УДК 004.057.4

ПРОТОКОЛЫ ПЕРЕДАЧИ ДАННЫХ В СЕТИ

Алешкевич П.А., Лихацевич А.В., Пташник Д.А.

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники» филиал Минский радиотехнический колледж, г. Минск, Республика Беларусь

Научный руководитель: Буянова С.Г. – преподаватель высшей категории дисциплин специального цикла

Аннотация. *Для передачи данных между устройствами, действующих в разных сетях, необходимо наличие протоколов и стандартов. Данные протоколы и стандарты соблюдают правила технической организации компьютерных сетей, которые позволяют проводить взаимодействие между собой в сети.*

Ключевые слова: *протокол, сеть, пакет, данные, информация, модель OSI, Internet.*

Введение. Из-за стремительного развития технологий в конце XX века, были созданы новые, скоростные виды коммуникации между людьми. Сначала сети применялись только в качестве объектов исследований и экспериментов, но с ходом времени они стали интегрироваться в повседневную жизнь человека.

В то же время сети были строго типизированы и конкретизированы для выполнения отдельных задач, не взаимодействуя друг с другом. Сформировать удобную и эффективную физическую сеть для глобального покрытия из технологий одного типа не представляется возможным, потому что она не сможет удовлетворить потребности всех её пользователей. Одной группе пользователей необходима скоростная сеть для связи между континентами, а другим надёжное соединение аппаратуры в одну систему в одном задании.

В итоге было принято решение соединить физические сети в одну мировую сеть, в которой используются и связи на физическом уровне и инновационные стандарты и протоколы. Эта технология, получившая в результате название Internet, позволила системам обмениваться данными, несмотря на различия в подсоединении.

Основная часть.

IP – Internet Protocol.

IP протокол является самым первым протоколом передачи данных. Протокол IP собирает отдельные сегменты в единую сеть, это позволяет передавать пакеты данных между любыми узлами сети через произвольное число промежуточных узлов, например, маршрутизатором. Протокол IP является протоколом сетевого уровня по модели OSI.

При использовании протокола IP отсутствует гарантия надёжной доставки пакетов от отправителя к получателю – в частности, пакеты могут прийти не в том порядке, в котором были отправлены или продублироваться (приходят две копии одного пакета), оказаться повреждёнными (обычно повреждённые пакеты уничтожаются) или не прийти вовсе. Гарантию безошибочной доставки пакетов дают некоторые протоколы более высокого уровня – транспортного уровня сетевой модели OSI – например, TCP, которые используют IP в качестве транспорта. По IP-протоколу передача данных происходит без установки соединения.

Главной задачей IP является маршрутизация датаграмм, то есть речь идет об определении пути следования данных по узлам сети. До сегодняшнего дня наиболее распространённой версией являлся IPv4 с 32-битными адресами. Но, как известно, 4.29 млрд IPv4-адресов – это много, но уже давно недостаточно. Поэтому существует IPv6, который призван решить проблему переполнения адресов.

TCP/IP – Transmission Control Protocol/Internet Protocol.

Название TCP/IP происходит из двух важнейших протоколов семейства – Transmission Control Protocol (TCP) и Internet Protocol (IP), которые были первыми разработаны и описаны в данном стандарте. Также изредка упоминается как модель DOD (Department of Defense) в связи с историческим происхождением от сети ARPANET из 1970-х годов (под управлением DARPA, Министерства обороны США.

Протокол TCP/IP является сетевой моделью передачи данных по модели OSI. Модель описывает способ передачи данных от источника информации к получателю. В модели предполагается прохождение информации через четыре уровня, каждый из которых описывается определённым правилом, называемым протоколом передачи. Наборы правил, решающих задачу по передаче данных, составляют стек протоколов передачи данных, на которых базируется Интернет [1, 2].

Стек протоколов TCP/IP включает в себя четыре уровня:

– прикладной уровень;

– транспортный уровень;

– сетевой уровень;

– канальный уровень.

Протоколы этих уровней полностью реализуют функциональные возможности модели OSI. На стеке протоколов TCP/IP построено всё взаимодействие пользователей в IP-сетях. Стек является независимым от физической среды передачи данных, благодаря чему, в частности, обеспечивается полностью прозрачное взаимодействие между проводными и беспроводными сетями.

UDP – User Datagram Protocol.

UDP – это один из протоколов транспортного уровня, предназначенный для передачи сообщений между компьютерами. При помощи UDP датаграммы можно посылать другим хостам по IP-сети, не устанавливая предварительно специальных путей передачи или каналов передачи.

Одной из особенностей UDP является слишком простая модель передачи, из-за которой данные могут прийти дважды, не в том порядке или не прийти вовсе, но если придут, то придут, не нарушив целостность. Из-за этой особенности протокол используется в ситуациях, когда ошибки не проверяются и не исправляются.

FTP – File Transfer Protocol.

FTP является одним из старейших созданных протоколов прикладного уровня и используется для передачи ПО и доступа удалённым хостам и гарантирует передачу из-за квотируемого протокола. Протокол имеет архитектуру «клиент-сервер», что позволяет построить коммуникацию для передачи файлов и команд между сервером и клиентом.

Также FTP имеет сессионный тип работы, позволяя серверу запомнить текущее состояние, в отличие от HTTP, который не имеет такой функции. FTP имеет множественное подключение, которое позволяет ускорить удобность работы из-за того, что один подключённый канал является управляющим т.е. отправляющий команды и получающий ответы от сервера, а другие каналы являются транспортными через которые происходит сама передачи файлов в двух направлениях.

DNS – Domain Name System.

DNS – это протокол прикладного уровня, реализующий систему для получения информации о домене. Может использоваться для получения SRV-записей (данных о маршрутизации почты и узлах обслуживания протоколов домена.

Для повышения устойчивости системы используется множество серверов, содержащих идентичную информацию, а в протоколе есть средства, позволяющие поддерживать синхронность информации, расположенной на разных серверах. Существует 13 корневых серверов, их адреса практически не изменяются [3].

Протокол DNS использует для работы TCP- или UDP-порт 53 для ответов на запросы. Традиционно запросы и ответы отправляются в виде одной UDP-датаграммы. TCP используется, когда размер данных ответа превышает 512 байт, и для AXFR-запросов.

В основе структуры DNS лежит структура иерархии имён и зон. Она подразумевает что ответственность за разные части имени домена лежат на разных организациях и сервер, владеющий доменом, отвечает только за свою часть имени.

HTTP – HyperText Transfer Protocol.

HTTP – часто используемый протокол передачи данных, изначально предназначенный для передачи гипертекстовых документов. Аббревиатура HTTP расшифровывается как HyperText Transfer Protocol. В соответствии со спецификацией OSI, HTTP является протоко-

лом прикладного (верхнего, 7-го) уровня. Актуальная на данный момент версия протокола, HTTP 1.1, описана в спецификации RFC 2616. Протокол HTTP предполагает использование клиент-серверной структуры передачи данных.

Клиент – это любой инструмент, который действует от имени пользователя. В основном эту роль выполняет веб-браузер, но помимо браузера это быть программы, используемые инженерами или веб-разработчиками для отладки своих приложений. Клиент всегда инициирует запрос, это никогда не делает сервер.

На другой стороне канала связи находится сервер, который обслуживает документ по запросу клиента. Хотя для пользователя сервер выглядит как одна виртуальная машина, на самом деле это может быть набор серверов, разделяющих нагрузку. С другой стороны, несколько серверов могут быть расположены на одной и той же машине. При HTTP/1.1 и заголовке Host они могут даже использовать один и тот же IP-адрес.

Клиентское приложение формирует запрос и отправляет его на сервер, после чего серверное программное обеспечение обрабатывает данный запрос, формирует ответ и передаёт его обратно клиенту. После этого клиентское приложение может продолжить отправлять другие запросы, которые будут обработаны аналогичным образом. Благодаря этой возможности клиент и веб-сервер могут обмениваться двоичными данными, хотя данный протокол является текстовым.

Также HTTP часто используется как протокол передачи информации для других протоколов прикладного уровня, таких как SOAP, XML-RPC и WebDAV [4]. В таком случае говорят, что протокол HTTP используется как «транспорт».

API многих программных продуктов также подразумевает использование HTTP для передачи данных – сами данные при этом могут иметь любой формат, например, XML или JSON.

Как правило, передача данных по протоколу HTTP осуществляется через TCP/IP-соединения. Серверное программное обеспечение при этом обычно использует TCP-порт 80 (и, если порт не указан явно, то обычно клиентское программное обеспечение по умолчанию использует именно 80-й порт для открываемых HTTP-соединений), хотя может использовать и любой другой.

NTP – Network Time Protocol.

NTP – это особый протокол для синхронизации времени в компьютерных системах по сетям передачи данных. Большой востребованностью NTP обязан активным развитием систем на основе Ethernet.

Из функций, которые может делать сервер времени, называют корректное формирование хронологии событий в системах управления для ведения логов, журналов, архивирования информации, построения трендов, графиков и т.д.

В системах видеонаблюдения сервер времени создает привязку отснятых видеозаписей к астрономическому времени. К тому же устройство может безошибочно сопоставлять и сравнивать информацию от разных информационных систем. К примеру, это могут быть системы видеоконтроля и системы безопасности.

Много протоколов информационного обмена используют метки времени напрямую в составе пакетов передаваемых данных. К таким протоколам можно отнести МЭК-101/104, которые применяются в современных системах телемеханики.

Одним из важных требований, предъявляемых в ряде промышленных приложений, являются требования информационной безопасности, исключающие выход в Интернет для выполнения функции синхронизации времени.

NTP использует иерархическую систему источников точного времени. Каждый уровень иерархии называется Stratum (слоем) и ему присваивается номер, начинающийся с 0 для эталонных часов на вершине иерархии. Сервер времени на слое N синхронизируется от серверов на уровне N-1. Число N представляет собой расстояние от эталонных часов и используется для предотвращения цикличности в процессе синхронизации.

В качестве эталонных часов на Stratum 0 выступают системы спутниковой навигации (ГЛОНАСС, GPS и пр.), атомные часы или радиопередатчики. Раз в секунду они генерируют

импульсный сигнал (1PPS), который вызывает прерывание и генерирует метку времени на подключенных устройствах. Устройства слоя 0 также известны как опорные часы. Серверы NTP не могут позиционировать себя в системе как Stratum 0. Если в пакете передачи данных в поле Stratum установлен 0, это указывает на неопределенный слой.

С момента появления протокола в 1985 году началось активное развитие и уже к 1992 году сменил четыре версии (от NTPv0 до NTPv3). Каждая новая версия добавляла функционал и оптимизировала его работу, но оставляла неизменным формат данных и сохраняла совместимость различных версий между собой. Последняя версия протокола была создана в 2010 году. NTP продолжает развитие и в наши дни, ведутся работы по созданию решения, технически схожего с более точным протоколом PTP (Precision Time Protocol).

SSH (Secure SHell – защищенная оболочка) – сетевой протокол прикладного уровня, предназначеный для безопасного удаленного доступа к UNIX-системам. Данный протокол эффективен тем, что шифрует всю передаваемую информацию по сети. По умолчанию, используется 22-й порт. В основном он нужен для удаленного управления данными пользователя на сервере, запуска служебных команд, работы в консольном режиме с базами данных.

Эта служба была создана в качестве замены не зашифрованному Telnet и использует криптографические техники, чтобы обеспечить, что всё сообщение между сервером и пользователем было зашифровано.

Технологии шифрования

Существует три различных технологий шифрования, используемых SSH:

– симметричное шифрование;

– асимметричное шифрование;

– хеширование.

Симметричное шифрование – это форма шифрования, где секретный ключ используется для шифрования и дешифровки сообщения как клиентом, так и хостом. Стоит отметить, что любой клиент имеющий ключ, может дешифровать передаваемое сообщение.

Асимметричное шифрование

В отличии от симметричного шифрования, асимметричное использует два отдельных ключа для шифрования и дешифровки. Эти два ключа также известны как приватный и публичный ключи. Вместе они формируют пару публичных-приватных ключей.

Хеширование

Одностороннее хеширование – это еще одна форма криптографии, которая используется в SSH. Такого рода хеширование отличается от двух упомянутых выше тем, что оно не предназначено для дешифровки. Оно создает уникальное значение фиксированной длины для каждого ввода, которое не показывает никакого общего поведения для его раскрытия. Это делает его практически невозможным для обратного преобразования.

Использование SSH подключения имеет ряд преимуществ:

Безопасная работа на удаленном сервере с использованием командной оболочки;

Использование разных алгоритмов шифрования (симметричного, асимметричного и хеширования);

Возможность безопасного использования любого сетевого протокола, что позволяет передавать по защищенному каналу файлы любого размера.

Использование SSH подключения имеет свой недостаток:

Протокол SSH не имеет средств защиты от действий злоумышленника, получившего root-доступ. Одной из мер предосторожности является ограничение использования режима root без острой необходимости.

Заключение. Выполнен полный анализ распространённых протоколов передачи данных в сети. Были рассмотрены их особенности и свойства. Определено, что каждый протокол передачи данных имеет свои преимущества и недостатки.

Предложено использовать определённые протоколы в зависимости от поставленной задачи. Таким образом, для обеспечения высокой скорости передачи данных без необходимости проверки на их целостность следует использовать протоколы UDP или IP. С другой стороны, для обеспечения целостности и надёжности передачи данных следует использовать протокол TCP/IP.

Список литературы

1. Модели OSI и TCP/IP // База знаний osLogic.ru [Электронный ресурс]. – 2022. Режим доступа: https://www.oslogic.ru/knowledge/245/modeli-osi-i-tcp-ip.htm. – Дата доступа: 26.03.2022.

2. Сетевые модели TCP/IP и OSI // Cisco Learning [Электронный ресурс]. – 2017. Режим доступа: https://ciscolearning.ru/basics/tcpip-osi.htm. – Дата доступа: 26.03.2022.

3. Domain Name System (DNS) IANA Considerations // tools.ietf.org. [Электронный ресурс]. – 2008. Режим доступа: https://datatracker.ietf.org/doc/html/rfc5395.htm. – Дата доступа: 27.03.2022.

4. Простым языком об HTTP // Habr [Электронный ресурс]. – 2014. Режим доступа: https://habr.com/ru/post/215117.htm – Дата доступа: 28.03.2022.

UDC 004.057.4

NETWORK COMMUNICATION PROTOCOLS

Aleshkevich P.A., Lihatsevich A.V., Ptashnik D.A.

Belarusian State University of Informatics and Radioelectronics Affiliate Minsk Radioengineering College, Minsk, Republic of Belarus

Byanova S.G. – teacher of the highest category of disciplines of a special cycle

***Annotation**. To transmit data between devices, that work in different networks, using of protocols and standards is required. Said protocols and standards are keeping computer network technical organization rules in order, which allows to perform communication in the network.*

***Keywords**. protocol, network, data, information, OSI model, Internet.*