



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)**

Факультет «Информатика и вычислительная техника»

Кафедра «Кибербезопасность информационных систем»

И.о. зав. кафедрой

«КБИС»

Д.А. Короченцев

«__»

2022 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Тема: «ПРОГРАММНОЕ СРЕДСТВО ОБНАРУЖЕНИЯ УСТРОЙСТВ ПРОСЛУШИВАНИЯ
СЕТЕВОГО ТРАФИКА В ОПТОВОЛОКОННОЙ СЕТИ»

Специальность 10.05.01 Компьютерная безопасность

Специализация Математические методы защиты информации

Обозначение ВКР 10.05.01.520000.000

Группа ВКБ62

Обучающийся

подпись, дата

А.А. Андриющенко

Руководитель ВКР

подпись, дата

доцент Н.В. Болдырихин

Нормоконтроль

подпись, дата

доцент Р.В. Егорова

Ростов-на-Дону

2022



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)**

Факультет «Информатика и вычислительная техника»
Кафедра «Кибербезопасность информационных систем»

И.о. зав. кафедрой «КБИС»

Д.А. Короченцев

подпись

«__» _____ 20__ г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

Тема: «ПРОГРАММНОЕ СРЕДСТВО ОБНАРУЖЕНИЯ УСТРОЙСТВ ПРОСЛУШИВАНИЯ
СЕТЕВОГО ТРАФИКА В ОПТОВОЛОКОННОЙ СЕТИ»

Обучающийся Андриющенко Александр Александрович

Обозначение ВКР 10.05.01.520000.000

Группа ВКБ62

Тема утверждена приказом по ДГТУ от «10» января 2022 г. № 10-ЛС-О

Срок представления ВКР к защите «__» _____ 2022 г.

Исходные данные для выполнения выпускной квалификационной работы:

Научные статьи и техническая литература по устройству оптоволоконных сетей, возможным угрозам, способам защиты оптоволоконных соединений, рефлектограммам.

Материалы исследований, проводимых в ходе преддипломной практики.

Содержание выпускной квалификационной работы

Введение:

Актуальность работы, цель исследования, научная задача, новизна.

Наименование и краткое содержание разделов:

1 Теоретические сведения.

Устройство оптоволоконного кабеля. Потенциальные места съема информации с волоконно-оптической линии. Явление полного внутреннего отражения и потери при соединении волоконно-оптических световодов. Защита информации на волоконно-оптических линиях. Анализ рефлектограммы оптического кабеля. Анализ существующих программных средств. Вывод к главе.

2 Алгоритмическая реализация программного средства.

Общая концепция программного средства. Таблица комментариев для событий на оптоволоконной линии. Алгоритм сравнения рефлектограмм для обнаружения прослушивающих устройств.

3 Программная реализация программного средства для обнаружения прослушивающих устройств сетевого трафика на оптоволоконной линии связи.

Обоснование выбора среды разработки и языка программирования. Предлагаемый алгоритм работы пользователя с программой. Демонстрация работы программного средства.

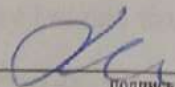
Заключение:

Основные выводы о проделанной работе, оценка достижения цели.

Перечень графического и иллюстративного материалов:


Презентация выпускной квалификационной работы на тему: «Программное средство обнаружения устройств прослушивания сетевого трафика в оптоволоконной сети»

Руководитель ВКР


подпись, дата

доцент Н.В. Болдырихин

Задание принял к исполнению


подпись, дата

А.А. Андрюшин

Аннотация

Данная работа направлена на разработку программного средства, которое позволяет обнаруживать прослушивающие устройства сетевого трафика в оптоволоконной сети методом анализа рефлектограмм. В выпускной квалификационной работе приводятся примеры уже существующих на рынке аналогов программных продуктов, способных открывать файлы рефлектограмм. В виде блок-схем представлены детали разрабатываемого программного средства. Приведена демонстрация работы программы, в результате которой можно сделать вывод о возможности внедрения в компании.

Объем работы – 73, количество иллюстраций – 23, таблиц – 1, приложений – 2, использовано источников – 15.

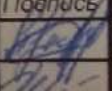
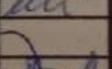


Annotation

This work is aimed at developing a software tool that allows detecting listening devices of network traffic in a fiber-optic network by analyzing reflectograms. In the final qualifying work, examples of analogues of software products that already exist on the market, capable of opening files of revlectograms, are given. The details of the software being developed are presented in the form of flowcharts. A demonstration of the program is given, as a result of which it is possible to conclude about the possibility of implementation in the company.

The volume of textual material - 73, the number of illustrations - 23, tables - 1, appendices - 2, sources used - 15.

Содержание

Введение	5
1 Теоретические сведения.....	8
1.1 Устройство оптоволоконного кабеля	8
1.2 Потенциальные места установки прослушивающих устройств	11
1.3 Явление полного внутреннего отражения и потери при сварке оптоволоконных соединений.....	13
1.4 Защита информации на оптоволоконных линиях связи.....	18
1.5 Анализ рефлектограммы оптического кабеля	19
1.6 Существующие программные средства	25
2 Алгоритмическая реализация программного средства	28
2.1 Таблица комментариев для событий на оптоволоконной линии	28
2.2 Анализ файла формата «SOR».....	31
3 Программная реализация программного средства для обнаружения прослушивающих устройств сетевого трафика на оптоволоконной линии	40
3.1 Обоснование выбора среды разработки и языка программирования...	40
3.2 Предлагаемый алгоритм работы пользователя с программой	41
3.3 Демонстрация работы программного средства.....	44
Заключение	50
Перечень использованных информационных ресурсов.....	51
Приложение А Техническое задание	53
Приложение Б Листинг	57

<h3 style="margin: 0;">10.05.01.520000.000 ПЗ</h3>				
Изм.	Лист	№ докум.	Подпись	Дата
Разраб.		Андрющенко А.А.		
Провер.		Болдырихин Н.В.		
Н.Контр		Егорова Р.В.		
Утверд.		Короченцев Д.А.		
<p>«Программное средство обнаружения устройств прослушивания сетевого трафика в оптоволоконной сети»</p> <p>Пояснительная записка</p>				
		Лит	Лист	Листов
		1	4	4
<p>Кафедра «КБИС» ДГТУ</p>				



Введение

Хотя на сегодняшний день существует множество способов передачи информации использующих множество видов беспроводных технологий, но подавляющее большинство линий передачи информации построены на основе обычных кабелей. Сейчас оптоволоконные кабели используются в, примерно, 80% случаев. В силу того, что оптоволоконный кабель использует свет вместо электричества, большинство проблем, присущих медному кабелю, такие как перекрестные помехи и электромагнитные помехи полностью устраняются.

До недавнего времени считалось, что волоконно-оптический кабель является одним из самых защищенных способов передачи информации на дальние расстояния, но по факту он не имеет защиты от возможных атак. Так как на сегодняшний день оптоволоконными линиями передачи информации пользуются не только обычные люди, а так же структуры, которые передают огромное количество критически важной информации, защита оптоволоконных кабелей является важной.

Подключением к оптоволокну можно назвать процесс, при котором безопасность оптического канала может быть нарушена путём вставки сплиттера или извлечением световой информации при помощи специальных ответвителей. Установка прослушивающих устройств может осуществляться как интрузивным так и неинтрузивным способами. Первый способ требует физического нарушения целостности оптоволокна и дальнейшего подключения оборудования для съема информации. Второй способ может выполнять съем информации без нарушения данных.

В настоящее время было обнаружено лишь несколько несанкционированных подключений к оптоволоконным линиям передачи информации. Так как при использовании неинтрузивного метода снятия информации оборудование работает в штатном режиме, момент подключения устройств прослушивания может быть утерян, а обнаружение такого

					10.05.01.520000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		5

устройства без подключения специального оборудования невозможно. Таким образом, на основе вышеописанного можно сделать вывод, что основными трудностями злоумышленника являются шифрование и расходы на оборудование.

Целью выпускной квалификационной работы является разработка программного средства обнаружения устройств прослушивания сетевого трафика в оптоволоконной сети.

Для достижения поставленной цели необходимо решить следующие задачи:

- провести анализ существующих программно-аппаратных решений обнаружения устройств прослушивания сетевого трафика в оптоволоконной сети;
- разработать алгоритмическое обеспечение программного средства обнаружения устройств прослушивания сетевого трафика в оптоволоконной сети;
- провести программную реализацию средства обнаружения устройств прослушивания сетевого трафика в оптоволоконной сети.

Объект исследования – оптоволоконная сеть.

Предмет исследования: процесс обнаружения прослушивающего устройства в оптоволоконной сети при помощи рефлектограмм.

Дипломная работа состоит из трех глав: теоретические сведения, алгоритмическое конструирование программного средства, демонстрация работы программного средства для обнаружения прослушивающих устройств в оптоволоконной линии.

Первая глава содержит шесть параграфов: устройство оптоволоконного кабеля, потенциальные места установки прослушивающих устройств, явление полного внутреннего отражения и потери при сварке оптоволоконных соединений, защита информации на оптоволоконных линиях связи, анализ рефлектограммы оптического кабеля, существующие программные средства

					10.05.01.520000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

для работы с рефлектограммами.

Вторая глава состоит из трех параграфов, которые объясняют алгоритм работы программы, каким образом происходит парсинг данных из рефлектограмм. Также в данной главе рассматривается таблица комментариев для событий затуханий на оптоволоконной линии и алгоритм сравнения для обнаружения прослушивающих устройств.

Третья глава посвящена демонстрации работы программы на уже построенных линиях связи. В данной главе обосновывается выбор среды разработки и языка программирования. Так же был предложен алгоритм работы пользователя с данным программным средством.

После основной части выпускной квалификационной работы присутствуют разделы такие как «Заключение» и «Перечень использованных информационных ресурсов». Так же в конце представлены два приложения с техническим заданием и листингом программы.

					10.05.01.520000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		7

1 Теоретические сведения

1.1 Устройство оптоволоконного кабеля

Каждый элемент оптоволокна представляет собой нечто вроде стеклянной трубы, которая в свою очередь находится в трубке с зеркальной поверхностью или обернута металлической фольгой, которая как раз и служит своеобразным экраном от потерь данных и физического разрушения кабеля. Это значит, что когда свет попадает в нее, то отражается от границ жил и проходит все дальше.

Стоит отметить, что расстояние, на которое будет передаваться информация, зависит не только от самого кабеля, но и от самого источника сигнала [1]. То есть чем он мощнее, тем большее расстояние сигнал сможет преодолеть.

Виды оптических кабелей:

- одномодовый;
- многомодовый.

В первом случае диаметр сердечника равен примерно 8,5 мкм, а во втором 51 или 62 мкм.

От вида кабеля напрямую зависит степень затухания передаваемого сигнала. Первый тип способен без потерь передавать данные на дистанцию до 9-10 километров. А второй – всего на 2-3 километра. При этом одномодовые оптические кабели обладают пропускной способностью до 100 Гб/с на км и используются все чаще и чаще. Они подразделяются на 3 типа:

- стандартная (SF, SM или SMF);
- со смещенной дисперсией (DS или DSF);
- с ненулевой смещенной дисперсией (NZ, NZDSF или NZDS).

Два последних типа используются на гораздо дальние дистанции, чем стандартное оптоволокно.

					10.05.01.520000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8

Сейчас ведутся исследования, в результате которых скоро появится возможность передавать данные со скоростью, достигающей до 160 Гбит/с.

На сегодняшний день существует множество технологий производства волоконно-оптических кабелей. Основное отличие между ними заключается в материале, из которого изготавливается ядро. Это может быть:

- кварцевое стекло;
- пластик;
- полимерные материалы.

От этого зависят не только пропускные характеристики, но и конечная стоимость оптоволоконного кабеля.

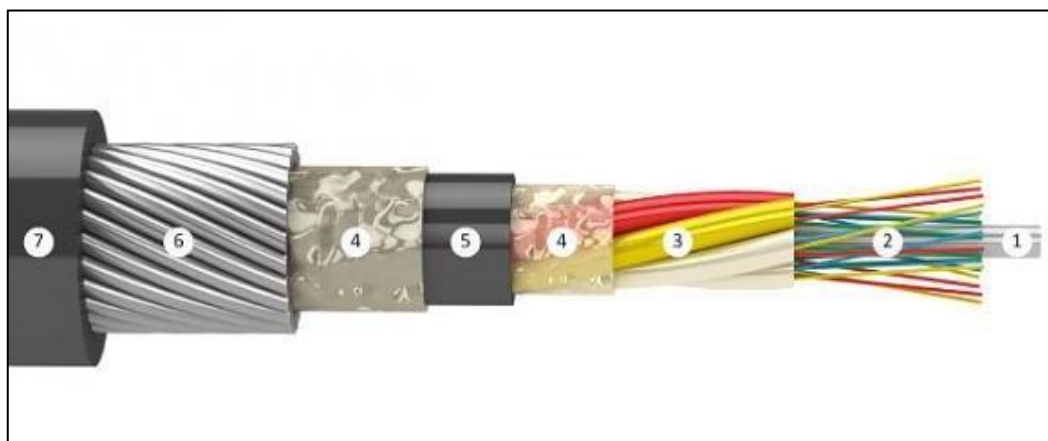


Рисунок 1.1 – Строение оптоволоконного кабеля

Строение оптоволоконного кабеля:

- центральный стержень;
- оптоволоконно;
- пластиковые модули;
- пленка с гидрофобным гелем;
- внутренняя оболочка из полиэтилена;
- броня (армирование);
- наружная полиэтиленовая оболочка.

По оптическому волокну могут распространяться два типа лучей. Первый тип называется меридиональными лучами. Меридиональные лучи — это лучи, проходящие через ось оптического волокна. Меридиональные лучи используются для иллюстрации основных свойств передачи оптических волокон. Второй тип называется косыми лучами. Косые лучи — это лучи, которые проходят через оптическое волокно, не проходя через его ось.

Меридиональные лучи могут быть классифицированы как связанные и несвязанные лучи. Связанные лучи остаются в сердцевине и распространяются вдоль оси волокна. Связанные лучи распространяются по волокну за счет полного внутреннего отражения. Несвязанные лучи преломляются сердцевинной оболочкой волокна. Возможный путь связанных и несвязанных лучей в волокне со ступенчатым показателем преломления. Сердцевина волокна со ступенчатым показателем преломления имеет показатель преломления n_1 . Оболочка ступенчатого показателя имеет показатель преломления n_2 , который меньше n_1 . Предполагается, что сердцевина-оболочка идеальна. Однако дефекты на границе сердцевины-оболочки вызовут преломление части связанных лучей из сердцевины в оболочку. Световые лучи, преломленные в оболочке, в конечном итоге выходят из волокна. В целом меридиональные лучи подчиняются законам отражения и преломления.

Поскольку среда рядом с волокном в точке запуска обычно представляет собой воздух, n_0 равно 1,00. Тогда мощность просто равна $\sin^2 \alpha$. Числовая амплитуда — это удобный способ измерения светосилы оптического волокна. Она используется для измерения эффективности передачи мощности между источником и волокном. Высокая числовая амплитуда указывает на высокую эффективность соединения источника с волокном. Типичные значения числовой апертуры находятся в диапазоне от 0,20 до 0,29 для стеклянных волокон. Пластиковые волокна обычно имеют более высокую числовую

					10.05.01.520000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		10

амплитуду. Числовая амплитуда для пластиковых волокон может быть выше 0,50.

Кроме того, обычно используется для определения многомодовых волокон. Однако для малых диаметров сердцевины, например, в одномодовых волокнах, лучевая теория не работает. Лучевая теория описывает только направление, которое плоская волна принимает в волокне. Лучевая теория исключает любые свойства плоской волны, которые мешают передаче света по волокну. В действительности плоские волны интерферируют друг с другом. Поэтому в оптическом волокне могут распространяться только определенные типы лучей. Оптические волокна могут поддерживать только определенное количество управляемых мод. В волокнах с малой сердцевиной число поддерживаемых мод равно одному или нескольким модам. Теория мод используется для описания типов плоских волн, способных распространяться по оптическому волокну.

Возможный путь распространения косых лучей представляет собой вид под углом, а вид В — вид спереди. Косые лучи распространяются, не проходя через центральную ось волокна. Угол приема косых лучей больше, чем угол приема меридиональных лучей. Это условие объясняет, почему косых лучей больше, чем меридиональных. Косые лучи часто используются при расчете приема света в оптическом волокне. Добавление косых лучей увеличивает светоемкость волокна. В крупных волокнах увеличение может быть значительным.

1.2 Потенциальные места установки прослушивающих устройств

Использование оптоволоконного кабеля имеет ряд преимуществ перед устаревшим медным кабелем, такие как: устойчивость к помехам и наводкам, более высокая скорость передачи данных, меньшее затухание сигнала. На сегодняшний день уже можно сказать, что оптоволоконный линии передач

					10.05.01.520000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		11

заменяли все существующие магистральные линии передачи информации. В связи с этим возникает проблема защиты информации в оптоволоконных линиях передачи данных. В начале необходимо определить возможные места установки устройств прослушивания сетевого трафика.

Расшифровка обозначений на рисунке 1.2:

- 1) устройство передачи оптического сигнала;
- 2) устройство приёма информации;
- 3) оборудование мультиплексирования;
- 4) оптоволоконный кабель;
- 5) места сварки оптоволоконных кабелей;
- 6) соединительная муфта;
- 7) пункт усиления оптического сигнала.

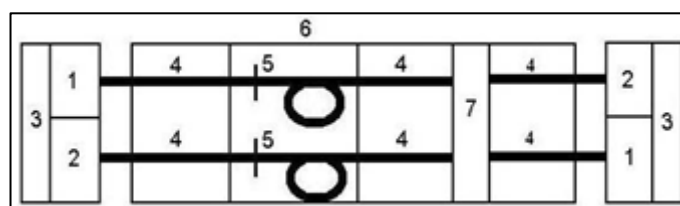


Рисунок 1.2 - Потенциально-возможные места установки прослушивающего устройства

На точках под номером 1,2,3,7 основной защитой является их расположение на автоматических телефонных станциях. Устройства усиления оптического сигнала, так же зачастую размещают внутри объектов, которые обеспечивают защиту от несанкционированного доступа. В итоге атака на данные точки вряд ли произойдет, из-за необходимости получения прямого доступа к оборудованию злоумышленником.

Так как устройства, позволяющие производить съём данных с оптоволоконной линии, стоят на вооружении инженеров компаний, предоставляющих подключение к интернету, руководителям данных предприятий необходимо тщательно производить отбор сотрудников, которым предоставляется доступ к данному оборудованию. Чаще всего

оптоволоконные прищепки используются для контроля качества передачи информации по оптоволоконным линиям.

Так как магистральные линии с использованием оптоволоконных сетей могут достигать огромного расстояния, именно поэтому они не обладают достаточной защищенностью от подключения злоумышленником. Оптоволоконные кабели, как описано в главе 1, имеют различные средства защиты от возможного воздействия окружающей среды и физических повреждений. Так же в зависимости от места прокладки кабеля устройство внешней защиты кабеля, как и его конструкция могут кардинально отличаться. Так, кабель рассчитанный на прокладку по канализации имеет более бронированное устройство, поэтому злоумышленнику будет сложнее разделить его в полевых условиях. А вот кабель, предназначенный для монтажа в зданиях, не имеет никакого бронирования, поэтому его разделка является более легкой и быстрой.

Так как сварное соединение (5) находится, чаще всего, в муфте (6), можно считать, что это одна точка где существует возможность установки устройств прослушивания. Так же злоумышленник может использовать некачественную сварку для снятия излучения, поэтому рекомендуется использовать комплект деталей для защиты сварных соединений.

Злоумышленник также может снимать информацию при помощи загиба оптоволокна, который возникает при укладке волокна в кассету муфты.

В итоге, менее защищенными являются соединения волокон в муфте и на строительных участках магистрали.

1.3 Явление полного внутреннего отражения и потери при сварке оптоволоконных соединений

Из-за того, что оптоволоконные линии для передачи информации используют свет, а не электрические сигналы, они изначально имеют более высокую степень защищенности от всевозможных несанкционированных

					10.05.01.520000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		13

действий злоумышленника. В оптическом волноводе электромагнитное излучение выходит за пределы волокна на расстояние не более длины волны при отсутствии внешнего воздействия на оптоволокно [2].

Стандартное оптоволокно состоит из: ядра присущим ему показателем преломления (зависит от использованного материала) n_1 и внешней оболочки с показателем преломления n_2 , при условии, что $n_1 > n_2$. Если луч сигнала падает на границу сред n_1 и n_2 под углом Q_1 . Луч изменяет свое направление, и в общем случае появляются преломленный и отраженный луч. Угол падения равен углу отражения ($Q_{отр}$):

$$Q_1 = Q_{отр}. \quad (1)$$

Углы падения (Q_1) и преломления (Q_2) согласно закону Снеллиуса связаны соотношением:

$$n_1 Q_1 = n_2 Q_2. \quad (2)$$

При определенном значении преломленный луч будет распространяться по поверхности раздела сред, при этом $Q_1 = 90$ градусов. Угол падения, при котором происходит указанное явление, называется критическим или предельным и определяется выражением:

$$Q_1 > Q_{крит}. \quad (3)$$

При $Q_1 > Q_{крит}$, (3) падающий луч полностью отражается от границы раздела сред и возвращается в среду падения. Это явление называется полным внутренним отражением. Для распространения лучей в оптически более плотной среде n_1 без проникновения в менее плотную n_2 , необходимо соблюдать условие (3).

					10.05.01.520000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		14

На оптоволоконном кабеле электромагнитное излучение экспоненциально падает. Поэтому, если злоумышленник, при использовании специальных устройств, будет производить загиб волокна и получится так, что угол падения станет меньше критического угла, то устройство сможет произвести перехват данных.

Оптоволоконные падения сигнала можно разделить на две группы:

- 1) некачественная сварка, при которой возможны: смещения двух стыкуемых волокон, наличие пузыря, который образуется при попадании на место сварки частиц пыли, неверный угол скола одного из волокон;
- 2) потери, которые образуются, если были сварены волокна с параметрами отличными от друг-друга;

На рисунке 1.3 показаны некоторые из описанных случаев.

Расшифровка рисунка 1.3:

- а) смещение при сварке;
- б) пузырь;
- в) различный угол скола;
- г) неверное расположение волокон при сварке;
- д) различие в параметрах.

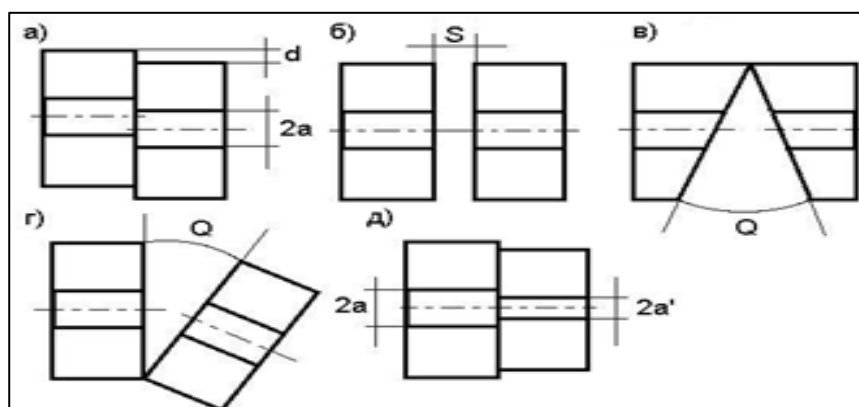


Рисунок 1.3 – Возможны потери при соединении оптических волокон

Возможность того, что на оптоволоконной линии установлено прослушивающее устройство, всегда существует. Существующие способы

прослушки сетевого трафика на волоконно-оптической линии можно разделить на несколько категорий [10]:

- 1 по возможному способу подсоединения:
 - 1.1 неинтрузивный;
 - 1.2 интрузивный;
 - 1.3 местный;
 - 1.4 протяженный.
- 2 по способу регистрации и усиления:
 - 2.1 пассивные – съем излучения с оболочки волокна;
 - 2.2 активные – съем излучения, при котором используются специальные устройства, которые меняют параметры сигнала;
 - 2.3 компенсационные – съем излучения, при котором используются специальные устройства, которые способны вводить излучение, компенсирующее потерю мощности при съеме информации.

Неинтрузивный местный несанкционированный доступ: основным и наиболее популярным способом является способ линзовой фокусировки вытекающих мод на изгибе волокна. Данный способ применяется в сварочных аппаратах для оптоволокна. Одним из наиболее популярных аппаратов для сварки оптоволокна является «Fujikura».

Интрузивный метод позволяет производить съем информации почти без потери уровня сигнала. Однако для такого подключения необходимо отключение линии связи на довольно большой промежуток времени, что в свою очередь могут заметить. Такой способ необходимо производить непосредственно во время монтажа линии.

Активные способы позволяют вывести через боковую поверхность оптоволокна излучение значительно большей мощности [2]. Однако при этом происходит изменение параметров распространяющегося по оптоволокну излучения, что может быть легко обнаружено сотрудниками службы безопасности. К способам этой группы относятся:

- физический изгиб оптоволокна;
- механическое вдавливание внешней оболочки в ядро;
- использование V-образного соединения;
- шлифование и растворение оболочки;
- подключение к оптоволокну фотоприемника с помощью делителя сигнала;
- термическое деформирование параметров оптоволокна;
- формирование трещин в оптоволокну.

Компенсационные способы сочетают в себе преимущества первых двух типов – сложность обнаружения и хороший уровень сигнала на выходе, но сопряжены с финансовыми трудностями при их реализации. Вывод излучения, формирование и обратный ввод через боковую поверхность должны осуществляться с коэффициентом передачи, близким к единице.

Интересным является также протяженный неинтрузивный съем информации, который можно осуществить или на пологом изгибе волокна или на прямом волокне под воздействием низких температур. Дело в том, что при низких температурах происходит изменение коэффициентов преломления стекла, в результате чего в сердцевине может повыситься уровень рассеяния. На основе данного способа работает так называемая оптоволоконная прищепка. Например ответвитель-прищепка FOD-5503 обеспечивает двунаправленное подключение к одномодовому волокну в 250-микронном покрытии без повреждения волокна. Совместно с другими волоконно-оптическими приборами ответвитель-прищепка используется для технического обслуживания, идентификации волокон и звуковой связи. FOD-5503 обеспечивает возможность подключения волоконно-оптических переговорных устройств в любой промежуточной точке волоконной линии, там, где есть доступ к волокну. Потери в ответвленном канале: <17 дБ (1550 нм), тип. 14 дБ <22 дБ (1300 нм), тип. 20 дБ, потери в сквозном канале: <7 дБ (1300/1550 нм), тип. 3 дБ.

1.4 Защита информации на волоконно-оптических линиях

На сегодняшний день множество компаний ведут разработки для создания оптоволоконных кабелей с высокой защитой от несанкционированного доступа. Выделяются три направления данных работ:

- разработка технических средств активной защиты от возможных подсоединений прослушивающих устройств сетевого трафика в оптоволоконные линии передачи данных;
- разработка средств мониторинга несанкционированного подключения к оптоволоконной линии;
- разработка средств защиты, в основу которых входит: маскировка передаваемого сигнала, добавление в линию связи помех и шумов, использование оптической и квантовой криптографии.

Большинство из них пока что небыли внедрены для использования в оптоволоконных линиях.

Ниже приведены возможности различных способов защиты оптоволоконных соединений:

1) Научная разработка конструкционных, механических и электрических средств защиты от несанкционированного доступа к оптическим кабелям и муфтам:

- затруднение доступа к оптоволокну посредством добавления промежуточной оболочки, металлоконструкций или же кевларовых нитей;
- использование внешней бронированной оболочки или металлических силовых элементов для питания датчиков, находящихся в разных концах линии (способ применим для магистральных трасс);
- применение комплекта защиты для места сварочных оптоволоконных соединений;
- использование многослойного оптического волокна со специальной структурой отражающих и защитных оболочек. Конструкция такого волокна

представляет собой многослойную структуру с одномодовой сердцевинной. Подобранное соотношение коэффициентов преломления слоев позволяет передавать по кольцевому направляющему слою многомодовый контрольный шумовой оптический сигнал. Связь между контрольным и информационным оптическими сигналами в нормальном состоянии отсутствует. Кольцевая защита позволяет также снизить уровень излучения информационного оптического сигнала через боковую поверхность ОВ (посредством мод утечки, возникающих на изгибах волокна различных участков линии связи). Попытки проникнуть к сердцевине обнаруживаются по изменению уровня контрольного (шумового) сигнала или по смещению его с информационным сигналом. Место несанкционированного доступа определяется с высокой точностью с помощью рефлектометра.

Подобную продукцию (многослойные оптические волокна с одномодовой сердцевинной и кольцевым сердечником) предлагает к поставке Инновационное предприятие «НЦВО-Фотоника».

- 2) специализированные системы диагностики состояния линии.
- 3) квантовая криптография.

1.5 Анализ рефлектограммы оптического кабеля

Устройство под названием рефлектометр или OTDR позволяет обнаруживать и выводить на графике порты, места сварочных соединений, заломы на оптоволоконном кабеле, трещины и устройства прослушки – события затухания на оптоволокне (Events) [5]. Все события могут как отражать сигнал, так просто создавать затухания. Начальные порты, неподсоединенный конец кабеля, обрыв волокна или же трещины которые образуют угол 90 градусов – все это примеры отражающих неоднородностей. В данных случаях происходит отражение луча обратно на приемник рефлектометра. На рефлектограмме данные события отображаются

					10.05.01.520000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		19

«всплесками». На рисунке 1.4 схематически представлен пример рефлектограммы на котором указаны такие события как:

- начало линии или же оптический разъем;
- потери в волокнах или релеевское рассеяние;
- трещина в волокне или отражающая неоднородность;
- сварное соединение волокон или иная неотражающая неоднородность;
- конец линии или отражение от торца волокна.

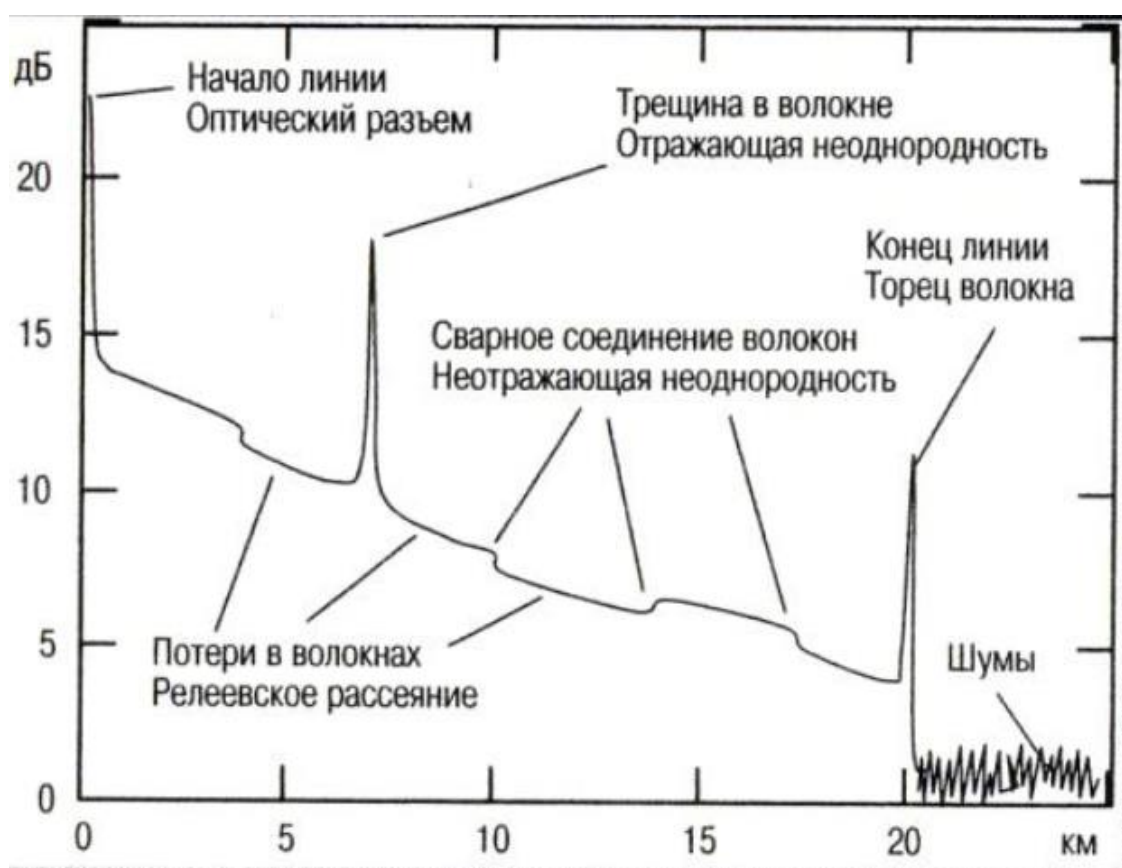


Рисунок 1.4 – Примеры событий на рефлектограмме

События, в которых доля излучения не превышает сотых долей процента, относятся к неотражающим[8]. Если в сегменте присутствуют коннекторные соединения APC с угловой полировкой, они будут отображаться на рефлектограмме как события без отражения, поскольку

имеют угол отклонения $7,5^\circ$. Также при должной полировке и аккуратности соединения APC могут не отразиться на данных рефлектограммы.

Угол наклона участков рефлектограммы между пиками характеризует погонное затухание по длине кабеля. Если кабель был смонтирован от точки А до точки Б без врезок или сварок, то график такого маршрута должен выглядеть как на рисунке 1.5.

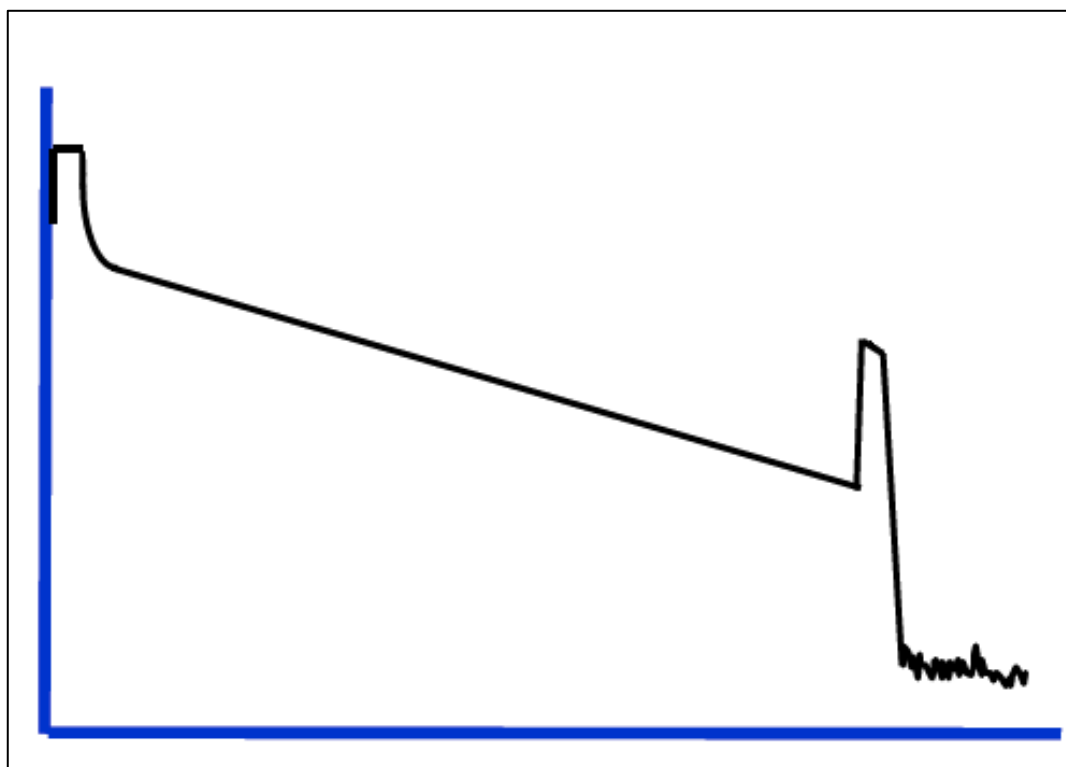


Рисунок 1.5 – Идеальное значение рефлектограммы

Если не проводить работы с данной смонтированной линией и в настройках рефлектометра изменить длину волны, то погонное затухание в дБ/км меняется, из-за чего изменяется также и угол наклона рефлектограммы. Пример одного и того же маршрута, но на разных длинах волны можно увидеть на рисунке 1.6.

Так же на рисунке 1.6 приведен пример рефлектограммы мультимодового оптического волокна.

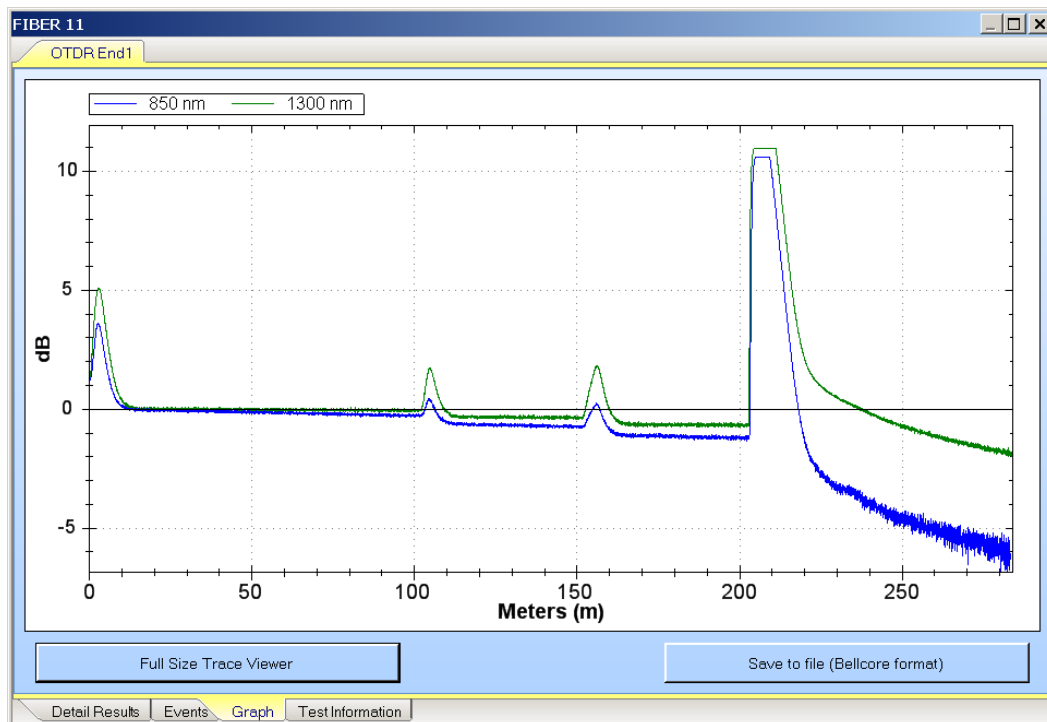


Рисунок 1.6 – Вывод рефлектограммы снятой на разных длинах волн

Анализ событий, а также корректность отображения графиков на рефлектограмме зависят от установленного программного обеспечения на OTDR. Так же существуют события, которые могут определяться рефлектометрами даже с самым базовым программным обеспечением.

На рисунке 1.7 наглядно видно отличие оптоволоконных событий в сочетании с отражением и такие события, как например сварка, которые не имеют отражения.

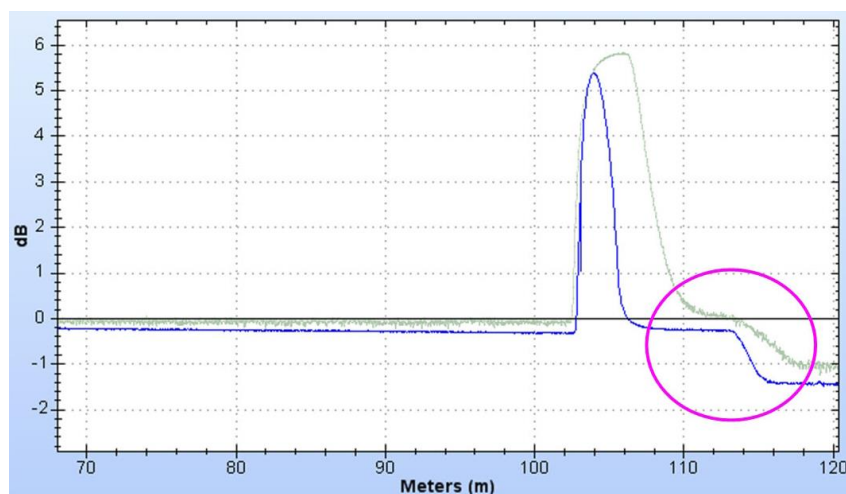


Рисунок 1.7 – Неотражающее событие (сварка, изгиб волокна)

Изм.	Лист	№ докум.	Подпись	Дата

10.05.01.520000.000 ПЗ

Лист

22

Для идентификации макроизгиба: следует измерить потери на коротких волнах существенно меньше, чем потери на длинных волна. Это неотражающее событие трактуется как макроизгиб, а не муфта.

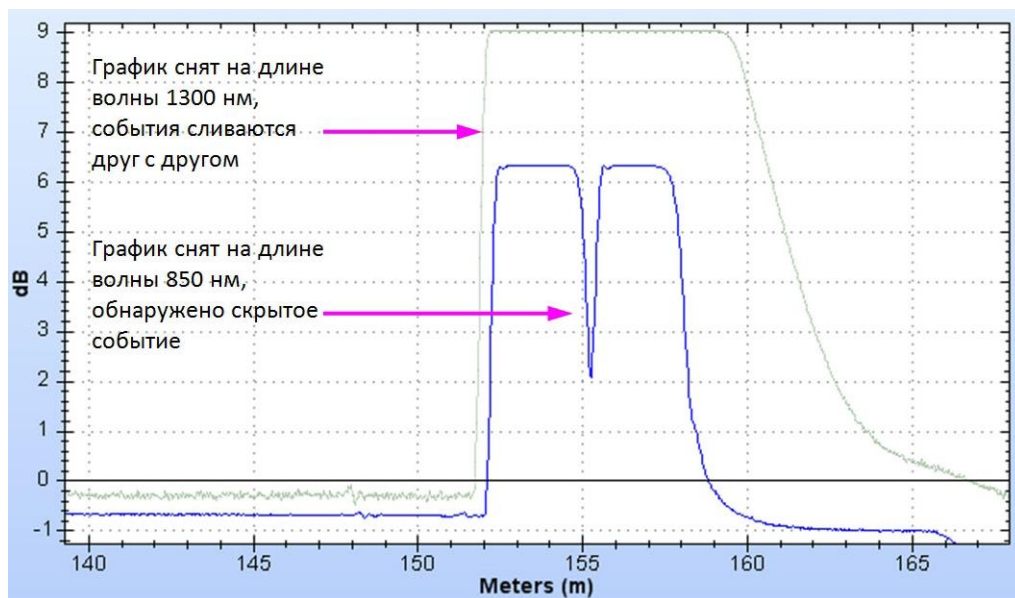


Рисунок 1.8 – Обнаружение скрытого макроизгиба

В инструкциях на рефлектометры производители указывают расстояния мертвых зон, а также степень затухания на разных длинах волн. Для того чтобы обнаружить события на мертвых зонах рефлектометра необходимо либо уменьшить длину волны, либо уменьшить ширину импульса. Так же такие затухания можно обнаружить, сняв рефлектограмму с другого конца оптоволоконного кабеля.

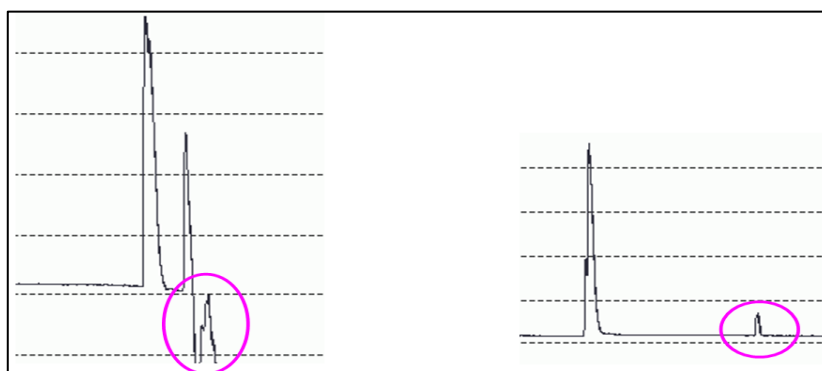


Рисунок 1.9 - Примеры событий снятых с разных сторон оптоволоконного кабеля

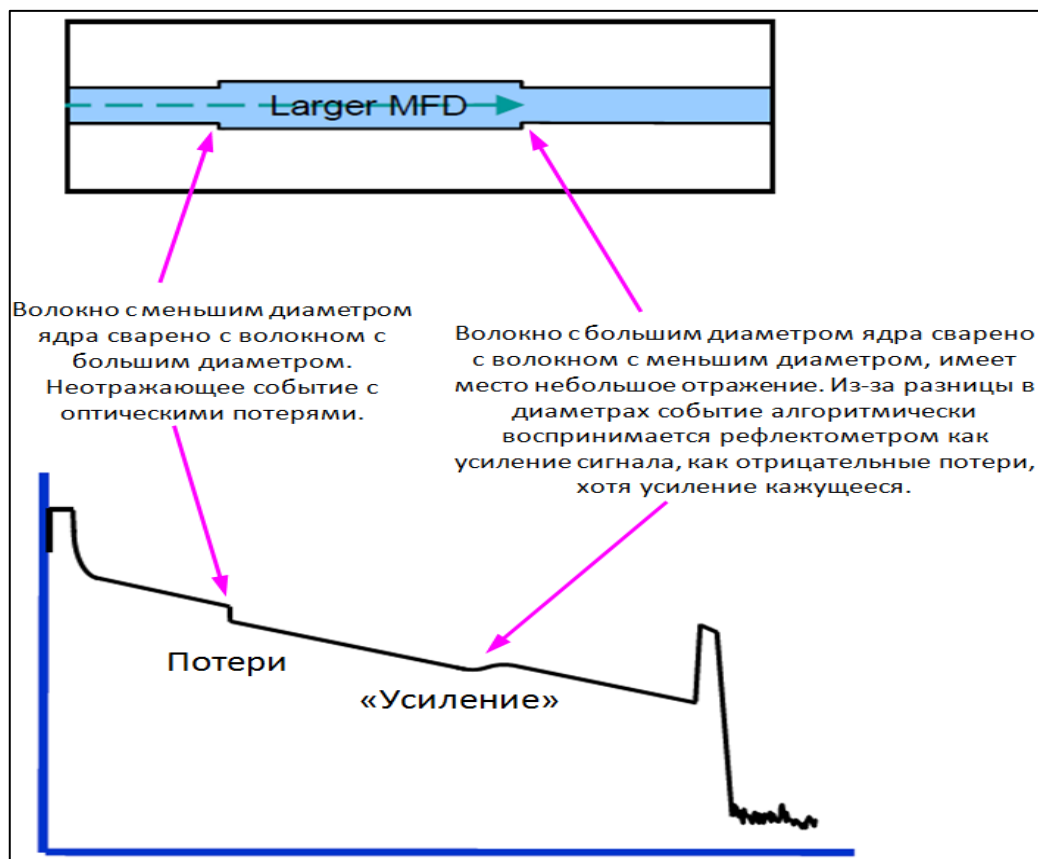


Рисунок 1.10 – Фантомное усиление сигнала и его возможная причина

Подобные фантомные усиления можно обнаружить путём снятия рефлектограммы с двух сторон проложенной оптоволоконной линии, после чего необходимо произвести усреднение потерь, так мы получим истинное значение затухания в данной точке. Если фантомное усиления существенное, необходимо проверить паспорта кабеля и при возможности заменить данную линию.

Большинство современных рефлектометров сглаживают шум на графике рефлектограммы, что облегчает визуальный осмотр оператору. Но иногда из-за того, что был выбран автоматический анализ, рефлектометр может неверно толковать события затухания. Так как не существует идеальных алгоритмов, при любых непонятных интерпретаций событий рефлектометром, советую использовать ручной режим.

В любом случае необходимо следовать прописанным инструкциям производителя.

1.6 Существующие программные средства

Программный продукт, который рассматривается довольно прост в использовании. SORTraceViewer имеет следующие возможности:

- открытие рефлектограммы и просмотр всех параметров, которые были записаны в файл;
- возможность печати или вывода в файл Excel отчета о рефлектограмме;
- работа с двунаправленными трассами;
- редактирование графиков рефлектограммы;
- возможность создания собственных рефлектограмм;
- исправление значения потерь на событии затухания.

Создатели данного продукта разработали удобный пользовательский интерфейс, который представлен на рисунке 1.11.

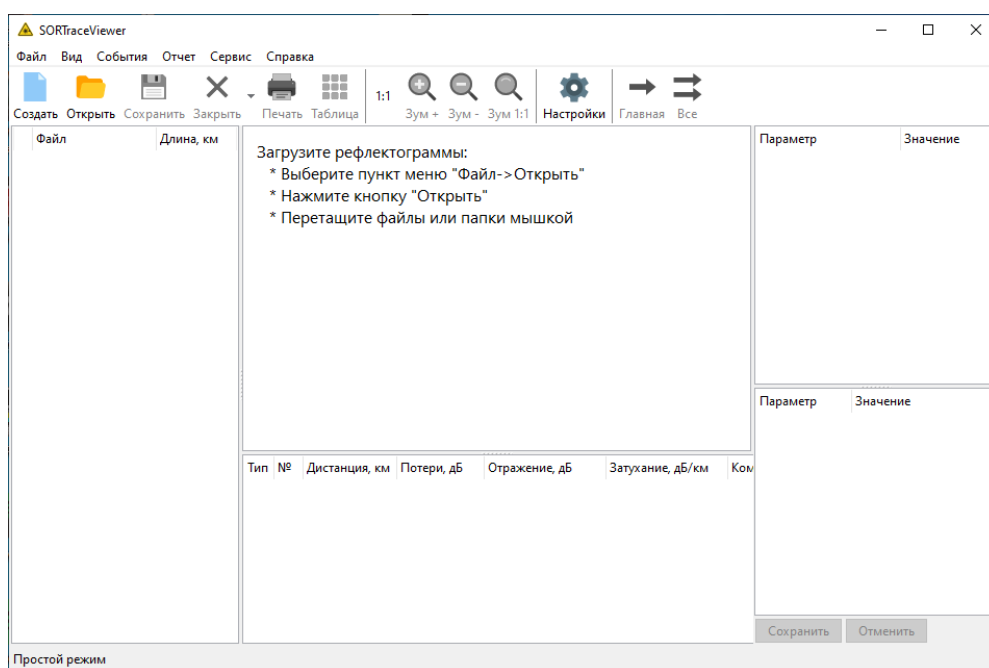


Рисунок 1.11 – Программное средство SORTraceViewer

Данный программный продукт распространяется по условно бесплатной системе, что является одним из главных преимуществ.

К недостаткам стоит отнести некорректный вывод графика рефлектограммы, при открытии двух и более файлов формата «.SOR» возможно экстренное закрытие программы.

SorEditor – визуальный редактор и вьюер файлов рефлектограмм в форматах Bellcore и SOR. Одним из главных плюсов данного программного средства является его постоянная поддержка. За последние 2 года было выпущено огромное количество обновлений различного рода. К возможностям SorEditor можно отнести следующее:

- визуализация графиков рефлектограмм;
- программный анализ оптоволоконных событий затухания;
- редактирование рефлектограмм;
- компенсация смещений;
- изменение наклона графика рефлектограммы;
- исправление порядка следования блоков в файлах.

Ниже, на рисунке 1.13 можно ознакомиться с главным интерфейсом программного средства

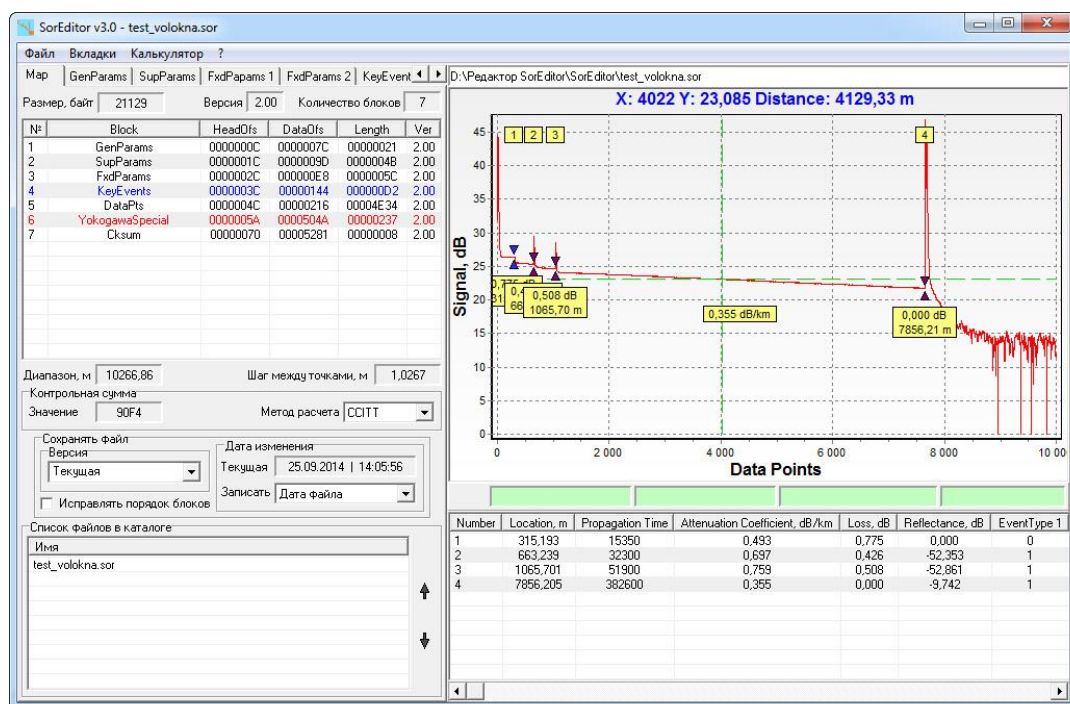


Рисунок 1.12 – Программный продукт SorEditor

К недостаткам программного средства можно отнести то, что оно сильно загружает центральный процессор и оперативную память персонального компьютера.

Вывод по главе

В данной главе были рассмотрены теоретические сведения об устройстве оптоволоконного кабеля, возможных угрозах оптоволоконной сети и способы защиты сетевого трафика.

Так же дана информация по одному из методов защиты информации, который был выбран в данной выпускной квалификационной работе как основной. Представлены уже существующие программные средства, которые выполняют идентичные функции с программой написанной в данной работе, такие как SORTraceViewer и SorEditor.

					10.05.01.520000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		27

2 Алгоритмическая реализация программного средства

2.1 Таблица комментариев для событий на оптоволоконной линии

На данный момент современные рефлектометры способны автоматически определять несколько событий.

Первым событием является «OTDR Port» - начальная точка измерений, связующий патч-корд или короб удлиняющего патч-корда (1-2 км). Определение мест событий и их затухания на данном участке невозможно. На данном сегменте рефлектометр может определить только общее затухание и уровень загрязнения порта. Если загрязненность оказалась слишком большой, то в таком случае прибор выведет информацию о том, что требуется очистка порта.

Патч-корд запуска – только если был применен. Использование связующего патч-корда позволит анализировать события, которые могут находиться в мертвой зоне рефлектометра. Так же его используют для измерения строительной длины оптоволоконно кабеля перед его монтажом.

Tail, Tail Cord - разъем шнура или перемычки «принимающего волокна», которое подключено к последней точки маршрута. Так же как и патч-корд, данное подключение позволяет измерять потери в мертвой зоне, но только на конце кабеля.

Loss, Loss Event – это оптические потери. Характеризуется достаточно малой долей отраженного луча, также может сочетать в себе потери с отражением. Такие события могут представлять собой сварное муфтовое соединение, изгиб или соединение APC/UPC. Коннекторное соединение со стандартной полировкой обычно приводит и к потерям, и к отражению части излучения в обратном направлении, поэтому оно классифицируется как отражающая неоднородность (Reflection, см. ниже). Если потери на большей длине волны превышают потери на меньшей длине волны, как правило, речь

					10.05.01.520000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		28

идет об изгибе волокна – это наиболее надежный способ идентифицировать макроизгиб. При подаче в сегмент узких импульсов некоторые события потерь могут трактоваться рефлектометром как события отражения.

Из вышеизложенного мы и будем составлять таблицу комментариев событий. Параметры прослушивающего устройства были взяты на сайте производителей.

Так как прослушивающие устройства создают оптическое событие на оптоволоконном соединении, необходимо определить в каких диапазонах программное средство будет выводить сообщение о возможном несанкционированном соединении. Так же необходимо определить диапазоны затуханий остальных событий на линии для облегчения анализа рефлектограммы для инженеров и монтажников, прокладывающих оптоволоконное соединение.

Для создания таблицы 1 были опрошены инженеры, ведущие инженеры таких компаний как: ПАО «Ростелеком», ПАО «МТС», ПАО «Вымпел-коммуникации». Так же был взят личный опыт в работе с оптоволоконными рефлектограммами. Для определения уровня затухания при использовании прослушивающего устройства были взяты данные производителя, так как покупка данных устройств на территории России запрещена.

Таблица 1 – Комментарии для событий на оптоволоконной линии

Значения затухания, дБ	Возможные причины	Необходимые действия
от ~ -0.1 до ~ -0.5	Фантомное усиление или погрешность прибора	Действий не требуется
от ~ -0.8 и далее	установлен усилитель сигнала или возможно фантомное усиление из-за различий во врезанном кабеле	если на линии присутствует усилитель, тогда действий не требуется. При его отсутствии требуется проверить прошлое событие и сравнить уровни затухания, если при их сложении сумма получается от 0.3 до 0.6, тогда можно судить о возможном различии кабелей

Окончание таблицы 1

Значения затухания, дБ	Возможные причины	Необходимые действия
от 0.1 до 1.5	Затухание, причиной может быть незначительный изгиб кабеля или сварка плохого качества	Если линия является подключением абонента не больше 1 км, исправления не требуются. Если линия является критически значимой или её расстояние превышает 1 км, требуется исправление
от 1.5 до 2	Значительное затухание, причина в значительном загибе кабеля или плохой сварке соединений	Если линия не превышает 500м исправления могут не понадобиться, в иных случаях требуется исправление
от 2 до 6	Критическое затухание, причина может быть в плохой сварке или критическим изгибом. Возможна установка прослушивающего устройства	Требуется проверка события и срочное устранение

Таким образом построена таблица событий и комментариев. Созданное программное средство при анализе событий следует алгоритму, представленному на рисунке 2.1.

Программа должна сначала проверить уровень затухания на существующем событии, далее провести сравнение затухания с данными заложенными в программу. После чего выводится комментарий о данном событии и возможных способов решения данной проблемы. Для того, чтобы скорректировать значения в таблице с затуханиями будет разработана подпрограмма, которая позволит операторам записывать свои данные на счет данных затуханий. Также планируется вывод данных о затухании на карту, для облегчения нахождения данных событий.



Рисунок 2.1 – Анализ событий

Если возможные причины, представленные в таблице, будут не соответствовать действительности, тогда пользователь может добавить в программу своё событие как уникальное и это будет учитываться при последующих анализах рефлектограмм

2.2 Анализ файла формата «SOR»

Формат данных SOR ("Стандартная запись OTDR") используется для хранения оптоволоконных данных OTDR (оптического рефлектометра во временной области). Формат определяется стандартом Telcordia SR-4731,

выпуск 2. Хотя это стандарт, он, к сожалению, не является открытым, поскольку особенности формата данных не являются общедоступными.

На самом деле существует две основные версии файлов OTDR SOR. Более ранние версии из Bellcore (в 1.x версии), новая версия 2.x. Файлы являются двоичными файлами данных, и все эти значения кодируются как с прямым порядком байтов или без знака целые числа с плавающей запятой представлены в масштабе целых чисел, то есть числа умножаются по какому-то признаку, как правило в степени 10, чтобы стать актуальной.

В обеих версиях данные расположены блоками; некоторые из них обязательны, некоторые являются дополнительными:

- map block (Map);
- general parameters block (GenParams);
- supplier parameters block (SupParams);
- fixed parameters block (FxdParams);
- key events block (KeyEvents);
- link Parameters block (LnkParams);
- data points block (DataPts);
- special proprietary block;
- checksum block (Cksum).

Map block - это первый блок, содержащий номер версии формата и подробную информацию о последующих блоках. Каждый отдельный блок в файле описывается своей собственной "картой", которая состоит из имени блока (строки), номера версии и размера блока (в байтах). Эти "карты" также определяют порядок, в котором блоки отображаются в файле; порядок может отличаться от поставщика к поставщику. Однако блок контрольной суммы всегда отображается как последний блок, и такое расположение имеет смысл для создания и последующего добавления контрольной суммы файла после ее вычисления. Блок общих параметров начинается с заголовка "GenParams\0" (строка, за которой следует

завершающий символ "\0"), затем два байта, указывающих язык (EN для английского, RU для русского). За этим следуют следующие поля (в дальнейшем все строки содержат завершающий символ '\0', если не указано иное):

- cable ID: string;
- fiber ID: string;
- fiber type: 2 byte unsigned integer;
- wavelength: 2 byte unsigned integer;
- location A (starting location): string;
- location B (ending location): string;
- cable code (or fiber type): string 8. build condition: 2 byte characters (no terminating '\0');
- user offset: 4 bytes integer;
- user offset distance: 4 bytes (only in version 2.x);
- operator: string;
- comments: string.

Блок параметров поставщика начинается со строки 'SupParams\0' в формате версии 2.x, в формате версии 1.x этот заголовок отсутствует. Поля в блоке параметров поставщика в формате версии 2.x следующие (все строки заканчиваются символом '\0'):

- supplier name;
- OTDR name;
- OTDR serial number;
- module name;
- module serial number;
- software version;
- other;
- eduk numb;
- modul sfp.

Блок фиксированных параметров начинается со строки "FxdParams\0" в формате версии 2.x; этот заголовок отсутствует в формате версии 1.x. Параметры в этом блоке часто указываются как "параметры измерения".

Поля для формата версии 2.x следующие (все являются целыми числами без знака, если не указано иное):

- 0-3: date/time : 4 bytes;
- 4-5: units : 2 characters;
- 6-7: wavelength : 2 bytes;
- 8-11: acquisition offset : 4 bytes integer;
- 12-15: acquisition offset distance : 4 bytes integer;
- 16-17: number of pulse width entries : 2 bytes;
- 18-19: pulse-width : 2 bytes (repeated);
- 20-23: sample spacing : 4 bytes (repeated);
- 24-27: number of data points in trace: 4 bytes (repeated);
- 28-31: index of refraction: 4 bytes;
- 32-33: backscattering coefficient: 2 bytes;
- 34-37: number of averages : 4 bytes;
- 38-39: averaging time: 2 bytes;
- 40-43: range : 4 bytes;
- 44-47: acquisition range distance : 4 bytes signed int;
- 48-51: front panel offset : 4 bytes signed int;
- 52-53: noise floor level : 2 bytes;
- 54-55: noise floor scaling factor : 2 bytes signed int;
- 56-57: power offset first point : 2 bytes;
- 58-59: loss threshold : 2 bytes;
- 60-61: reflection threshold : 2 bytes The supplier parameters block The fixed parameters block;
- 62-63 : end-of-transmission threshold: 2 bytes;
- 64-65 : trace type : 2 characters;

- 66-69: X1 : 4 bytes signed int;
- 70-73: Y1 : 4 bytes signed int;
- 74-77: X2 : 4 bytes signed int;
- 78-81: Y2 : 4 bytes signed int.

Поле даты/времени представляет собой 4-байтовое целое число без знака, представляющее собой время Unix (или POSIX) и представляющее собой количество секунд, прошедших с 00:00:00 UTC 1 января 1970 года. Два байта, которые следуют за ним, могут быть связаны с часовым поясом (бывают случаи, когда время, отображаемого считывателями OTDR, отличается на один час от того, что я получаю, интерпретируя его как время Unix), но до сих пор я не смог определить, как оно кодируется.

Блок ключевых событий начинается со строки 'KeyEvents\0' в формате версии 2.x; в формате версии 1.x этот заголовок отсутствует. Форматы версии 1.x и версии 2.x немного отличаются. Я начал с версии 2.x. Первые два байта, следующие за заголовком, представляют собой целое число без знака, представляющее собой общее количество событий. Каждое событие представляет собой фиксированную 42-байтовую запись, за которой следует строка комментария, заканчивающаяся '\0' (которая может быть пустой). Фиксированные 42 байта выглядят следующим образом:

- 00-01: event number (1, 2, 3, etc.); 2 bytes, unsigned integer;
- 02-05: time-of-travel; 4 bytes, unsigned integer;
- 06-07: slope; 2 bytes, signed integer;
- 08-09: splice loss; 2 bytes, signed integer;
- 10-13: reflection loss; 4 bytes, signed integer;
- 14-21: event type: 8 characters;
- 22-25: end of previous event;
- 26-29: beginning of current event;
- 30-33: end of current event;
- 34-37: beginning of next event;

– 38-41: peak point in current event.

Начало и конец (позиции) текущего, предыдущего или следующего события, а также пиковое положение текущего события являются целыми числами, которые также представляют время прохождения (в единицах 0,1 наносекунды). Перевод целых чисел в километры выполняется по той же формуле, что и кодирование расстояния события.

Наконец мы подходим к блоку точек данных, который кодирует саму кривую трассировки. Подобно другим блокам, блок начинается со строки заголовка "DataPts\0" в формате версии 2.x, но заголовок отсутствует в формате версии 1.x. Формат как для версии 1.x, так и для версии 2.x одинаков.

После заголовка (если применимо) блок точек данных начинается с 12 байт. Первые 4 байта представляют собой целое число без знака, которое представляет собой количество точек данных (это будет совпадать с количеством точек данных из фиксированных параметров). За этим следует 2 байта - количество трасс, соответствующее количеству записей ширины импульса в блоке Fxparams.

Следующие 4 байта - это повторение числовых точек данных, за которыми следуют еще 2 байта (целое число без знака), которое является коэффициентом масштабирования (почти всегда 1000). После начальных 12 байт идут данные трассировки OTDR (дБ в зависимости от расстояния). Каждая точка данных представляет собой целое число без знака в 2 байта. Умножьте на отрицательный коэффициент масштабирования, умноженный на 10^{-6} , чтобы перевести значение в дБ (при этом все значения преобразуются в ноль или отрицательные). Различные устройства чтения файлов OTDR SOR смещают данные по-разному; некоторые смещают данные так, чтобы максимальное значение составляло 0 дБ. Другие добавляют смещение, чтобы сделать минимальное значение 0 дБ. Если имеется более одной трассировки, шаблон повторяется: 4 байта для

					10.05.01.520000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		36

количества точек данных, 2 байта для коэффициента масштабирования, затем данные трассировки. Точки данных расположены на равном расстоянии друг от друга по значению "расстояние между образцами", указанному в блоке фиксированных параметров (после преобразования его в расстояние или "разрешение" измерения). Можно применить значение "начальная позиция волокна", указанное в блоке ключевых событий, но некоторые устройства чтения файлов OTDR SOR этого не делают.

Коэффициент масштабирования почти всегда равен 1000. В этом случае 2 байта на точку данных могут дать вам максимум динамический диапазон 65,535 дБ. Я лично никогда не сталкивался с файлом SOR, где этот коэффициент масштабирования не был 1000; информация о коэффициенте масштабирования была любезно сообщена мне читателем.

Блок контрольной суммы также начинается с заголовка "Cksum\0" в формате 2.x и отсутствует в формате 1.x. Само значение контрольной суммы составляет 2 байта (16 бит). Алгоритм вычисления контрольной суммы использует определенный 16-битный алгоритм (или функция) CRC (циклическая проверка избыточности). Для читателя, который не знаком с алгоритмами CRC, пожалуйста, ознакомьтесь с отличной статьей "Безболезненное руководство по алгоритмам обнаружения ошибок CRC" Росса Н. Уильямса (1993). Специфический аромат, используемый для формата OTDR SOR, иногда известен как "CRC-16/CCITT-FALSE".

Точный алгоритм вычисления контрольной суммы (как для версии 1.x, так и для версии 2.x) выглядит следующим образом: возьмите все содержимое файла, включая заголовок 'Cksum\0', в виде одной огромной двоичной строки и вычислите контрольную сумму по этой строке. Контрольная сумма будет составлять два байта (16 бит), которые затем добавляются в файл (после 'Cksum\0' заголовка). Однако два байта необходимо поменять местами, потому что соглашение в файле SOR заключается в том, чтобы хранить числа в меньшем порядке байтов.

Например, если контрольная сумма равна 0xD680, последние два байта файла SOR равны 0x80, 0xD6.

В некотором смысле это оказывается очень неловким. Очень интересным свойством алгоритма CRC является то, что если вы добавите два байта 16-битной контрольной суммы в файл (или строку), а затем запустите функцию CRC для него, контрольная сумма будет равна нулю! Но это работает только в том случае, если контрольная сумма добавляется в большом порядке байтов. Говоря по-другому, необходимо взять последние два байта файла SOR, поменять их местами и запустите функцию контрольной суммы на нем. Если контрольная сумма равна нулю, то все в порядке (или, скорее всего, все будет в порядке; код обнаружения ошибок CRC будет обнаруживает большинство запасных ошибок, но в конце концов не обнаруживает все ошибки). Было бы неплохо, если бы замена двух байтов контрольной суммы не была необходимой.

Мною был проверен этот алгоритм CRC-16 на многих версиях файлов 1.x и 2.x. Почти все они проверяются. Я считаю, что алгоритм верен, но мною были найдены исключения в немногих образцах файлов SOR из некоторых установленных программных программ, и есть одна конкретная программа, которая генерирует контрольную сумму совершенно по-другому (она не соответствует ни одному из обычно используемых алгоритмов CRC-16, или другие широко известные алгоритмы 16-битной контрольной суммы). С другой стороны, эта программа также принимает файлы SOR с "правильной" контрольной суммой.

Есть еще несколько частей формата SOR, которые нуждаются в уточнении, но я считаю, что почти вся кодировка, как мною было описано в данной выпускной квалификационной работе, должна быть правильной, благодаря многим людям, которые комментировали и писали мне с информацией и исправлениями. Простая (и крайне неэффективная) программа Pubotdrпрограмма в основном представляет собой реализацию

					10.05.01.520000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		38

результатов, описанных выше (хотя и неполных). Также доступна версия Python: pyOTDR. Вы можете свободно использовать эти программы и информацию в этой статье по своему усмотрению. Но еще раз, все это предоставляется бесплатно, без каких-либо гарантий и гарантий, используйте это на свой страх и риск. Который написал свою собственную программу просмотра трассировки SOR. Я хотел бы поблагодарить его здесь за исправление нескольких ошибок и объяснение нескольких других параметров в формате OTDR, в частности, в блоке KeyEvents и исправленном блок параметров (в этих двух разделах есть существенные исправления). В частности, я изначально неверно истолковал "расстояние между образцами" ("расстояние между образцами") как расстояние вместо времени; то, что я считал "магическим числом", на самом деле было половиной скорости света в вакууме. Инженер также сообщил мне информацию о коэффициенте масштабирования в разделе "Точка данных".

					10.05.01.520000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		39

3 Программная реализация программного средства для обнаружения прослушивающих устройств сетевого трафика на оптоволоконной линии связи

3.1 Обоснование выбора среды разработки и языка программирования

В качестве среды разработки мной была выбрана платформа Visual Studio ver.17. Данная платформа позволяет обеспечить максимальную эффективность ПО на протяжении всего жизненного цикла, начиная от проектирования и заканчивая внедрением в компании. Visual studio является кросс платформенной средой разработки. Так же данная платформа позволяет следить за использованием программой ресурсов центрального процессора, оперативной памяти и так далее.

Visual Studio ver.17 имеет простой, понятный и удобный интерфейс, так же лёгкий процесс установки. Данная платформа позволяет быстро и удобно устанавливать библиотеки через графический интерфейс. Также можно выбрать версию программного пакета и переключаться между ними. В Visual Studio имеется встроенный редактор кода и помощник по исправлению ошибок, который даёт возможность изучить информацию об исключениях. Данная функция полезна для проектов с большим количеством строк.

Программное средство в данной выпускной квалификационной работе написано на языке C++. Данный язык компилируемый, статически типизированный, который позволяет создать программу абсолютно любого уровня сложности. C++ включает огромный выбор различных стилей и технологий программирования. Синтаксис C++ был унаследован от более старого языка C. Данный язык программирования имеет богатую стандартную библиотеку, в которой имеются распространенные контейнеры и алгоритмы, ввод-вывод, поддержка многопоточности,

					10.05.01.520000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		40

различные выражения и другие возможности. Именно из-за данных качеств и был выбран данный язык программирования.

3.2 Предлагаемый алгоритм работы пользователя с программой

Программное средство в данной выпускной квалификационной работе в первую очередь разрабатывается для компаний, которые предоставляют услуги по монтажу оптоволоконных линий связи. В связи с этим было необходимо составить инструкцию работы с программой.

Для начала составим алгоритм действий для инженера или монтажника оптоволоконных сетей. В первую очередь, перед монтажом линий требуется проверить оптоволоконный кабель на целостность. Для этого требуется снять рефлектограмму с каждого волокна и записать её в памяти рефлектометра, чтобы облегчить данную задачу возможно использовать Fast-коннекторы.

Далее необходимо загрузить рефлектограммы в программное средство и провести анализ. Если никаких событий не было обнаружено, тогда можем производить монтаж линии.

После того как линия была смонтирована, снимаем проверочные данные при помощи рефлектометра на двух длинах волны 1550 и 1310 нм. Загружаем рефлектограммы в наше программное средство и производим анализ каждой рефлектограммы по отдельности. Если на длине не присутствуют события необходимо переименовать и записать данные измерения как основные. Или же если присутствуют замечания, провести исправления событий, используя подсказки и опыт других пользователей данного программного средства. Блок-схема данного алгоритма представлена на рисунке 3.1. Так же при определенных отличиях между фактической причиной и причинами, указанными в комментарии, будет возможность внести свой личный комментарий по данному затуханию. При большом распространении программного средства, оно будет коррелироваться.

					10.05.01.520000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		41

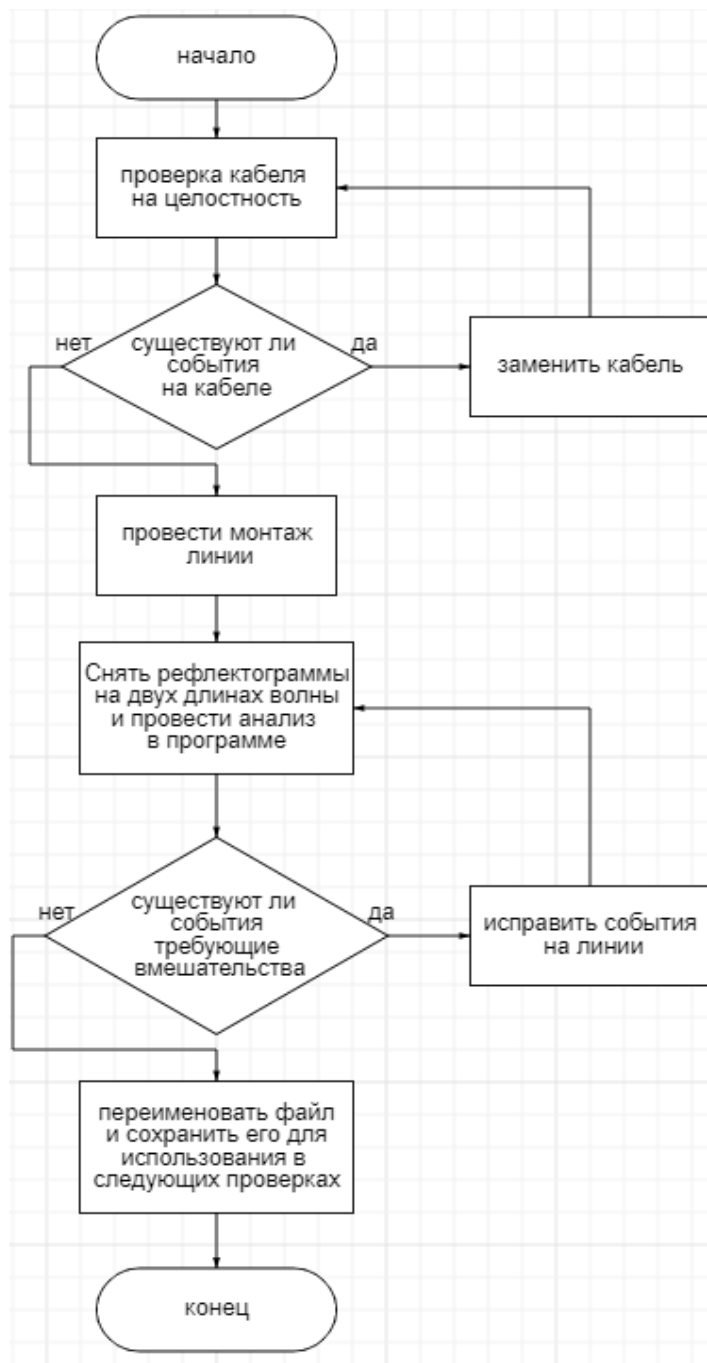


Рисунок 3.1 – Алгоритм работы с программой для инженеров и монтажников

Дальнейшие действия рекомендуется проводить сотрудниками службы безопасности, если такой отдел присутствует в компании. Если же нет, то мероприятия по повторной проверке линий требуется возложить на ведущих инженеров компании. Так как прослушивающее устройство может быть установлено в любой промежуток времени после монтажа и проверки линии, рекомендуется проводить повторные снятия рефлектограммы с

оптоволоконных линий не реже чем один раз в месяц. Перед снятием рефлектограммы требуется отключить подключенное оборудование для корректного получения информации. Рефлектограммы должны быть записаны на двух длинах волн 1310 и 1550 нм. После загрузки в программу, необходимо открыть файл, который является основным для этого волокна и провести сравнение в программе. Если программой не были найдены отличия, тогда можно считать, что на линии отсутствуют прослушивающие устройства. При нахождении отличий и появлении новых событий требуется провести проверку. Блок-схема данного алгоритма представлена на рисунке 3.2.

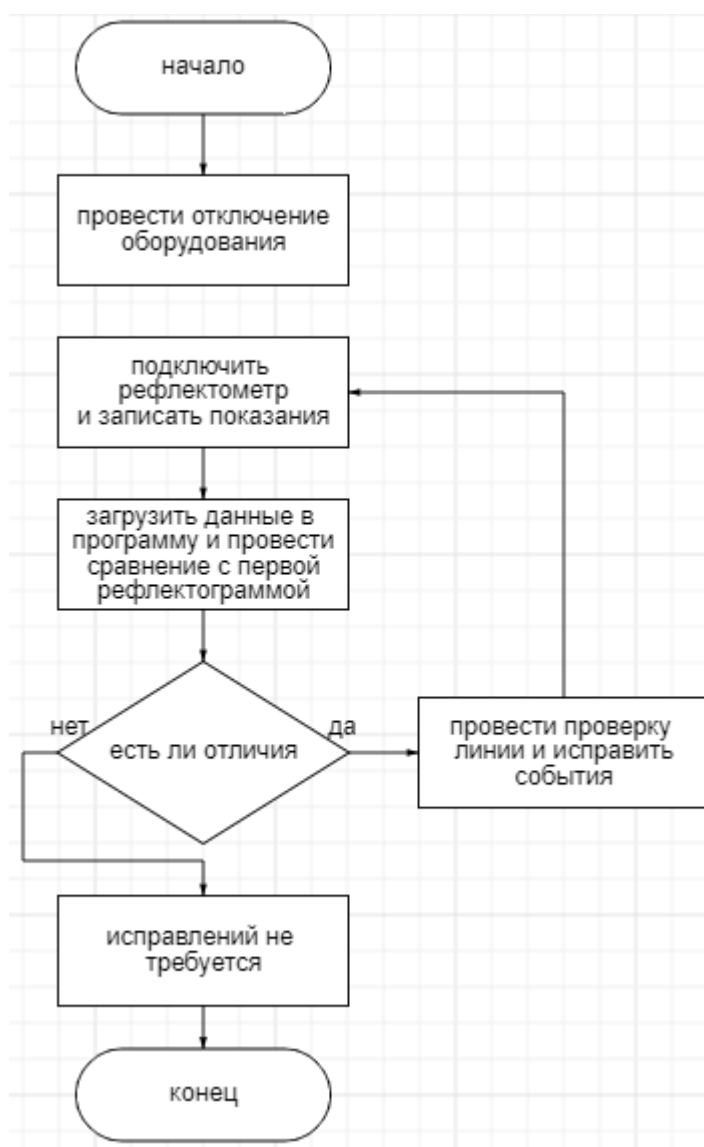


Рисунок 3.2 – Алгоритм работы с программой для сотрудников СБ

Таким образом, предложенный алгоритм действий облегчит жизнь как рядовым сотрудникам, так и сотрудникам службы безопасности.

3.3 Демонстрация работы программного средства

Для демонстрации программного средства было взято несколько, уже построенных оптоволоконных линий различной длины и с разным количеством событий. На рисунке 3.3 видно, несколько кнопок для работы с программой. Клавиша «file 1» позволяет нам открыть требуемы файл, выбрав таковой из нашей системы. После чего нам будет доступна вся информация содержащаяся в файле формата «.SOR». В окне выводится график, который мы парсировали из рефлектограммы, на котором видно расстояние, общее затухание и графически отмеченные точки событий на оптоволоконной линии.

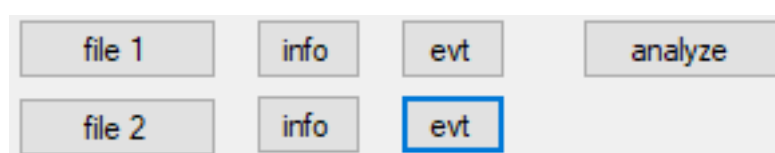


Рисунок 3.3 - Интерфейс

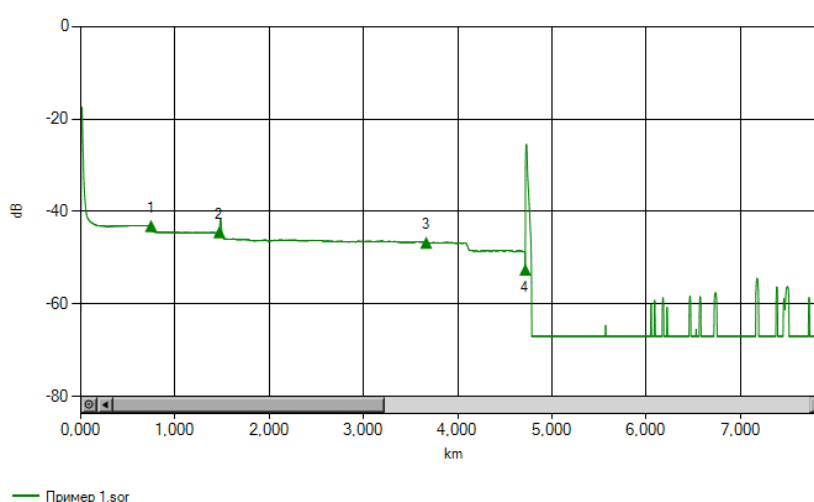


Рисунок 3.4 – График рефлектограммы

При необходимости, на данном графике мы можем приближать интересующий нас участок, как это показано на рисунке 3.5. А для отдаления использовать кнопку в левом нижнем углу графика.

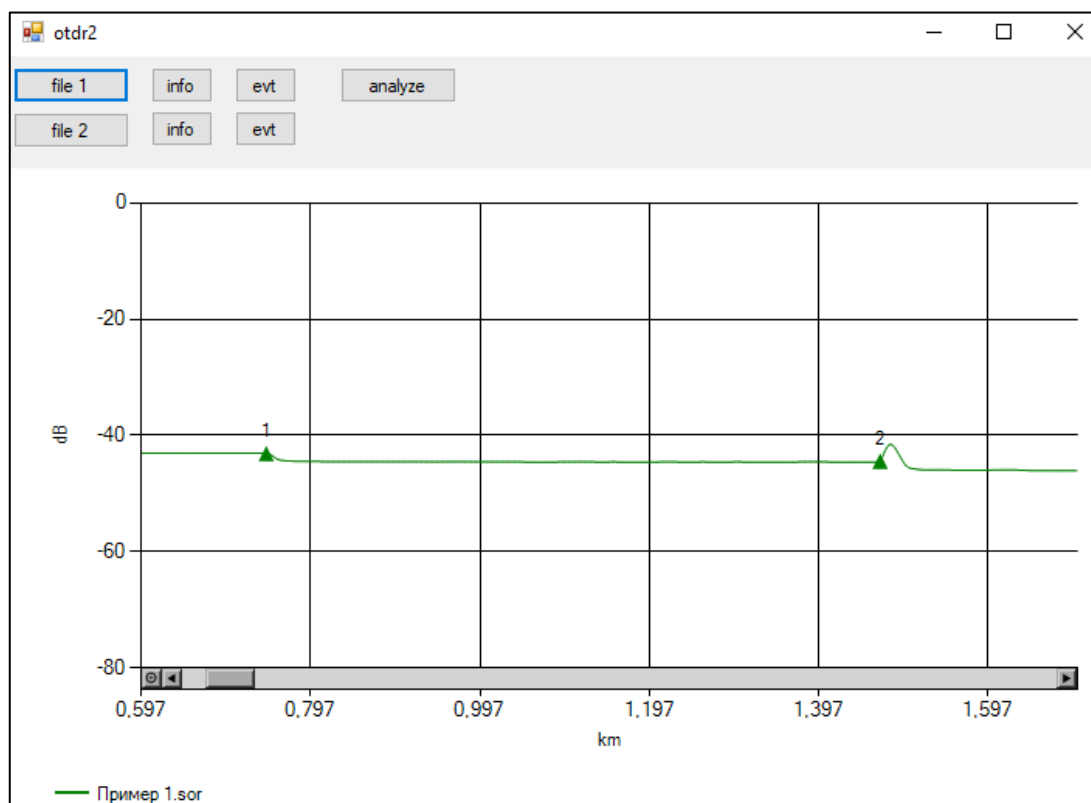


Рисунок 3.5 – Масштабирование графика

При нажатии на кнопку «info» программное средство будет выводить следующую информацию:

- наименование, серийный номер, версия ПО рефлектометра с помощью которого была снята информация;
- длина волны на которой производились замеры;
- общая длина оптоволоконной линии;
- общее затухание отраженного сигнала и так далее.

Пример окна вывода общей информации представлен на рисунке 3.6.

C:\Users\miggil\Desktop>tagan.SOR		C:\Users\miggil\Desktop>tagan.SOR	
Параметр	Значение	Параметр	Значение
filename	C:\Users\miggil\Desktop>tagan.SOR	pulse width	275
General params:		sample spacing	0,00625
cable ID		dpnt count	15670
fiber ID	Fiber0053	index of refraction	1,4677
fiber type	G.652 (standard SMF)	backscattering coefficient	-79,4
wavelength	1310	number of averaging	43242
location A		averaging time	3
location B		range	19,5825
cable code		acquisition range distance	0
build condition	BC (as-built)	front panel offset	0
operator		noise floor level	44351
comments		noise floor scaling factor	1000
Supplier params:		power offset first point	0
supplier name		loss threshold	0,02
OTDR name		reflection threshold	-65,535
OTDR serial number		EOT threshold	5
module name	FTB-7200D-023B-EI	trace type	ST (standard trace)
module serial number	401473	X1	0
software version	1.4.38.138	Y1	0
other		X2	0
Fixed params:		Y2	0
timestamp	946674309	adjusted resolution	1,27662523846835
datetime	Sat Jan 1 00:05:09 2000	adjusted range	20,0047174867991
unit	(m) meter	Data points:	
wavelength	1320,1	number of data points	15670
acquisition offset	0	number of traces	1
acquisition offset distance	0	scaling factor	1
number of pulse width entries	1	max	-63,999
pulse width	275	min	-23,834
sample spacing	0,00625		

Рисунок 3.6 – Пример окна вывода информации о линии

Далее при нажатии кнопки «evt» программное средство выведет окно с информацией о всех событиях затухания на оптоволоконной линии. Выводится таблица событий в которой присутствует следующая информация:

- дистанция от места съёма рефлектограммы до события в километрах;
- общий уровень затухания до данного события в дБ на километр;
- уровень затухания на оптоволоконном событии в дБ;
- уровень отражения в дБ;
- тип затухания, определенный автоматически;
- комментарий к данному событию.

Пример окна вывода информации о событиях затухания представлен на рисунке 3.7.

C:\Users\migg\\Desktop\tagan.SOR								
	#	unajasted distance	distance (km)	slope (dB/km)	splice loss (dB)	reflection loss (dB)	event type	comment
▶	1	0	0,000	0	0	-22,323	multiple	
	2	15000	0,306	0,25	-0,659	0	loss/drop/gain	
	3	131125	2,678	0,321	1,633	0	loss/drop/gain	
	4	200813	4,102	0,309	-1,111	0	loss/drop/gain	
	5	266250	5,438	0,369	0,093	0	loss/drop/gain	
	6	330813	6,757	0,346	0,07	0	loss/drop/gain	
	7	347188	7,092	0,355	0,032	0	loss/drop/gain	
	8	425750	8,696	0,344	0,078	0	loss/drop/gain	
	9	436750	8,921	0,327	1,41	0	loss/drop/gain	
	10	440188	8,991	0,25	-0,888	0	loss/drop/gain	
	11	444625	9,082	0,304	-0,14	0	loss/drop/gain	
	12	538875	11,007	0,375	0,338	0	loss/drop/gain	
	13	542688	11,085	0,328	-0,107	0	loss/drop/gain	
	14	561750	11,474	0,25	0,046	0	loss/drop/gain	
	15	655888	16,888	0,25	0,046	0	loss/drop/gain	

Рисунок 3.7 – Пример окна вывода информации об оптоволоконных событиях

Если необходимо сравнить две рефлектограммы, то в основном окне необходимо открыть сначала основной файл, а после, нажать клавишу «file 2», для открытия рефлектограммы, которую будет сравнивать с основной. Как мы видим на рисунке 3.8, программное средство не открывает новое окно с графиком для отображения второй рефлектограммы, а выводит два графика в одном окне, для облегчения визуального анализа. Производится разделение двух рефлектограмм на разные цвета, какой цвет и за что отвечает можно будет увидеть в левом нижнем углу. Далее так же будет возможность заменять привычные графики на схему маршрута прокладки оптоволоконной линии. Что в свою очередь позволит легче определять визуально точное место волоконно-оптического события затухания. Но тогда придётся более качественно составлять схемы прокладки кабеля, точного указания запасов кабеля.

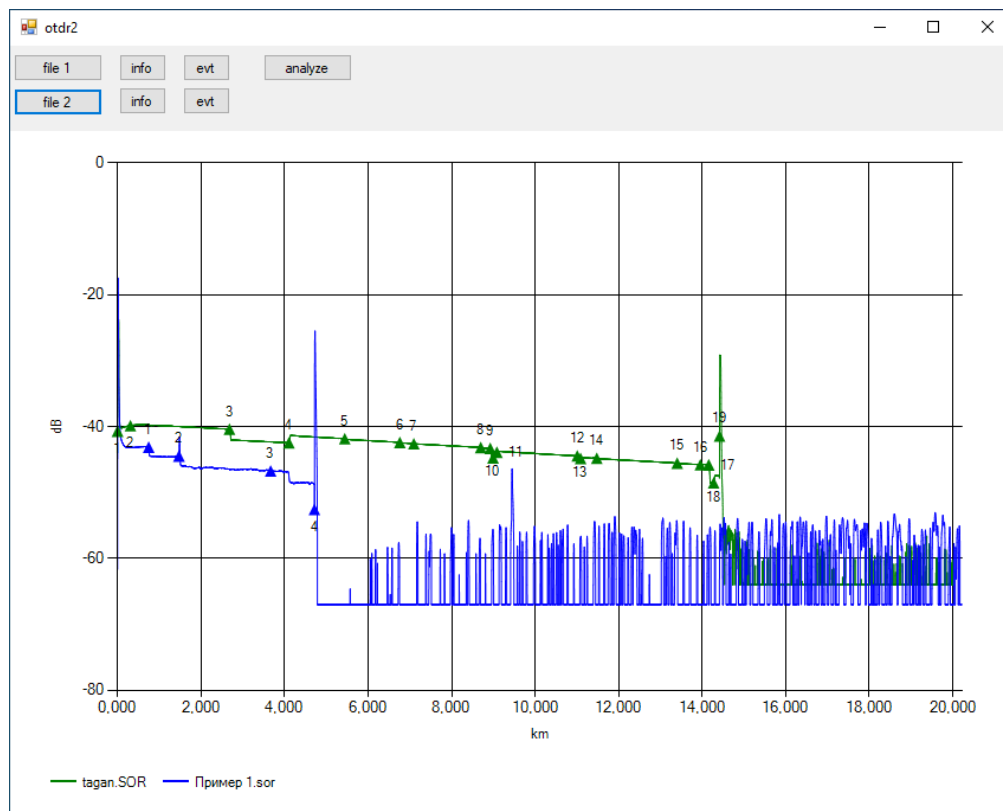


Рисунок 3.8 – Одновременный вывод двух рефлектограмм

Если пользователю необходимо обнаружить прослушивающее устройство или иные отличия на линии со временем, то ему необходимо нажать кнопку «analyze», после чего программа будет сравнивать оптоволоконные события затухания и, если различия имеются, будет производить сравнение с составленной таблицей событий. Ниже на рисунке 3.9 будет представлен вывод программы при нахождении отличий на линии, а на рисунке 3.10 вывод окна при идентичных рефлектограммах.

Form4						
	#	distance (km)	splice loss 1 (dB)	splice loss 2 (dB)	delta (dB)	comment
▶	1	11,007	0	0,338	0,338	Затухание, прич...
	2	14,155	0	2,739	2,739	Критическое за...
	3	14,273	0	-1,158	-1,158	установлен уси...

Рисунок 3.9 – Результат сравнения рефлектограмм с отличиями

Form4						
	#	distance (km)	splice loss 1 (dB)	splice loss 2 (dB)	delta (dB)	comment
▶						замечаний нет

Рисунок 3.10 – Результат сравнения рефлектограмм без явных отличий

В данной главе были обоснован выбор среды разработки и языка программирования. Также был предложен алгоритм работы с данным программным средством. В последнем параграфе была продемонстрирована работа программного средства. Демонстрация проводилась на уже существующих линиях оптоволоконных сетей, для проверки реакции на прослушивающее устройство кабель был искусственно загнут для получения необходимого уровня затухания.

Заключение

В рамках дипломной работы была сформулирована архитектура итогового программного средства, которая в полной мере реализует свой функционал, а именно обнаружение устройств прослушивания в оптоволоконной сети.

Также были описаны алгоритмы разработки и подробно рассмотрено каждое графическое окно, его назначение и функции, выполняющиеся в нём. Приведены зависимости и составы классов серверной части, задана логика воспроизведений функций-обработчиков запросов.

В процессе исследования и разработки дипломной работы была достигнута цель исследования – разработка программного средства для обнаружения устройств прослушивания сетевого трафика в оптоволоконной сети.

Для достижения цели работы были выполнены задачи:

- предоставлены теоретические сведения;
- рассмотрены аналогичные программные продукты;
- разработана программное средство для анализа рефлектограмм.

Перечень использованных информационных ресурсов

1. Белоногов, Г.Г. Автоматизация процессов накопления, поиска и обобщения информации / Г.Г. Белоногов, А.П. Новоселов – Москва : Наука, 2017. – 256 с. – Текст: непосредственный.
2. Беляев, М.А. Основы информатики / М.А. Беляев – Москва : Триумф, 2015. – 200 с. – Текст: непосредственный.
3. Биячуев, Т.А. Безопасность корпоративных сетей / Т.А. Биячуев – Москва : Триумф, 2016. – 670 с. – Текст: непосредственный.
4. Василенко, О.К. Оптоволоконные сети / О.К. Василенко – Москва : Техносфера, 2017. – 288 с. – Текст: непосредственный.
5. Гоппа, В.Д. Введение в алгебраическую теорию информации / В.Д. Гоппа – Москва : ФИЗМАТЛИТ, 2013. – 112 с. – Текст: непосредственный.
6. Саломаа, А. Рефлектограммы / А. Саломаа – Москва : Техносфера, 2014. – 368 с. – Текст: непосредственный.
7. Фомичев, В.Д. Дискретная математика и криптология / В.Д. Фомичев – Москва : Триумф 2016. – 345 с. – Текст: непосредственный.
8. Черкесова, Л.В. Сравнительный анализ систем мгновенного обмена сообщениями / Л.В. Черкесова, О.А. Сафарьян, А.Р. Тосунова, Г.А. Положий – Молодой исследователь Дона №4(25), 2020. – С. 4-7 – Текст: непосредственный.
9. Шварцман, В.О. Теория передачи дискретной информации. Учебник для вузов связи / В.О. Шварцман, М.Г. Емельянов – Москва : Связь, 2016. – 59 с. – Текст: непосредственный.
10. Cherckesova, L. Integration of random numbers generator for creating the gamma of Vernam key in the system of instant message exchange / L. Cherckesova, O. Safaryan, E. Pinevich, I. Reshetnikova, A. Andryushchenko – E3S Web of Conferences, 2021. – С. 4-10 – Текст: непосредственный.
11. Специальные требования и рекомендации по технической защите конфиденциальной информации : Руководящий документ ФСТЭК России

					10.05.01.520000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		51

[утверждено приказом председателя Гостехкомиссии России № 282 от 30 августа 2002 года]. – Москва : Стандартинформ, 2001 – 26 с.

12. Технологии защиты данных. – Текст : электронный // SearchInfo : [сайт]. – 2020. - URL: <https://searchinform.ru/informatsionnaya-bezopasnost/zaschita-informatsii/zaschita-dannykh/tekhnologii-zaschity-dannykh/> (дата обращения 17.11.2021 г.).

13. Двойная аутентификация. – Текст : электронный // CyberVozh : [сайт]. - 2020 г. - URL: <https://book.cyberyozh.com/ru/dvojnaya-autentifikatsiya/> (дата обращения 29.10.2021 г.).

14. Методика определения угроз безопасности информации в информационных системах. – Текст : электронный // Методический документ ФСТЭК России : [сайт]. – 2015. – URL: https://mindstep.ru/wiki/index.php/Методика_определения_угроз_безопасности_информации_в_и_нформационных_системах (дата обращения 26.09.2021 г.).

15. Защита от несанкционированного доступа к информации. Термины и определения : Руководящий документ ФСТЭК России [утверждено решением председателя Гостехкомиссии при Президенте Российской Федерации от 30 марта 1992 года]. – Москва : Кремль, 1992 – 8 с.

Приложение А Техническое задание

СОГЛАСОВАНО

Руководитель проекта

(подпись) Н.В. Болдырихин
«___» _____ 2021 г.

УТВЕРЖДЕНО

И. о. зав. Каф. «КБИС»

(подпись) Д.А. Короченцев
«___» _____ 2021 г.

А.1 Введение

А.1.1 Наименование программы

Наименование программы – «OTDR2».

А.1.2 Область применения

Областями применения данного программного средства являются компьютерные и телефонные оптоволоконные сети.

А.2 Основания для разработки

Разработка ведётся на основании приказа об утверждении тем дипломных работ, утверждённого Донским государственным техническим университетом. Тема разработки: «Программное средство для обнаружения устройств прослушивания сетевого трафика в оптоволоконной сети».

А.3 Назначение разработки

					10.05.01.520000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		53

А.3.1 Функциональное назначение

Функциональным назначением программного средства является просмотр рефлектограмм и анализ событий затухания.

А.3.2 Эксплуатационное назначение

Эксплуатационное назначение: программное средство предназначено для эксплуатации в организациях, деятельность которых связана с монтажом и обслуживанием оптоволоконных линий связи.

А.4 Требования к программе

А.4.1 Требования к функциональным характеристикам

Программное средство должно выводить информацию о рефлектограмме, событиях на линии и сравнивать информацию для обнаружения прослушивающих устройств.

А.4.2 Требования к надёжности

Контроль входной информации осуществляется программой автоматически. Обеспечивается защита от задания неверных значений параметров.

А.4.3 Условия эксплуатации

Условия эксплуатации совпадают с условиями эксплуатации персональных ЭВМ IBM PC и совместимых с ними персональных

					10.05.01.520000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		54

компьютеров. Требуется наличие одного оператора. Программа рассчитана на пользователя, владеющего базовыми навыками работы с десктопными приложениями. Дополнительных требований и ограничений не вводится.

А.4.4 Требования к составу и параметрам технических средств

Для работы программы требуется наличие IBM PC – совместимого персонального компьютера с тактовой частотой процессора не менее 1,5 Гц, объёмом свободной оперативной памяти не менее 512 Мб, монитором и клавиатурой. Необходимое свободное место на жёстком диске – 65 Мб.

А.4.5 Требования к информационной и программной совместимости

Программа должна работать под управлением операционной системы Windows XP и выше. Базовый язык программирования – C++. Среда разработки – Microsoft Visual Studio ver. 17.

А.4.6 Требования к маркировке и упаковке

Требований к маркировке и упаковке не предъявлялось.

А.4.7 Требования к транспортированию и хранению

Требований к транспортированию и хранению не предъявлялось.

А.5 Требования к программной документации

Программная документация должна включать следующие документы:

– документ «Техническое задание» (ГОСТ 19.201-78);

					10.05.01.520000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		55

А.6 Стадии и этапы разработки

1. Системный анализ.
2. Общесистемное проектирование.
3. Подготовка технологических средств.
4. Программная реализация, рабочий проект.
5. Отладка программного средства в статике.
6. Тестовые испытания программного комплекса.

А.7 Порядок контроля и приёмки

Контроль и приёмка разработки осуществляются на основе испытаний контрольно-отладочных примеров. Примеры должны демонстрировать правильность работы используемых в программе структур данных и алгоритмов в различных ситуациях, которые могут возникнуть при выполнении программы. При этом проверяются выполнение всех функций программы и полнота документации.

Разработчик: Андрющенко Александр Александрович

Дата начала разработки: «10» сентября 2021 г.

					10.05.01.520000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		56

Приложение Б Листинг

Листинг Б.1 – Файл otdr2.cpp

```
#include <string>
#define _CRT_SECURE_NO_WARNINGS
#include "stdafx.h"
#include "Form1.h"

using namespace otdr2;

[STAThreadAttribute]
int main(void)
{
    // Включение визуальных эффектов Windows XP до создания каких-
    либо элементов управления
    Application::EnableVisualStyles();
    Application::SetCompatibleTextRenderingDefault(false);

    // Создание главного окна и его запуск
    Application::Run(gcnew Form1());
    return 0;
}
```

Листинг Б.2 – Файл otdr.cpp

```
#include "stdafx.h"
#include <cstdio>
#include <cstring>
#include <ctime>
#include "otdr.h"

// отключаем предупреждения компилятора об устаревших функциях
#define _CRT_NONSTDC_NO_WARNINGS
#pragma warning(disable:4996)

double sol = 299792.458 / 1.0e6; // скорость света

using namespace System;

OTDR::~OTDR()
{
    if(filename != NULL) delete filename;
    if(gen.data != NULL) delete gen.data;
    if(fxd.data != NULL) delete fxd.data;
    if(sup.data != NULL) delete sup.data;
    if(pnt.data != NULL) delete pnt.data;
    if(evt.data != NULL) delete evt.data;
    if(events != NULL) delete events;
```

```

}

// конструктор, читающий файл .sor
OTDR::OTDR(char *fname)
{
    unsigned char buf[4];
    char *p;
    WORD ver, count;
    DWORD size;
    long offset;
    FILE* f;
    struct BlockInfo *bi, tmp;

    filename = strdup(fname);
    f = fopen(filename, "rb");
    fread(buf, 4, 1, f);

    if (memcmp(buf, "Map", 3) != 0) fseek(f, 0, SEEK_SET);

    fread(&ver, sizeof(WORD), 1, f);
    fread(&size, sizeof(DWORD), 1, f); // размер блока "Map"

    fseek(f, 0, SEEK_SET);
    map.data = new char [size];
    fread(map.data, size, 1, f);

    p = map.data;
    if((ver/100)>1) p+=4; // skip 'Map' signature
    map.ver = *(WORD *)p/100.0; p+=sizeof(WORD);
    map.size = *(DWORD *)p; p+=sizeof(DWORD);
    map.offset = 0;
    offset = map.size;

    count = *(WORD*)p; p+=sizeof(WORD);
    count--; // включая Block Map

    while(count--) {
        if (memcmp(p, "GenParams", 9) == 0) bi = &gen;
        else if (memcmp(p, "SupParams", 9) == 0) bi = &sup;
        else if (memcmp(p, "FxdParams", 9) == 0) bi = &fxd;
        else if (memcmp(p, "KeyEvents", 9) == 0) bi = &evt;
        else if (memcmp(p, "DataPts", 7) == 0) bi = &pnt;
        else bi = &tmp;

        while(*p++);

        bi->ver = *((WORD *)p)/100.0; p+=sizeof(WORD);
        bi->size = *(DWORD *)p; p+=sizeof(DWORD);
        bi->offset = offset;
    }
}

```

```

        if (bi==&gen || bi==&sup || bi==&fxd || bi==&evt ||
bi==&pnt) {
            bi->data = new char [bi->size];
            fseek(f, offset, SEEK_SET);
            fread(bi->data, bi->size, 1, f);
        }

        offset += bi->size;
    }

    parseGenParams();
    parseSupParams();
    parseFxdParams();
    parseKeyEvents();
    parseDataPts();

    fclose(f);
    delete map.data;
}

void OTDR::parseGenParams()
{
    char *p;

    p = gen.data + 2; // skip 2xBYTE lang code
    if (gen.ver >= 2) p += 10;

    cableID = p; while(*p++);
    fiberID = p; while(*p++);
    if (gen.ver >= 2) {
        fiberType = *((WORD *)p); p += sizeof(WORD);
    }
    else
        fiberType = 0; // повесить на заглушку для v1
    wavelength = *((WORD *)p); p += sizeof(WORD);
    locationA = p; while(*p++);
    locationB = p; while(*p++);
    cableCode = p; while(*p++);
    buildCondition = p; p += 2;
    Operator = p; while(*p++);
    comments = p;
}

void OTDR::parseSupParams()
{
    char *p;

    p = sup.data;
    if (sup.ver >= 2) p += 10;

    supplierName = p; while(*p++);

```

```

    OTDRName = p; while(*p++);
    OTDRSerialNumber = p; while(*p++);
    moduleName = p; while(*p++);
    moduleSerialNumber = p; while(*p++);
    softwareVersion = p; while(*p++);
    supOther = p;
}

void OTDR::parseFxdParams()
{
WORD *p;

    p = (fxd.ver >= 2) ? (WORD *) (fxd.data + 10) : (WORD
*) fxd.data;

    time = *(DWORD *)p; p+=2;
    units = (*p == 0x6d6b) ? 1 : (*p == 0x746d) ? 2 : (*p ==
0x7466) ? 3 :
        (*p == 0x666b) ? 4 : (*p == 0x696d) ? 5 : 0; p++;
    fxdWavelength = *p * 0.1; p++;
    acqOffset = *(DWORD *)p; p+=2;
    acqOffsetDistance = *(DWORD *)p; p+=2;
    numPulseWidthEntries = *p; p++;
    pulsewidth = *p; p++;
    sampleSpacing = *(DWORD *)p * 1e-8; p+=2;
    dpntCnt = *(DWORD *)p; p+=2;
    indexOfRefraction = *(DWORD *)p * 1e-5; p+=2;
    backscatteringCoefficient = *p * -0.1; p++;
    numberOfAvg = *(DWORD *)p; p+=2;
    avgTime = *p * 0.1; p++;
    range = *(DWORD *)p * 2e-5; p+=2; // km (x2)
    acqRangeDistance = *(DWORD *)p; p+=2;
    frontPanelOfst = *(DWORD *)p; p+=2;
    noiseFloorLevel = *p; p++;
    noiseFloorScaling = *p; p++;
    powerOffsetFP = *p; p++;
    lossThreshold = *p * 0.001; p++;
    reflectionThreshold = *p * -0.001; p++;
    EOTThreshold = *p * 0.001; p++;
    TraceType = (*p == 0x5453) ? 1 : (*p == 0x5452) ? 2 : (*p ==
0x5444) ? 3 :
        (*p == 0x4652) ? 4 : 0; p++;
    X1 = *(DWORD *)p; p+=2;
    Y1 = *(DWORD *)p; p+=2;
    X2 = *(DWORD *)p; p+=2;
    Y2 = *(DWORD *)p;

    double dx = sampleSpacing * sol / indexOfRefraction;
    adjResolution = dx * 1000.0;
    adjRange = dx * dpntCnt;
}

```

					10.05.01.520000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		60

```

void OTDR::parseKeyEvents()
{
    char *p;
    double dx = sol / (indexOfRefraction * 10000.0);

    p = evt.data;
    if (evt.ver >= 2) p+=10;

    eventCount = *(WORD *)p; p+=2;
    events = new struct EventEntry [eventCount];

    for(int i=0; i<eventCount; i++) {
        p += sizeof(WORD);
        events[i].unadjustedDistance = *(DWORD *)p;
        p+=sizeof(DWORD);
        events[i].distance = events[i].unadjustedDistance * dx;
        events[i].slope = *(short *)p/1000.0; p+=sizeof(WORD);
        events[i].spliceLoss = *(short *)p/1000.0;
        p+=sizeof(WORD);
        events[i].reflectionLoss = *(short *)p/1000.0;
        p+=sizeof(DWORD);
        events[i].eventType = p; p+= 28;
        events[i].comment = p; while(*p++);
    }
}

void OTDR::parseDataPts()
{
    WORD *p;

    pntMax = 0;
    pntMin = 65535;

    p = (pnt.ver >= 2) ? (WORD *) (pnt.data + 8) : (WORD *)pnt.data;

    numDataPoints = *(DWORD *)p; p+=2;
    numOfTraces = *p; p+=3;
    scalingFactor = *p/1000.0; p++;
    dataPoints = p;

    for(int i=0; i<numDataPoints; i++, p++) {
        if((double)*p>pntMax) pntMax = (double)*p;
        if((double)*p<pntMin) pntMin = (double)*p;
    }

    pntMax *= scalingFactor * -0.001;
    pntMin *= scalingFactor * -0.001;
}

```

```

String^ OTDR::strUnit()
{
    String^ s;

    switch(units) {
        case 1: return s = gcnew String("(km) kilometer");
        case 2: return s = gcnew String("(m) meter");
        case 3: return s = gcnew String("(ft) foot");
        case 4: return s = gcnew String("(kft) kilo-foot");
        case 5: return s = gcnew String("(mi) mile");
    }
    return s = gcnew String("unknown");
}

String^ OTDR::strFiberType()
{
    String^ s;

    switch(fiberType) {
        case 651: return s = gcnew String("G.651 (50um core multimode)");
        case 652: return s = gcnew String("G.652 (standard SMF)");
        case 653: return s = gcnew String("G.653 (dispersion-shifted fiber)");
        case 654: return s = gcnew String("G.654 (1550nm loss-minimized fiber)");
        case 655: return s = gcnew String("G.655 (nonzero dispersion-shifted fiber)");
    }
    return s = gcnew String("(unknown)");
}

String^ OTDR::strBuildCondition()
{
    String^ s;

    if (buildCondition[0]=='B' && buildCondition[1]=='C') return s = gcnew String("BC (as-built)");
    if (buildCondition[0]=='C' && buildCondition[1]=='C') return s = gcnew String("CC (as-current)");
    if (buildCondition[0]=='R' && buildCondition[1]=='C') return s = gcnew String("RC (as-repaired)");
    if (buildCondition[0]=='O' && buildCondition[1]=='T') return s = gcnew String("OT (other)");

    return s = gcnew String("unknown");
}

String^ OTDR::shortFilename()
{
    String^ s = gcnew String(filename);
}

```

```

String^ shortFileName = System::IO::Path::GetFileName(s);

return shortFileName;
}

String^ OTDR::strTraceType()
{
    String^ s;

    switch(TraceType) {
        case 1: return s = gcnew String("ST (standard trace)");
        case 2: return s = gcnew String("RT (reverse trace)");
        case 3: return s = gcnew String("ST (difference trace)");
        case 4: return s = gcnew String("RF (reference)");
    }
    return s = gcnew String("(unknown)");
}

String^ EventEntry::strComment()
{
    String^ s;
    return s = gcnew System::String(comment);
}

String^ EventEntry::strType()
{
    /*
    Code for the event is a 6-byte string:
        Byte 1: 0 = nonreflective, 1 = reflective, 2 = saturated
reflective
        Byte 2: A = added by user, M = moved by user, E = end of
fibre, F = found by software, O = out of range, D = modified end
of fibre.
        Remaining bytes are the Landmark number if used - 9s
otherwise
        Loss measurement technique - 2P for two point, LS for least
squares, OT for other
    */
    String^ s;

    if (eventType[1] == 'E' || eventType[1] == 'D') return s =
gcnew String("end of fibre");
    if (eventType[1] == 'O') return s = gcnew String("out of
range");

    switch (eventType[0]) {
        case '0': return s = gcnew String("loss/drop/gain");
        case '1': return s = gcnew String("reflection");
        case '2': return s = gcnew String("multiple");
    }
}

```



```

        return s = gcnew String("unknown");
    }

```

Листинг Б.3 – Файд forms.cpp

```

#include "StdAfx.h"
#include "otdr.h"
#include "Form1.h"
#include "Form2.h"
#include "Form3.h"
#include "Form4.h"

namespace otdr2 {
// загрузка графика 1
System::Void Form1::button3_Click(System::Object^ sender,
System::EventArgs^ e) {
    using namespace
System::Windows::Forms::DataVisualization::Charting;
    using namespace System::Collections::Generic;
    using namespace System::Runtime::InteropServices;

    openFileDialog1->ShowDialog();

    sor1 = new
OTDR((char*)Marshal::StringToHGlobalAnsi(openFileDialog1-
>FileName).ToPointer());
    chart1->Series[0]->Name = sor1->shortFilename();

    // построение графика
    Dictionary <double, double> f1 = gcnew Dictionary<double,
double>();
    Dictionary <int, double> evtIdx = gcnew Dictionary<int,
double>();

    double sol = 299792.458 / 1.0e6; // скорость света
    double dx = sor1->sampleSpacing * sol / sor1-
>indexOfRefraction, pos = 0;

    for (int i = 0, evt = 0; i < sor1->dpntCnt; i++, pos += dx)
    {
        if (pos >= sor1->events[evt].distance && evt < sor1-
>eventCount) evtIdx.Add(evt++, i);
        f1.Add(pos, (double)sor1->dataPoints[i] * -0.001 * sor1-
>scalingFactor);
    }

    chart1->Series[0]->Points->DataBindXY(f1.Keys, f1.Values);

    for (int i = 0; i < sor1->eventCount; i++) {
        chart1->Series[0]->Points[evtIdx[i]]->MarkerStyle =
MarkerStyle::Triangle;
        chart1->Series[0]->Points[evtIdx[i]]->MarkerSize = 10;
    }
}

```

```

        chart1->Series[0]->Points[evtIdx[i]]->Label = (i +
1).ToString();
    }

    this->chart1->Visible = true;
}

// информация по графику 1
System::Void Form1::button2_Click(System::Object^ sender,
System::EventArgs^ e) {

    if (sor1 == NULL) return;

    Form2^ F2 = gcnew Form2;

    F2->Text = "Information " + sor1->shortFilename();
    F2->printOTDR(sor1);
    F2->Show();
}

// события графика 1
System::Void Form1::button4_Click(System::Object^ sender,
System::EventArgs^ e) {

    if (sor1 == NULL) return;

    Form3^ F3 = gcnew Form3;

    F3->Text = "Events " + sor1->shortFilename();
    F3->showOTDRevents(sor1);
    F3->Show();
}

// загрузка графика 2
System::Void Form1::button1_Click_1(System::Object^ sender,
System::EventArgs^ e) {
    using namespace
System::Windows::Forms::DataVisualization::Charting;
    using namespace System::Collections::Generic;
    using namespace System::Runtime::InteropServices;

    String^ shortFileName;

    openFileDialog1->ShowDialog();

    sor2 = new
OTDR((char*)Marshal::StringToHGlobalAnsi(openFileDialog1-
>FileName).ToPointer());
    chart1->Series[1]->Name = sor2->shortFilename();

    // построение графика

```

					10.05.01.520000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		65

```

        Dictionary <double, double> f2 = gcnew Dictionary<double,
double>();
        Dictionary <int, double> evtIdx = gcnew Dictionary<int,
double>();

        double sol = 299792.458 / 1.0e6; // скорость света
        double dx = sor2->sampleSpacing * sol / sor2-
>indexOfRefraction, pos = 0;

        for (int i = 0, evt = 0; i < sor2->dpntCnt; i++, pos += dx)
        {
            if (pos >= sor2->events[evt].distance && evt < sor2-
>eventCount) evtIdx.Add(evt++, i);
            f2.Add(pos, (double)sor2->dataPoints[i] * -0.001 * sor2-
>scalingFactor);
        }

        chart1->Series[1]->Points->DataBindXY(f2.Keys, f2.Values);

        for (int i = 0; i < sor2->eventCount; i++) {
            chart1->Series[1]->Points[evtIdx[i]]->MarkerStyle =
MarkerStyle::Triangle;
            chart1->Series[1]->Points[evtIdx[i]]->MarkerSize = 10;
            chart1->Series[1]->Points[evtIdx[i]]->Label = (i +
1).ToString();
        }

        this->chart1->Visible = true;
    }

// информация по графику 2
System::Void Form1::button5_Click(System::Object^ sender,
System::EventArgs^ e) {
    if (sor2 == NULL) return;

    Form2^ F2 = gcnew Form2;

    F2->Text = "Information " + sor2->shortFilename();
    F2->printOTDR(sor2);
    F2->Show();
}

// события графика 2
System::Void Form1::button6_Click(System::Object^ sender,
System::EventArgs^ e) {

    if (sor2 == NULL) return;

    Form3^ F3 = gcnew Form3;

    F3->Text = "Events " + sor2->shortFilename();

```

```

        F3->showOTDRevents(sor2);
        F3->Show();
    }

    // анализ двух файлов
    System::Void Form1::button7_Click(System::Object^ sender,
    System::EventArgs^ e) {
        if (sor1 == NULL) return;
        if (sor2 == NULL) return;

        Form4^ F4 = gcnew Form4;

        F4->Text = "Analyze " + sor1->shortFilename() + " & " + sor2->
shortFilename();
        F4->Analyze(sor1, sor2);
        F4->Show();
    }

    // заполнение окна данными из файла OTDR
    void Form2::printOTDR(OTDR* sor)
    {
        using namespace System::Runtime::InteropServices;
        using namespace std;

        int i = 0;
        String^ s;

        dataGridView1->Rows->Add(57);

        dataGridView1->Rows[i]->Cells[0]->Value = "filename";
        dataGridView1->Rows[i++]->Cells[1]->Value = s = gcnew
String(sor->filename);

        dataGridView1->Rows[i]->DefaultCellStyle->BackColor =
Color::Cyan;
        dataGridView1->Rows[i]->Cells[0]->Value = "General params:";
        dataGridView1->Rows[i++]->Cells[1]->Value = "";

        dataGridView1->Rows[i]->Cells[0]->Value = "cable ID";
        dataGridView1->Rows[i++]->Cells[1]->Value = s = gcnew
String(sor->cableID);

        dataGridView1->Rows[i]->Cells[0]->Value = "fiber ID";
        dataGridView1->Rows[i++]->Cells[1]->Value = s = gcnew
String(sor->fiberID);

        dataGridView1->Rows[i]->Cells[0]->Value = "fiber type";
        dataGridView1->Rows[i++]->Cells[1]->Value = sor->
strFiberType();

        dataGridView1->Rows[i]->Cells[0]->Value = "wavelength";

```

					10.05.01.520000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		67

```

        dataGridView1->Rows[i++]->Cells[1]->Value = sor->wavelength;

        dataGridView1->Rows[i]->Cells[0]->Value = "location A";
        dataGridView1->Rows[i++]->Cells[1]->Value = s = gcnew
String(sor->locationA);

        dataGridView1->Rows[i]->Cells[0]->Value = "location B";
        dataGridView1->Rows[i++]->Cells[1]->Value = s = gcnew
String(sor->locationB);

        dataGridView1->Rows[i]->Cells[0]->Value = "cable code";
        dataGridView1->Rows[i++]->Cells[1]->Value = s = gcnew
String(sor->cableCode);

        dataGridView1->Rows[i]->Cells[0]->Value = "build condition";
        dataGridView1->Rows[i++]->Cells[1]->Value = sor-
>strBuildCondition();

        dataGridView1->Rows[i]->Cells[0]->Value = "operator";
        dataGridView1->Rows[i++]->Cells[1]->Value = s = gcnew
String(sor->Operator);

        dataGridView1->Rows[i]->Cells[0]->Value = "comments";
        dataGridView1->Rows[i++]->Cells[1]->Value = s = gcnew
String(sor->comments);

        dataGridView1->Rows[i]->DefaultCellStyle->BackColor =
Color::Cyan;
        dataGridView1->Rows[i]->Cells[0]->Value = "Supplier
params:";
        dataGridView1->Rows[i++]->Cells[1]->Value = "";

        dataGridView1->Rows[i]->Cells[0]->Value = "supplier name";
        dataGridView1->Rows[i++]->Cells[1]->Value = s = gcnew
String(sor->supplierName);

        dataGridView1->Rows[i]->Cells[0]->Value = "OTDR name";
        dataGridView1->Rows[i++]->Cells[1]->Value = s = gcnew
String(sor->OTDRName);

        dataGridView1->Rows[i]->Cells[0]->Value = "OTDR serial
number";
        dataGridView1->Rows[i++]->Cells[1]->Value = s = gcnew
String(sor->OTDRSerialNumber);

        dataGridView1->Rows[i]->Cells[0]->Value = "module name";
        dataGridView1->Rows[i++]->Cells[1]->Value = s = gcnew
String(sor->moduleName);

```

```

        dataGridView1->Rows[i]->Cells[0]->Value = "module serial
number";
        dataGridView1->Rows[i++]->Cells[1]->Value = s = gcnew
String(sor->moduleSerialNumber);

        dataGridView1->Rows[i]->Cells[0]->Value = "software
version";
        dataGridView1->Rows[i++]->Cells[1]->Value = s = gcnew
String(sor->softwareVersion);

        dataGridView1->Rows[i]->Cells[0]->Value = "other";
        dataGridView1->Rows[i++]->Cells[1]->Value = s = gcnew
String(sor->supOther);

        dataGridView1->Rows[i]->DefaultCellStyle->BackColor =
Color::Cyan;
        dataGridView1->Rows[i]->Cells[0]->Value = "Fixed params:";
        dataGridView1->Rows[i++]->Cells[1]->Value = "";

        dataGridView1->Rows[i]->Cells[0]->Value = "timestamp";
        dataGridView1->Rows[i++]->Cells[1]->Value = sor->time;

        dataGridView1->Rows[i]->Cells[0]->Value = "datetime";
        dataGridView1->Rows[i++]->Cells[1]->Value = s = gcnew
String(std::ctime(&sor->time));

        dataGridView1->Rows[i]->Cells[0]->Value = "unit";
        dataGridView1->Rows[i++]->Cells[1]->Value = sor->strUnit();

        dataGridView1->Rows[i]->Cells[0]->Value = "wavelength";
        dataGridView1->Rows[i++]->Cells[1]->Value = sor-
>fxdWavelength;

        dataGridView1->Rows[i]->Cells[0]->Value = "acquisistion
offset";
        dataGridView1->Rows[i++]->Cells[1]->Value = sor->acqOffset;

        dataGridView1->Rows[i]->Cells[0]->Value = "acquisistion
offset distance";
        dataGridView1->Rows[i++]->Cells[1]->Value = sor-
>acqOffsetDistance;

        dataGridView1->Rows[i]->Cells[0]->Value = "number of pulse
width entries";
        dataGridView1->Rows[i++]->Cells[1]->Value = sor-
>numPulseWidthEntries;

        dataGridView1->Rows[i]->Cells[0]->Value = "pulse width";
        dataGridView1->Rows[i++]->Cells[1]->Value = sor->pulsewidth;

        dataGridView1->Rows[i]->Cells[0]->Value = "sample spacing";

```

```

        dataGridView1->Rows[i++]>Cells[1]>Value      =      sor-
>sampleSpacing;

        dataGridView1->Rows[i]>Cells[0]>Value = "dpnt count";
        dataGridView1->Rows[i++]>Cells[1]>Value = sor->dpntCnt;

        dataGridView1->Rows[i]>Cells[0]>Value      =      "index      of
refraction";
        dataGridView1->Rows[i++]>Cells[1]>Value      =      sor-
>indexOfRefraction;

        dataGridView1->Rows[i]>Cells[0]>Value      =      "backscattering
coefficient";
        dataGridView1->Rows[i++]>Cells[1]>Value      =      sor-
>backscatteringCoefficient;

        dataGridView1->Rows[i]>Cells[0]>Value      =      "number      of
averaging";
        dataGridView1->Rows[i++]>Cells[1]>Value      =      sor-
>numberOfAvg;

        dataGridView1->Rows[i]>Cells[0]>Value = "averaging time";
        dataGridView1->Rows[i++]>Cells[1]>Value = sor->avgTime;

        dataGridView1->Rows[i]>Cells[0]>Value = "range";
        dataGridView1->Rows[i++]>Cells[1]>Value = sor->range;

        dataGridView1->Rows[i]>Cells[0]>Value = "acquisition range
distance";
        dataGridView1->Rows[i++]>Cells[1]>Value      =      sor-
>acqRangeDistance;

        dataGridView1->Rows[i]>Cells[0]>Value      =      "front      panel
offset";
        dataGridView1->Rows[i++]>Cells[1]>Value      =      sor-
>frontPanelOfst;

        dataGridView1->Rows[i]>Cells[0]>Value      =      "noise      floor
level";
        dataGridView1->Rows[i++]>Cells[1]>Value      =      sor-
>noiseFloorLevel;

        dataGridView1->Rows[i]>Cells[0]>Value      =      "noise      floor
scaling factor";
        dataGridView1->Rows[i++]>Cells[1]>Value      =      sor-
>noiseFloorScaling;

        dataGridView1->Rows[i]>Cells[0]>Value = "power offset first
point";
        dataGridView1->Rows[i++]>Cells[1]>Value      =      sor-
>powerOffsetFP;

```

					10.05.01.520000.000 ПЗ	Лист
						70
Изм.	Лист	№ докум.	Подпись	Дата		

```

        dataGridView1->Rows[i]->Cells[0]->Value = "loss threshold";
        dataGridView1->Rows[i++]->Cells[1]->Value = sor->lossThreshold;

        dataGridView1->Rows[i]->Cells[0]->Value = "reflection
threshold";
        dataGridView1->Rows[i++]->Cells[1]->Value = sor->reflectionThreshold;

        dataGridView1->Rows[i]->Cells[0]->Value = "EOT threshold";
        dataGridView1->Rows[i++]->Cells[1]->Value = sor->EOTThreshold;

        dataGridView1->Rows[i]->Cells[0]->Value = "trace type";
        dataGridView1->Rows[i++]->Cells[1]->Value = sor->strTraceType();

        dataGridView1->Rows[i]->Cells[0]->Value = "X1";
        dataGridView1->Rows[i++]->Cells[1]->Value = sor->X1;
        dataGridView1->Rows[i]->Cells[0]->Value = "Y1";
        dataGridView1->Rows[i++]->Cells[1]->Value = sor->Y1;
        dataGridView1->Rows[i]->Cells[0]->Value = "X2";
        dataGridView1->Rows[i++]->Cells[1]->Value = sor->X2;
        dataGridView1->Rows[i]->Cells[0]->Value = "Y2";
        dataGridView1->Rows[i++]->Cells[1]->Value = sor->Y2;

        dataGridView1->Rows[i]->Cells[0]->Value = "adjusted
resolution";
        dataGridView1->Rows[i++]->Cells[1]->Value = sor->adjResolution;

        dataGridView1->Rows[i]->Cells[0]->Value = "adjusted range";
        dataGridView1->Rows[i++]->Cells[1]->Value = sor->adjRange;

        dataGridView1->Rows[i]->DefaultCellStyle->BackColor =
Color::Cyan;
        dataGridView1->Rows[i]->Cells[0]->Value = "Data points:";
        dataGridView1->Rows[i++]->Cells[1]->Value = "";

        dataGridView1->Rows[i]->Cells[0]->Value = "number of data
points";
        dataGridView1->Rows[i++]->Cells[1]->Value = sor->numDataPoints;

        dataGridView1->Rows[i]->Cells[0]->Value = "number of traces";
        dataGridView1->Rows[i++]->Cells[1]->Value = sor->numOfTraces;

        dataGridView1->Rows[i]->Cells[0]->Value = "scaling factor";

```



```

        dataGridView1->Rows[i++]>Cells[1]>Value = sor->scalingFactor;

        dataGridView1->Rows[i]>Cells[0]>Value = "max";
        dataGridView1->Rows[i++]>Cells[1]>Value = sor->pntMax;

        dataGridView1->Rows[i]>Cells[0]>Value = "min";
        dataGridView1->Rows[i++]>Cells[1]>Value = sor->pntMin;
    }

    // заполнение окна событиями из файла OTDR
    void Form3::showOTDRevents(OTDR* sor)
    {
        dataGridView1->Rows->Add(sor->eventCount);

        for (int i = 0; i < sor->eventCount; i++) {
            dataGridView1->Rows[i]>Cells[0]>Value = i + 1;
            dataGridView1->Rows[i]>Cells[1]>Value = sor->events[i].unajustedDistance;
            dataGridView1->Rows[i]>Cells[2]>Value = sor->events[i].distance.ToString("0.000");
            dataGridView1->Rows[i]>Cells[3]>Value = sor->events[i].slope;
            dataGridView1->Rows[i]>Cells[4]>Value = sor->events[i].spliceLoss;
            dataGridView1->Rows[i]>Cells[5]>Value = sor->events[i].reflectionLoss;
            dataGridView1->Rows[i]>Cells[6]>Value = sor->events[i].strType();
            dataGridView1->Rows[i]>Cells[7]>Value = sor->events[i].strComment();
        }
    }

    void Form4::Analyze(OTDR* sor1, OTDR* sor2)
    {
        int linenummer, eventnumber;
        double loss1, loss2, delta;

        for (linenummer = 0, eventnumber = 0; eventnumber < sor2->eventCount; eventnumber++) {

            loss1 = loss2 = delta = 0.0;

            // ищем событие в первом файле примерно в этом месте
            for (int j = 0; j < sor1->eventCount; j++) {
                double n = sor2->events[eventnumber].distance - sor1->events[j].distance;
                if ((n * n) < 0.0001) {
                    loss1 = sor1->events[j].spliceLoss;
                    break;
                }
            }
        }
    }

```

```

    }
}

loss2 = sor2->events[eventnumber].spliceLoss;
delta = loss2 - loss1;

if (delta > -0.8 && delta < 0.1) continue; // фантомное
усиление, погрешность прибора или незначительное затухание

dataGridView1->Rows->Add();
dataGridView1->Rows[linenumber]->Cells[0]->Value =
linenumber + 1;
dataGridView1->Rows[linenumber]->Cells[1]->Value =
sor2->events[eventnumber].distance.ToString("0.000");
dataGridView1->Rows[linenumber]->Cells[2]->Value =
loss1;
dataGridView1->Rows[linenumber]->Cells[3]->Value =
loss2;
dataGridView1->Rows[linenumber]->Cells[4]->Value =
delta;

if (delta <= -0.8)
    dataGridView1->Rows[linenumber]->Cells[5]->Value =
"установлен усилитель сигнала или возможно фантомное усиление из-
за различий во врезанном кабеле";

if (delta >= 0.1 && delta <= 1.5)
    dataGridView1->Rows[linenumber]->Cells[5]->Value =
"Затухание, причиной может быть незначительный изгиб кабеля или
сварка плохого качества";

if (delta > 1.5 && delta <= 6.0)
    dataGridView1->Rows[linenumber]->Cells[5]->Value =
"Критическое затухание, причина может быть в плохой сварке или
критическим изгибом. Возможна установка прослушивающего
устройства";

linenumber++;
}
if (dataGridView1->RowCount == 0) {
    dataGridView1->Rows->Add();
    dataGridView1->Rows[0]->Cells[5]->Value = "замечаний
нет";
}
}
}

```

Приложение А Техническое задание

СОГЛАСОВАНО

Руководитель проекта

Н.В. Болдырихин

(подпись)

« » 2021 г.

УТВЕРЖДЕНО

И. о. зав. Каф. «КБИС»

Д.А. Короченцев

(подпись)

« » 2021 г.

А.1 Введение

А.1.1 Наименование программы

Наименование программы – «OTDR2».

А.1.2 Область применения

Областями применения данного программного средства являются компьютерные и телефонные оптоволоконные сети.

А.2 Основания для разработки

Разработка ведётся на основании приказа об утверждении тем дипломных работ, утверждённого Донским государственным техническим университетом. Тема разработки: «Программное средство для обнаружения устройств прослушивания сетевого трафика в оптоволоконной сети».

А.3 Назначение разработки

Лист

53

10.05.01.520000.000 ПЗ



SHOT ON REDMI 9
AI QUAD CAMERA

Изм.	Лист	№ докум.	Подпись	Дата