

Методы и особенности взаимодействия клиентов информационной системы с центром управления в целях мониторинга инфраструктуры предприятия

Е.А. Верещагина, А.К. Рудниченко, Д.С. Колесникова

Дальневосточный федеральный университет, Владивосток

Аннотация: В данной работе приведено описание методов взаимодействия клиентов информационной системы с центром управления. Определён протокол для данного взаимодействия, а также описаны его преимущества и недостатки. Произведены вычисления времени передачи полного пакета информации в рамках задачи мониторинга рабочих станций в организации, а также сделаны выводы на основе результатов вычислений.

Ключевые слова: Active Directory, клиент, сервер, мониторинг, TCP, протокол, особенности, скорость, RTT.

В настоящее время, когда широко распространено применение информационных технологий во всех сферах жизни человека, довольно трудно представить крупную организацию с парком компьютерной техники, состоящим менее, чем из 100 единиц [1]. Администрирование вручную в таком случае становится практически невозможным, вследствие чего системные администраторы прибегают к централизованному управлению.

Самым распространённым видом централизованного управления можно считать службы каталогов Microsoft Active Directory [2]. Active Directory характеризуется наличием центрального сервера (контроллер домена), к которому обращаются все клиентские машины. Использование данной службы делает возможным применение настроек и установку программного обеспечения на конечных рабочих станциях.

Стандартными средствами Active Directory невозможно произвести аудит и/или мониторинг рабочих станций. В частности, проблемы возникают в процессе получения имени учетной записи лица, осуществляющего работу за данной рабочей станцией, характеристик компьютера и т. д. В таких случаях необходимо использовать пользовательские скрипты, написанные на

языке сценариев Microsoft Visual Basic Script Edition (VBS) [3]. При этом получение информации о, например, температуре процессора, в данной ситуации и вовсе невозможно. В связи с этим на данный момент осуществляется разработка информационной системы для мониторинга и управления инфраструктурой предприятия.

Для того, чтобы осуществлять централизованное управление рабочими станциями в пределах одной локальной вычислительной сети необходимо, чтобы рабочая станция пользователя получала некоторый набор данных от центра управления. Передача команд, как правило, осуществляется согласно принципам клиент-серверной архитектуры.

Клиент-серверная архитектура – вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами. Относительно данной задачи клиент-серверная архитектура будет состоять в следующем [4]:

- Сервер будет получать информацию о рабочих станциях (их нагрузке, температуре процессора и т. д.), а затем аккумулировать полученные данные в базах данных. Кроме этого, он будет посылать набор управляющих команд для конечных рабочих станций.
- Клиенты в данном случае подразделяются на два вида. Первые – рабочие станции пользователей. Агенты, установленные на рабочих станциях, будут собирать информацию о компьютере и отправлять на сервер. Вторые – это системные администраторы организации. Они будут получать данные на основе информации в базах данных сервера. На стороне клиента будут строиться графики и таблицы.

Существует два основных метода взаимодействия клиента и сервера относительно задачи мониторинга и управления инфраструктурой предприятия в одной локальной вычислительной сети:

1. Клиент (рабочая станция для мониторинга) отправляет пакет необходимых для мониторинга данных серверу с интервалом, ранее установленным в настройках.

2. Сервер опрашивает клиента по заданному интервалу, а клиент в ответ направляет пакет необходимых данных для мониторинга.

Стоит отметить, что второй метод взаимодействия возможно применять только тогда, когда клиент и сервер находятся в одной локальной вычислительной сети. Первый же метод возможно развернуть на разные сети, для этого необходимо будет назначить статический IP-адрес серверу и опубликовать его в сети Интернет (дать доступ по определённому порту извне).

Несмотря на то, что для разных сетей первый метод предпочтительней, для одной локальной вычислительной сети всё же более предпочтителен второй метод. Это связано с тем, что если сервер слушает порт и принимает пакеты, то в одну единицу времени по порту может прийти несколько пакетов. В таком случае создаётся очередь в буфере. В связи с этим актуален вопрос безопасности сервера, так как при предоставлении такой возможности «как есть» злоумышленники могут произвести атаку с переполнением буфера и вывести сервер из строя программным методом.

В случаях, когда сервер сам опрашивает клиента по заданному интервалу (второй метод), он может сам регулировать нагрузку на аппаратную часть.

Чтобы исключить недостаток первого метода, необходимо выстроить алгоритм следующим образом:

- Клиент отправляет серверу данные (контрольный пакет) о том, что готовится передать информацию о рабочей станции для мониторинга.

- Сервер оценивает нагрузку и одобряет (или не одобряет) дальнейшую передачу данных. При одобрении сервер отправляет в поток информацию о том, что можно передавать данные.
- Клиент отправляет данные для мониторинга, сервер их принимает и анализирует.

В случае, если серверу необходимо передать управляющую команду клиенту (например, установка программного обеспечения или выполнение скрипта в командной строке), он будет ожидать запроса клиента на передачу данных.

Предполагается, что взаимодействие между клиентом и сервером будет происходить по TCP протоколу. TCP протокол имеет ряд недостатков [5].

TCP-сервер на всё время ожидания подключений открывает 1 серверный (LISTENING) сокет, привязывая его к конкретному номеру порта. В связи с этим, в один момент времени сервер может получить данные только от одного клиента, вследствие чего TCP-сервер должен уметь управлять очередью данных (буфером), так как чтение данных и их обработка занимает некоторое время.

Второй недостаток является следствием из первого. Так как прием данных происходит по порту в одном потоке, с ростом количества управляемых рабочих станций необходимо увеличивать и производительность сервера [6].

Несмотря на вышеуказанные ограничения, у данного протокола есть и преимущества. Одно из главных преимуществ состоит в том, что TCP-протокол имеет функционал отправки «контрольного пакета», вследствие чего разработчику не нужно самостоятельно узнавать доступность данных [7].

Второе преимущество – контроль потока TCP-протокола предотвращает отправку слишком большого количества данных получателю,

которые он не сможет обработать [7]. Чтобы этого не происходило, каждая сторона TCP-соединения сообщает размер доступного места в буфере для поступающих данных. Таким образом, если объём пакета превышает допустимый предел, установленный разработчиком, он не берётся во внимание.

На рисунке 1 показана диаграмма деятельности процесса передачи данных для мониторинга рабочей станции с последующим получением и выполнением определённой команды. На данном примере применяется AES шифрование при передаче данных по открытому каналу связи.



Рис. 1. – Диаграмма деятельности взаимодействия сервера и клиента

Стоит отметить, что несколько TCP-соединений и одно соединение TCP, равные по объёму, даже по локальной вычислительной сети передаются по-разному. Например, 8 соединений по 256 байт (всего 2048 байт) и одно соединение с 2048 байт не будут иметь одинаковую скорость передачи и последующего чтения [8].

Максимальная скорость отдельного TCP соединения определяется различными факторами [9]:

- максимальная пропускная способность самого медленного участка пути;
- время между отправкой запроса и получением ответа;
- нагрузка на локальную вычислительную сеть и сервер;
- автоматическое понижение скорости при обнаружении потерь пакетов (стандартный механизм TCP-протокола по предотвращению перегрузок).

В рамках эксперимента возможно рассчитать, за какое время клиент передаст, а сервер прочитает данные размером в 367 КБ (376 351 байт). Указанный размер файла взят из расчета того, что в рамках регистрации рабочей станции, а также минимум 2 раза в день, клиент отправляет большой пакет данных, который включает в себя перечень установленного программного обеспечения, сервисы и запущенные процессы.

Учитывая работу протокола TCP, можно рассчитать максимальную скорость передачи данных в один поток:

$$V_{max} = \frac{W}{RTT},$$

где W – размер окна (буфера) в байтах, RTT – время в секундах, которое проходит с момента отправки сообщения до момента получения ответа.

Из данной формулы можно вывести значение RTT :

$$RTT = \frac{W}{V}$$

В локальной вычислительной сети данные передаются на скорости 100 Мбит/с. В размер окна (буфера) установим значение равное 1024 байт. После перевода величин и расчёта получается следующий результат:

$$RTT = \frac{W}{V} = \frac{1\,024 * 2^3}{100 * 2^{20}} = \frac{8\,192}{104\,857\,600} = 0,000078125 \text{ сек} \approx 0,0781 \text{ мс}$$

Из расчётов видно, что для передачи 1024 байт информации в идеальных условиях необходимо затратить 0,0781 миллисекунд реального времени. Следовательно, для передачи 376 351 байт информации необходимо будет затратить:

$$RTT_{all} = \frac{0,078125}{1\ 024} * 376\ 351 \approx 28,71\ \text{мс} \approx 0,02871\ \text{сек}$$

Данный расчёт верен для идеальных условий без учёта различных оптимизаций данных и процесса передачи информации по открытым каналам связи. Полученное значение времени возможно уменьшить в несколько раз, прибегнув к сжатию трафика. При помощи оптимизации возможно достичь более лучших результатов [10].

Стоит отметить, что при повторной передаче информации по протоколу TCP время сокращается. Принято считать, что время повторной передачи равно следующему [11]:

$$RTT = 90\% * RTT_{previous} + 10\% * RTT_{current} ,$$

Где $RTT_{previous}$ – предыдущее значение RTT , а $RTT_{current}$ – текущее значение RTT .

Передача данных по TCP-протоколу не является трудозатратным процессом для операционной системы и рабочей станции в целом. Учитывая, что процесс передачи по TCP-протоколу происходит с помощью отдельного фонового потока в процессе, временем передачи и вовсе можно пренебречь в рамках мониторинга рабочих станций.

Таким образом, определен протокол, а также метод взаимодействия клиентов информационной системы с центром управления в целях мониторинга инфраструктуры предприятия. Описан процесс взаимодействия между клиентом и сервером в рамках поставленной задачи.

Литература

1. Информационное общество // Федеральная служба государственной статистики. URL: gks.ru/folder/14478 (дата обращения: 09.05.2020).
 2. Централизованное управление локальной сетью в реализации Microsoft // ITCua. URL: itc.ua/articles/centralizovannoe_upravlenie_lokalnoj_setyu_v_realizacii_microsoft_3357/ (дата обращения: 10.05.2020).
 3. Scripts to manage active directory users // ActiveXperts Software. URL: activexperts.com/admin/vbscript-collection/usersgroups/users/ (дата обращения: 12.05.2020).
 4. Harrington Jan L. Client-Server Architecture // ScienceDirect. URL: sciencedirect.com/topics/computer-science/client-server-architecture (дата обращения: 09.05.2020).
 5. Внутренние механизмы TCP, влияющие на скорость загрузки: часть 2 // Habr. URL: habr.com/ru/company/weboblog/blog/327050/ (дата обращения: 12.05.2020).
 6. Акользин Д.Н. Обзор методов повышения производительности программного обеспечения диспетчерского центра // Инженерный вестник Дона, 2014, №2. URL: ivdon.ru/ru/magazine/archive/n2y2014/2394.
 7. Внутренние механизмы TCP, влияющие на скорость загрузки: часть 1 // Habr. URL: habr.com/ru/company/weboblog/blog/326258/ (дата обращения: 12.05.2020).
 8. The Cable Guy: TCP Receive Window Auto-Tuning // Microsoft Docs. URL: [docs.microsoft.com/en-us/previous-versions/technet-magazine/cc162519\(v=msdn.10\)?redirectedfrom=MSDN](http://docs.microsoft.com/en-us/previous-versions/technet-magazine/cc162519(v=msdn.10)?redirectedfrom=MSDN) (дата обращения: 12.05.2020).
-

9. Как получить и измерить высокоскоростное соединение по TCP // Habr. URL: habr.com/ru/post/115527/ (дата обращения: 11.05.2020).

10. Сироткин А.В., Брачун Т.А., Бархатов Н.И. Моделирование приоритетного управления информационными потоками с использованием сокетов // Инженерный вестник Дона, 2012, №4. URL: ivdon.ru/ru/magazine/archive/n4p1y2012/1192.

11. Транспортный уровень. Протокол управления передачей (Transmission Control Protocol — TCP) // НОУ ИНТУИТ. URL: www.intuit.ru/studies/curriculums/21342/courses/2/lecture/46?page=5 (дата обращения: 12.05.2020).

References

1. Informationsionnoe obshchestvo [Information society]. Federal'naya sluzhba gosudarstvennoy statistiki. URL: gks.ru/folder/14478 (accessed 05/09/2020)

2. Tsentralizovannoe upravlenie lokal'noy set'yu v realizatsii Microsoft [Centralized LAN management in Microsoft implementation]. ITCua. URL: itc.ua/articles/centralizovannoe_upravlenie_lokalnoj_setyu_v_realizacii_microsoft_3357/ (accessed 05/10/2020).

3. Scripts to manage active directory users. ActiveXperts Software. URL: activexperts.com/admin/vbscript-collection/usersgroups/users/ (accessed 05/12/2020).

4. Harrington Jan L. Client-Server Architecture. ScienceDirect. URL: sciencedirect.com/topics/computer-science/client-server-architecture (accessed 05/09/2020).

5. Vnutrennie mekhanizmy TSR, vliyayushchie na skorost' zagruzki: chast' 2 [Internal TCP mechanisms that affect download speed: Part 2]. URL: habr.com/ru/company/webo/blog/327050/ (accessed 05/12/2020).



6. Akol'zin D.N. Inzenernyj vestnik Dona 2014, №2. URL: ivdon.ru/ru/magazine/archive/n2y2014/2394.
7. Vnutrennie mekhanizmy TSR, vliyayushchie na skorost' zagruzki: chast' 1 [Internal TCP mechanisms that affect download speed: Part 1]. URL: habr.com/ru/company/webo/blog/326258/ (accessed 05/12/2020).
8. The Cable Guy: TCP Receive Window Auto-Tuning. Microsoft Docs. URL: [docs.microsoft.com/en-us/previous-versions/technet-magazine/cc162519\(v=msdn.10\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/technet-magazine/cc162519(v=msdn.10)?redirectedfrom=MSDN) (accessed 05/12/2020).
9. Kak poluchit' i izmerit' vysokoskorostnoe soedinenie po TCP [How to get and measure high speed TCP connection]. URL: habr.com/ru/post/115527/ (accessed 05/11/2020).
10. Sirotkin A.V., Brachun T.A., Barkhatov N.I. Inzenernyj vestnik Dona, 2012, №4. URL: ivdon.ru/ru/magazine/archive/n4p1y2012/1192.
11. Transportnyy uroven'. Protokol upravleniya peredachey (Transmission Control Protocol — TCP) [Transport level. Transmission Control Protocol (TCP)].
NOU INTUIT. URL: intuit.ru/studies/curriculums/21342/courses/2/lecture/46?page=5 (accessed 05/12/2020).