ю., ЕМЕЛЬЯНОВ М.М. и др. *Widows 2000: Server и Professional.* – СПБ.: «БХВ-Санкт-Петербург», 2001. – 1055 с.
2. АСРАТЯН Р.Э., ЛЕБЕДЕВ В.Н. *Интернет-служба обеспечения информационного взаимодействия в современных распределенных гетерогенных системах.* – М.: ЛЕНАНД, 2009. – 128 с.
3. АСРАТЯН Р.Э. *Межсерверная маршрутизация HTTP/SOAP-взаимодействий в распределенных системах* // Проблемы управления. – 2008. – №5. – С. 57–61.
4. ДЖАМСА К., КОУП К. *Программирование для Интернет в среде Windows.* – СПб.: Питер, 1996. – 659 с.
5. КЕЛЛИ-БУТЛ С. *Введение в Unix.* – М.: ЛОРИ, 1995. – 596 с.
6. МАК-ДОНАЛЬД М., ШПУШТА М. *Microsoft ASP.NET 3.5 с примерами на C# 2008 и Silverlight 2 для профессионалов.* — М.: Вильямс, 2009 – 1408 с.

7. СНЕЙДЕР Й. *Эффективное программирование TCP/IP. Библиотека программиста.* – СПб.: Символ-Плюс, 2002. – 320 с.
8. ХАНТ К. *TCP/IP. Сетевое администрирование.* – СПб.: Питер, 2007. – 816 с.
9. ШАПОШНИКОВ И.В. *Web-сервисы Microsoft .NET*. – СПб.: «БХВ–Петербург», 2002. – 336 с.

ORGANIZATION OF PROTECTED HTTP-INTERACTION IN MULTI-NETWORK ENVIRONMENT

Ruben Asratian, Institute of Control Sciences of RAS, Moscow, Cand.Sc., assistant professor (rea@ipu.ru).
Vitali Lebedev, Institute of Control Sciences of RAS, Moscow, Cand.Sc., assistant professor (lebvini@ipu.ru).

Abstract: The problem is considered of information interchange in .NET-based geographically distributed information systems designed for complex multi-network environments consisting of several private enterprise networks. We suggest using the protected inter-network channels based on a system of proxy servers. The servers route HTTP-interactions using names of network nodes and resources.

Keywords: distributed systems, Internet-based technologies, network protocols, web-services.

Статья представлена к публикации членом редакционной коллегии А. А. Печниковым