

## МОДЕРНИЗАЦИЯ АРХИТЕКТУРЫ IP-ТЕЛЕФОНИИ В ИНФРАСТРУКТУРЕ ВИРТУАЛЬНЫХ РАБОЧИХ СТАНЦИЙ

<sup>1</sup>Козиков В.Д., <sup>2</sup>Ляхов Ф.А., <sup>1</sup>Семашко А.В.

<sup>1</sup>ФГБОУ ВПО «Нижегородский государственный технический университет им. Р.Е. Алексеева», Нижний Новгород, Россия (603950, ГСП-41, Н.Новгород, ул. Минина, 24), e-mail: vkozikov@mera.ru

<sup>2</sup>ООО «МЕРА НН» Нижний Новгород, Россия (603163, Н.Новгород, ул. Деловая, 13), e-mail: lyakhov@mera.ru

**В статье рассматриваются вопросы функционирования приложений реального времени в инфраструктуре виртуальных рабочих станций. Проводится краткий обзор технологии виртуальных станций, а также устанавливаются основные проблемы таких систем, с которыми сталкиваются пользователи и разработчики решений универсальных коммуникаций и IP-телефонии. Сущность проблемы сводится к необходимости модернизации архитектуры маршрутизации данных с целью улучшения качества медиа в процессе связи и снижения нагрузки на сервера и виртуальную сеть в целом. Рассматриваются существующие методы решения данной проблемы, а также присутствующие на рынке решения от крупнейших производителей средств универсальных коммуникаций. Проводится их анализ и определяются основные недостатки. На основании этого предлагается создание универсального кроссплатформенного решения для решения рассматриваемой проблемы.**

Ключевые слова: виртуальные сети, инфраструктура виртуальных рабочих станций, IP-телефония

## MODERNIZATION OF IP-TELEPHONY ARCHITECTURE IN VIRTUAL DESKTOP INFRASTRUCTURE

<sup>1</sup>Kozikov V.D., <sup>2</sup>Lyakhov F.A., <sup>1</sup>Semashko A.V.

<sup>1</sup>FGBOU VPO Nizhny Novgorod State Technical University n.a. R.E. Alekseev, Nizhny Novgorod, Russia (603950, Nizhny Novgorod, street Minina, 24), e-mail: vkozikov@mera.ru

<sup>2</sup>LLC "MERA NN", Nizhny Novgorod, Russia (603163, N. Novgorod, street Delovaya, 13), e-mail: lyakhov@mera.ru

**The article considers the issues about the real-time applications functioning in virtual desktop infrastructure. It contains short review of this technology and describes the main problems which are faced by users and developers of the Unified Communications and IP-telephony solutions. The basic task boils down to the necessity of data routing modernization in order to improve the quality of media and reduce the load on the servers and virtual network. The article provides the analysis of present methods and solutions from the vendors of unified communications and determines their main disadvantages. The universal cross-platform solution is proposed to solve the considered issue.**

Keywords: virtual networks, virtual desktop infrastructure, IP-telephony

Администрирование рабочих мест для IT-персонала всегда было нетривиальной и требующей времени задачей. А в последнее время с увеличением рисков безопасности данных и количеством так называемых «мобильных пользователей» – тех, кто пользуется ресурсами компании с мобильных устройств – это стало еще более сложной проблемой. В типичной ситуации, если возникает какая-либо неполадка, IT-работник должен физически добраться до проблемного компьютера. Не стоит также забывать о необходимости регулярного обновления программного обеспечения, которое не всегда может быть автоматизировано, а для выполнения не всегда хватает знаний ПК обычным пользователем. Таким образом, стараясь найти способы уменьшения затрат на улучшение и содержание информационных систем и их безопасности, компании стали все чаще обращаться к технологиям виртуализации. Началось активное совершенствование данного подхода и более

глубокая интеграция в информационные системы, в частности – в виртуализацию рабочих станций. Инфраструктура, которая была реализована с помощью данного подхода, получила название Virtual Desktop Infrastructure (Инфраструктура виртуальных рабочих станций, VDI).

Главная идея технологии заключается в том, что операционная система пользователя со всеми его приложениями и данными находится не на физической машине, которая стоит у него на рабочем месте, а на виртуальной, размещенной на сервере в центре данных компании [6]. Работа с виртуальной машиной осуществляется через так называемый виртуальный рабочий стол. Для подключения к нему можно воспользоваться обычным ПК, но гораздо более рационально с точки зрения экономии ресурсов использовать специальные устройства – тонкие клиенты. Тонкий клиент представляет собой системный блок, со значительно меньшей по сравнению с традиционным ПК производительностью и вычислительной мощностью. К нему так же подключаются монитор, клавиатура, мышь и другие периферийные устройства, но благодаря нетрадиционной архитектуре и некоторым особенностям – например отсутствию жесткого диска – это устройство обладает значительно меньшими размерами и, главное, стоимостью. Клиент подключается к виртуальному рабочему столу через определенный протокол и начинает работать с виртуальной машиной. В результате чего пользователь имеет доступ ко всем своим данным и приложениям. Но со стороны это выглядит так, будто все хранится и обрабатывается на локальной машине.

Таким образом, можно выделить 3 основных преимущества технологии VDI:

- Снижение издержек на поддержку и обновление рабочих мест.
- Повышение уровня безопасности данных и отказоустойчивости системы.
- Доступ к рабочему месту из любой точки (из офиса, дома, в поездке), с любого устройства.

Но, как и любая развивающаяся система, VDI не лишена недостатков, существенным из которых является низкое качество работы мультимедиа-приложений реального времени, например, аудио- и видеотелефонии. Чтобы понять основную причину данной проблемы, рассмотрим традиционную архитектуру IP-телефонии в системах, где используются физические ПК во время звонка. При установлении соединения при аудио- и/или видеосессии с удаленным клиентом VoIP-приложение должно получить его IP-адрес от сервера сигнализации. С помощью аудио и видео устройств осуществляется запись звука и изображения с последующим их кодированием [1]. Получаемые в результате закодированные данные в виде пакетов отправляются на IP-адрес удаленного клиента. Одновременно с этим, VoIP-приложение считывает пакеты с порта, указанного в параметрах сессии, которые приходят от удаленного клиента. Эти данные декодируются в медиа-поток и отправляются на устройства вывода. Архитектура может усложняться в зависимости от типа используемого

продукта, наличия прокси-серверов в сети или файервола, но базовый принцип остается прежним [2].

Далее рассмотрим ситуацию на примере клиента IP-телефонии, установленного на виртуальной машине в архитектуре VDI. Во время звонка входящий медиа-поток проходит через сервер виртуализации, затем попадает в виртуальную машину и декодируется VoIP-приложением. Получаемое в результате аудио проигрывается в виртуальном аудио устройстве, видео отрисовывается на виртуальной видеокarte. Затем все эти данные снова кодируются и пересылается по каналу между виртуальной машиной и VDI клиентом, в очередной раз проходя через сервер виртуализации. При этом данный канал часто не поддерживает разделение информации отдельно на аудио и видео данные. В результате чего такая архитектура не поддерживает необходимый уровень качества обслуживания (Quality of Service, QoS). Такая схема обработки (называемая «hairpinning») приводит к низкому качеству потока медиа, приходящего к пользователю. Такой поток характеризуется большой задержкой и джиттером («jitter») – колебанием или дрожанием задержки при передаче пакетов, а также потерей качества звука и изображения при перекодировании. Кроме того, повышается нагрузка на сервер виртуализации и на локальную сеть.

С увеличением числа компаний, которые стали применять продукты, основанные на технологиях VDI, указанная проблема начала приобретать все большую актуальность. Поэтому улучшением качества работы медиа-приложений в условиях виртуальных рабочих станций занимаются как разработчики виртуальных инфраструктур (Citrix, VMWare, Microsoft, Red Hat), так и разработчики средств VoIP-телефонии (Microsoft, Avaya, Cisco) [5]. Рассмотрим самые популярные решения.

- Avaya VDI Communicator (VDI-C). Данный продукт компании Avaya является не просто решением для реализации архитектуры VoIP в виртуальной среде, а полноценным приложением IP-телефонии, основанным на протоколе SIP и развернутом на тонком клиенте. Такой подход дает определенные преимущества, включая возможность совершать звонки даже в случае потери связи с сервером виртуализации, но реализация при этом не лишена недостатков:

- Жесткая привязанность к UC-инфраструктуре от Avaya.
- Ядро VDI-C основано на проприетарных технологиях Avaya Spark Engine и Avaya Shared Control, что сводит на нет уровень универсальности.
- Технические сложности при попытке портирования VDI-C на другие протоколы по причине тесной интеграции с SIP.

Таким образом, решение проблемы, рассматриваемой в данной статье, присутствует лишь частично, а возможности применения сильно ограничены производителем.

- Cisco Virtualization Experience Media Engine (VXME). Данное решение технологически во многом совпадает с Avaya VDI Communicator, т. е. на стороне клиента имеется полноценное VoIP-приложение. Соответственно, эта технология обладает схожими преимуществами и недостатками. Стоит отметить, что в отличие от Avaya Shared Control, VXME использует технологию SIP CTI, которая является гораздо более универсальным решением. Но факт, что продукт Cisco работает исключительно с собственными UC-клиентами, также сводит на нет универсальность решения.

- Microsoft Lync for VDI. Как и большинство решений от компании Microsoft, данный продукт работает только с системами на операционной системе Windows. Кроме того, он имеет высокие системные требования, которыми обладают только топовые тонкие клиенты. Функционирование изначально было основано только на протоколах RDP 7 и RemoteFX, однако уже заявлена поддержка VDI-технологий Citrix Receiver и VMWare Horizon View. Реализация не совсем очевидна, информации в свободном доступе не так много, что характерно для продуктов от Microsoft. Судя по информации из пресс-релизов компании, можно предположить, что на тонком клиенте аналогичным образом используется VoIP-приложение, использующее протокол SIP. Таким образом, решение подобно реализациям Avaya VDI-C и Cisco VXME. Следовательно, учитывая при этом особенности политики Microsoft, оно обладает теми же недостатками.

- Citrix HDX RealTime optimization pack for Microsoft Lync. Этот продукт создан на основе технологии Avistar C3 Unified for MS Lync 2010. Судя по описанию, он представляет собой плагин для коммуникатора Lync, похожий на Avaya Communicator for Lync. Как и в случае с Microsoft Lync for VDI, очень мало информации находится в свободном доступе. Судя по существующей документации можно предположить, что реализация этого продукта основана на обработке сигналинга VoIP-приложением на виртуальной машине, а за обработку потоков медиа отвечает медиа-ядро, расположенное на тонком клиенте. Этот подход схож с концепцией предлагаемого решения, представленного в данной статье. Однако документального подтверждения этому найти не удалось. Так или иначе, существенным недостатком снова является закрытость продукта и жесткая привязка к определенному поставщику UC-решений (Avistar или Microsoft) и поставщику VDI-решений (Citrix).

Таким образом, в результате проведенного анализа было установлено, что существующие на сегодняшний день решения местами сильно ограничены с точки зрения универсальности из-за условий, создаваемых конкуренцией между компаниями-производителями, а также с технической стороны вопроса. На основании этой информации можно сделать вывод об отсутствии решения описанной проблемы на глобальном уровне.

Для решения проблемы предлагается создать систему управления виртуального медиа

(Virtual Media Controller, VMC), основной задачей которой является исключение процесса прохождения потока медиа через виртуальную машину и сервер виртуализации. Отличительной особенностью VMC является универсальность. Благодаря использованию кроссплатформенных технологий и открытых стандартов будет исключена привязанность к определенным поставщикам UC-систем или инфраструктурам VDI. Система состоит из четырех взаимосвязанных компонентов: Agent, Engine, Transport и Overlay Renderer.

– Agent. Сервис, работающий на виртуальной машине и предоставляющий набор интерфейсов управления потоками медиа для клиентского приложения (например, для используемого на машине VoIP-коммуникатора). Традиционное средство IP-телефонии имеет несколько модулей, одним из которых является медиа-ядро, отвечающее за обработку звука, его воспроизведение и запись, а также передачу и отправку по сети. Этот элемент приложения может быть полностью интегрирован в него, или представлен в виде библиотеки или отдельного процесса в системе. Взаимодействие остальных модулей с ядром осуществляется разными способами, одним из которых является использование интерфейсов программирования приложений API (Application Programming Interface) [4]. При таком подходе приложение (библиотека, сервис) предоставляет набор своих готовых процедур, классов или функций для использования другими программами. Сильная сторона использования API заключается в том, что он определяет функциональность, которую предоставляет программный компонент, позволяя при этом абстрагироваться от того, как именно эта функциональность реализована. В системе VMC средство IP-телефонии включает в себя Agent в качестве медиа-ядра и взаимодействует с ним средствами предоставляемых интерфейсов. Но так как Agent расположен на виртуальной машине, а главной задачей является перенос обработки данных на тонкого клиента, Agent в свою очередь делегирует эту задачу следующему элементу.

– Engine. Медиа-ядро, развернутое на VDI-клиенте пользователя. На нем происходит вся обработка принимаемых и отправляемых данных медиа средствами самого тонкого клиента. В традиционных коммуникаторах это ядро либо вшито в само приложение, либо запускается отдельным процессом, но находится на одной машине с другими компонентами. Основными требованиями к реализации является оптимизация под работу на VDI-клиентах и использование открытых стандартов VoIP. Для этих целей подходящим вариантом является технология Google WebRTC Media Engine. Она не слишком требовательна к ресурсам, что положительным образом сказывается при использовании в VDI-инфраструктуре, а также поддерживает все самые распространенные операционные системы – Linux, Windows, OS X, iOS, Android. Кроме того важно отметить, что WebRTC распространяется по лицензии BSD-3 и является проектом с открытым исходным кодом.

– Transport. Компонент, обеспечивающий связь между Engine и Agent. Данный элемент необходим для организации передачи данных через каналы, предоставляемые сторонними системами доступа к удаленным виртуальным машинам. Необходимым требованием является возможность использования в каналах передачи наиболее распространенных VDI-решений – VMWare Horizon View, Citrix HDX, Microsoft RemoteFX и RedHat Spice. Предполагается, что взаимодействие Agent и Engine будет реализовано на основе технологий вызова удаленных процедур (Remote Procedure Call, RPC). RPC представляет собой класс технологий, позволяющих приложениям вызывать методы или функции в других адресных пространствах [3]. В случае с VMC необходимо с расположенного на виртуальной машине Agent получить доступ к методам Engine, находящегося на тонком клиенте пользователя. Одной из возможных реализаций может быть использование технологии Thrift от компании Apache Software Foundation. Она представляет собой язык описания интерфейсов (Interface Description Language, IDL) и является фреймворком под технологию RPC. Thrift уже включается в себя готовый программный конвейер для работы с клиентской и серверной частью, поддерживает большинство популярных языков программирования и имеет открытый исходный код. Другим вариантом может быть система межпроцессного взаимодействия D-Bus, часть проекта freedesktop.org. Вне зависимости от используемой технологии главной целью является реализация методов RPC между компонентами Engine и Agent.

– Overlay Renderer. Компонент, отвечающий за интеграцию отображения видеоданных, получаемых от Engine, в сторонний клиент доступа к удаленной виртуальной машине. Этот элемент имеет следующую концепцию работы: на виртуальной машине приложение (например, IP-коммуникатор) создает пустое окно, которое будет предназначаться для отображения видео. Его координаты и размеры передаются в Agent и затем сообщаются Engine, который выполняет отрисовку видео с заданной геометрией и помещает его внутри окна клиента доступа к виртуальной машине под контролем Overlay Renderer. В результате для пользователя создается ощущение, что видео отображается в окне IP-коммуникатора.

Система, реализованная с помощью указанных компонентов, работает по следующему принципу: средство универсальной коммуникации (например, IP-коммуникатор) находится на виртуальной машине. Оно использует Agent в качестве своего медиа-ядра. Взаимодействие и управление осуществляется средствами стандартизированных API, предоставляемых Agent. Таким образом, вся обработка медиа-данных делегируется VoIP-приложением компоненту Agent, который в свою очередь с помощью технологий RPC выполняет эту задачу средствами Engine. Взаимодействие между двумя элементами

обеспечивает Transport. Компонент Overlay Renderer становится задействованным при инициализации видеозвонка и реализует отображение видео в предназначенное для этого IP-коммуникатором окно. В результате, поток медиа, принимаемый от удаленного пользователя, избегает прохождения через виртуальную машину или сервер виртуализации. Таким образом, решается проблема «hairpinning», и снижается нагрузка на виртуальную сеть. Отличие от описанных в данной статье существующих решений заключается в том, что VoIP-приложение становится независимым от используемой VDI-инфраструктуры. Все проблемы интеграции берет на себя VMC. При таком подходе большие преимущества получают разработчики средств IP-телефонии и универсальных коммуникаций. Кроме модернизации архитектуры в сторону улучшения качества, с технической стороны в приложениях пропадает необходимость в подстраивании под разные реализации VDI.

Основные преимущества:

- VMC является решением, не зависящим от используемых в системе протоколов. Существующие реализации для систем универсальных коммуникаций в большинстве случаев включают в себя базовое ядро, работающее на тонком клиенте, для обработки сигналинга. Это необходимо по причине использования основного протокола (например, SIP в случае с Avaya VDI-C) в качестве транспортного, т.е. для реализации взаимодействия между компонентами на виртуальной машине и на физическом клиенте. Такой подход создает дополнительные ограничения, когда требуется поддержка не одного, а нескольких протоколов. В предлагаемом решении данная проблема исключена.

- VMC сфокусирован в первую очередь на интеграцию с инфраструктурами виртуальных рабочих станций. Модернизация с применением этого решения относительно проста, в тот момент как попытка модернизировать существующие решения от крупнейших UC-поставщиков потребует серьезных затрат.

- VMC основан на универсальных и распространенных технологиях. Например, Google WebRTC Media Engine, пользующаяся большой популярностью и хорошо зарекомендовавшая себя среди разработчиков. В случае, когда в UC-системе уже используется этот продукт (Wildix PBX, Oracle Communications Unified Communications Suite и т.д.), процесс модернизации средствами VMC будет еще менее затратным. При переходе с другого медиа-ядра трудности также будут минимальными, благодаря открытому исходному коду используемых компонентов.

Практически все существующие на сегодняшний день решения направлены на улучшение качества медиа при работе в инфраструктуре виртуальных рабочих станций. В основе всех продуктов лежит один и тот же принцип – исключение процесса «hairpinning», но реализация в каждом случае отличается. Практически во всех решениях использован

подход, основанный на закрытых технологиях вместо открытых стандартов. Из-за этого при разработке UC-приложений для систем VDI возникает необходимость тщательно рассматривать особенности каждой системы и подстраивать под них, что создает заметные неудобства. Предлагаемая система Virtual Media Controller – это независимое и обобщенное решение рассматриваемой задачи, потенциальная реализация которого проще большинства существующих решений, и при этом может иметь более широкое применение благодаря своей универсальности.

### Список литературы

1. Гольдштейн, В.С. IP-Телефония / В.С.Гольдштейн, А.В. Пинчук, А.Л. Суховицкий – М.: Радио и связь, 2001. – 336 с.
2. Кузьмин, В.В. Методы управления трафиком в кампусных сетях оператора связи / В.В. Кузьмин, А.В. Семашко // Электросвязь. – 2012. - № 5.
3. Олифер, В. Г. Компьютерные сети. Принципы, технологии, протоколы: учебное пособие для вузов / В. Г. Олифер, Н. А. Олифер. – СПб. [и др.], 2007. – 957 с.
4. Рихтер, Дж. Windows для профессионалов: создание эффективных Win32 приложений с учетом специфики 64-разрядной версии Windows / Пер. с англ. — 4е изд. — СПб.: Питер; М.: Издательство «Русская Редакция»; 2008. — 720 с.
5. Vanover, R. VDI Display Protocols – Why They Are Important [Электронный ресурс]. – Режим доступа: <http://virtualizationreview.com/articles/2010/05/01/vdi-display-protocols.aspx> (дата обращения: 17.04.14).
6. Virtual Desktop Infrastructure Documentation [Электронный ресурс]. – Режим доступа: [http://www.vmware.com/support/pubs/vdi\\_pubs.html](http://www.vmware.com/support/pubs/vdi_pubs.html) (дата обращения: 29.11.14).

### Рецензенты:

Хранилов В.П., д.т.н., профессор, зам. директора по НИР и магистратуре НГТУ им. Р.Е. Алексеева, г. Нижний Новгород;

Соколова Э.С., д.т.н., профессор, зав. кафедрой «Информатика и системы управления» НГТУ им. Р.Е. Алексеева, г. Нижний Новгород.