

АЛГОРИТМ ТСП-ВЗАИМОДЕЙСТВИЯ ПРИ ОДНОНАПРАВЛЕННОЙ ПЕРЕДАЧЕ ДАННЫХ

Белорусский Государственный Университет, Минск, Республика Беларусь

В докладе обосновывается актуальность однонаправленной передачи данных по протоколу ТСП. Предлагается для обеспечения передачи данных через протокол ТСП необходимо использовать прокси-сервер, на котором специальное программное обеспечение принимает ТСП подключение, извлекает данные и однонаправленно передает их по протоколу UDP через оптическую гальваническую развязку в соответствии с разработанным алгоритмом.

Протокол ТСП [1] (Transmission Control Protocol) является одним из основных протоколов передачи данных в сетях. ТСП обеспечивает надежную и устойчивую передачу данных между компьютерами, поддерживая контроль над потоком данных и обеспечивая доставку пакетов в правильном порядке.

Актуальность ТСП взаимодействия в информационных системах обусловлена тем, что информационные системы являются сложными сетевыми структурами, состоящими из множества компонентов, которые должны взаимодействовать между собой. Примером такой системы может служить Интернет, в котором миллионы компьютеров и устройств взаимодействуют друг с другом, обмениваясь данными. ТСП обеспечивает надежную передачу данных между различными узлами сети, что является важным аспектом кодирования и передачи информации. Он предоставляет механизм проверки целостности данных, контроль ошибок и управление потоком данных [2]. Также ТСП обеспечивает эффективную передачу данных на больших расстояниях путем их разбиения на множество малых пакетов.

В современных информационных системах существует необходимость передавать различного рода информацию из одних информационных систем в другие. При использовании аппаратно-программных средств однонаправленной передачи данных решается вопрос безопасной передачи данных между информационными системами с разной степенью конфиденциальности. Многие информационные системы, например, критически важные объекты информатизации, передают данные с датчиков по протоколу ТСП, работа которого основывается на двунаправленном взаимодействии. Возникает необходимость в сборе информации и мониторинге ее источников с учетом необходимости обеспечения безопасной передачи данных (например, однонаправленной) и исключения потенциального вредоносного воздействия на источники критически важного объекта информатизации.

Однонаправленная передача данных предполагает, что устройство в компьютерных сетях может только передавать данные или только получать их. При этом устройство – источник данных может осуществлять их передачу одному или нескольким устройствам – приемникам, но последние не могут передавать данные источнику. Однонаправленная передача данных применяется для безопасной передачи информации, например, файлов, журналов событий, почтовых сообщений, промышленных протоколов, обновлений программного обеспечения [3]. Однако, в условиях однонаправленной передачи данных протокол ТСП напрямую работать не может. Требуемую функциональность аппаратно-программного средства однонаправленной передачи данных обеспечивает архитектурное решение [4], включающее 2 медиаконвертера, 2 прокси-сервера, оптический разветвитель (сплиттер). Прокси-сервер отправителя и прокси-сервер получателя обеспечивают однонаправленную передачу данных, работая на транспортном уровне UDP [5] модели OSI через медиаконвертеры.

Секция 2. Прикладные проблемы информатики

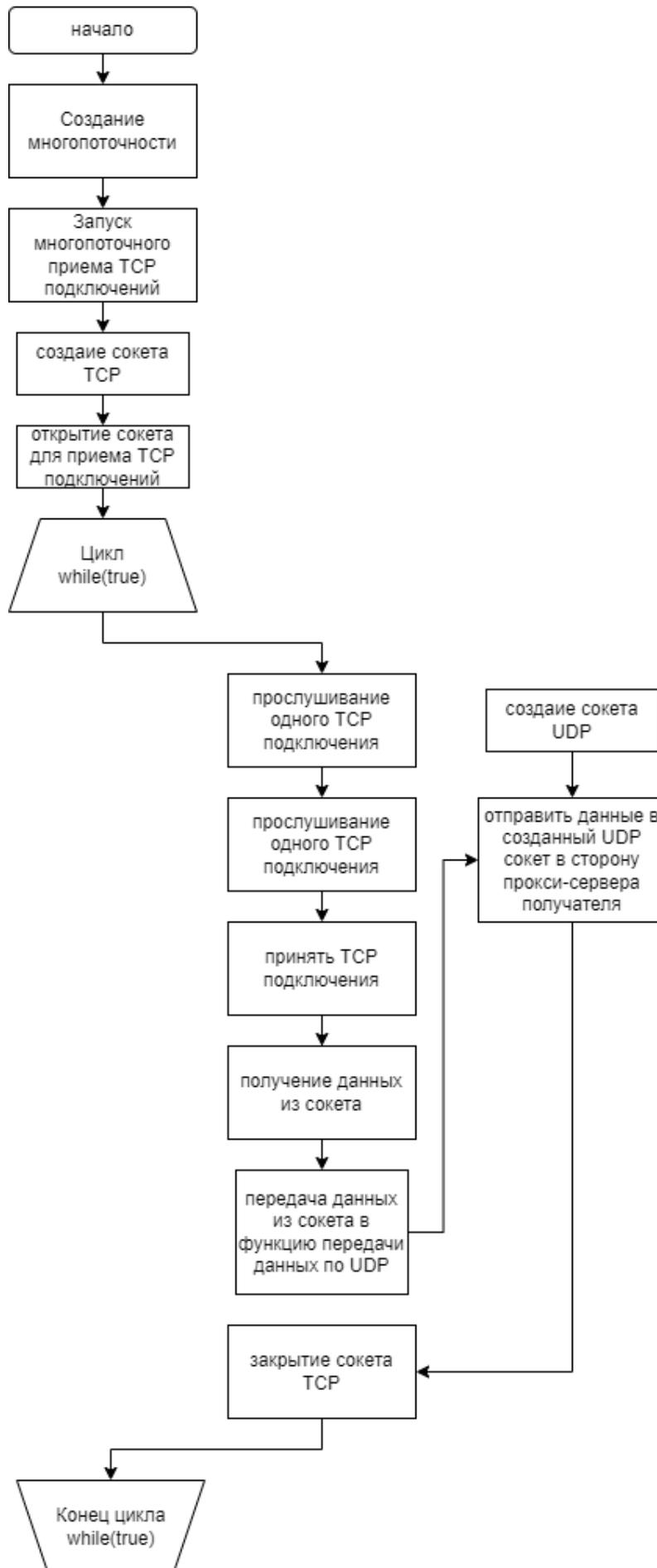


Рисунок 1 – Алгоритм однонаправленной передачи данных при работе с TCP протоколом.

Алгоритм однонаправленной передачи данных, реализованный на прокси-сервере отправителя (рисунок 1), включает следующие основные шаги:

1. Создание и запуск многопоточной работы для обеспечения приема одновременных TCP подключений от нескольких источников данных;
2. Создание TCP сокета [6] для приема подключений;
3. Установка параметров сокета, такие как адрес и порт прослушивания, размер буфера памяти с помощью функции `bind()`;
4. Запуск цикла обработки TCP подключений;
5. Разрешить серверу принимать соединения;
6. Ожидание клиентского подключения;
7. Прием данных от клиента размером с буфер памяти, определенного ранее;
8. Передача обработанных данных в функцию передачи по протоколу UDP;
9. Передача данных по протоколу UDP в предварительно созданный UDP сокет с параметрами, обеспечивающими передачу данных на прокси-сервер получателя;
10. Закрытие сокета TCP;
11. Окончание работы цикла обработки TCP подключений.

Предложенный алгоритм работы протокола TCP при однонаправленной передаче данных позволяет реализовать передачу данных в критически важных объектах информатизации [7].

Особенностями предложенного алгоритма являются:

1. Обеспечение информационной безопасности критически важного объекта информатизации и его элементов путем использования однонаправленной передачи данных;
2. Возможность реализации специального программного обеспечения с использованием различных языков программирования;
3. Использование прокси-серверов для реализации преобразования TCP в UDP и исключение необходимости доработки (перепрошивки) источников данных благодаря их работе на существующем стеке протоколов TCP/IP.

Список литературы

1. RFC 793: Transmission Control Protocol [Электронный ресурс]. Режим доступа: <https://www.ietf.org/rfc/rfc0793.txt>. Дата доступа: 20.04.2021
2. Олифер В.Г. Компьютерные сети. Принципы, технологии, протоколы: учеб. пособие для студ. вузов / В.Г. Олифер, Н.А.Олифер. 4-е изд. –М.; СПб.; Н. Новгород [и др.]: Питер, 2010. – 944 с.
3. Румас, Р. А., Воротницкий Ю.И. Анализ принципов работы сети при однонаправленной передаче данных в компьютерных сетях / Р. А. Румас, Ю.И. Воротницкий // Комплексная защита информации: материалы 26 науч.-практ. конф., Минск, 25–27 мая 2021 г. – Минск : Издатель Владимир Сивчиков, 2021. – С. 258–259.
4. Румас, Р. А., Воротницкий Ю.И. Архитектура средства однонаправленной передачи данных в компьютерных сетях / Р. А. Румас, Ю.И. Воротницкий // Компьютерные технологии и анализ данных (СТДА'2022): материалы III Междунар. науч.-практ. конф., Минск, 21–22 апр. 2022 г. / Белорус. гос. ун-т. – Минск : БГУ, 2022. – С. 132–135.
5. RFC 768: User Datagram Protocol [Электронный ресурс]. Режим доступа: <https://www.ietf.org/rfc/rfc0768.txt>. Дата доступа: 20.04.2021.
6. Socket – низкоуровневый сетевой интерфейс [Электронный ресурс]. Режим доступа: <https://docs.python.org/3/library/socket.html> Дата доступа: 01.11.2022
7. О некоторых мерах по совершенствованию защиты информации: Указ Президента Респ. Беларусь, 16 апр. 2013 г., № 196 // Нац. реестр правовых актов Респ. Беларусь. – 2013. – 1/14225.