

ТОРАЗ ТР

(Трассировка)

643.17480174.00001-01 31-14

ОПИСАНИЕ ПРОГРАММЫ

<i>Инв. №</i>	<i>Подпись и дата</i>	<i>Взам. инв.</i>	<i>Инв. №</i>	<i>Подпись и дата</i>

Москва 2023

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

БД – база данных

КА – коммутационный автомат

ОД – остовное дерево графа

ФЦ – фундаментальный простой цикл

SCADA – Supervisory Control And Data Acquisition (диспетчерское управление и сбор данных)

АННОТАЦИЯ

В данном программном документе приведено описание приложения TOPAZ TR (Трассировка). Документ содержит сведения о логической структуре и функционировании данного приложения.

СОДЕРЖАНИЕ

1.	ОБЩИЕ СВЕДЕНИЯ	5
2.	ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ	5
3.	ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ	5
3.1.	Алгоритмы программы	5
3.1.1.	До нормальных токоразделов	7
3.1.2.	От текущих токоразделов	7
3.1.3.	Топологический остров	7
3.1.4.	В сторону питания	7
3.1.5.	В сторону потребления	7
3.1.6.	До ближайших КА	7
3.1.7.	Поиск закольцованных участков.....	8
3.2.	Используемые методы	8
3.3.	Структура программы с описанием функций составных частей и связи между ними	8
3.4.	Связи программы с другими программами	8
4.	ИСПОЛЬЗУЕМЫ ТЕХНИЧЕСКИЕ СРЕДСТВА.....	8
5.	ВЫЗОВ И ЗАГРУЗКА	9
5.1.	Способ вызова программы	9
5.2.	Входные точки в программу	9
6.	ВХОДНЫЕ ДАННЫЕ	9
6.1.	Характер, организация и предварительная подготовка входных данных	9
7.	ВЫХОДНЫЕ ДАННЫЕ	9
7.1.	Характер, организация и предварительная подготовка выходных данных.....	9

1. ОБЩИЕ СВЕДЕНИЯ

1.1. Обозначение и наименование программы

Наименование программы – ТОРАЗ TR (Трассировка).

1.1. Программное обеспечение, необходимое для функционирования программы

Системные программные средства, используемые программой – операционные системы реального времени на основе ядра LINUX.

1.2. Языки программирования, на которых написана программа

Программа написана на языке C#.

2. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

Приложение осуществляет трассировку по распределенной электрической сети. Под трассировкой понимается запускаемый пользователем поиск путей в топологии сети между различными видами оборудования, отвечающих определенным условиям. Каждый тип трассировки привязывается к определенному начальному оборудованию, относительно которого осуществляется поиск целевых элементов сети.

Типы трассировок, осуществляемых приложением:

- трассировка до источника;
- трассировка до потребителей;
- трассировка до ближайших КА;
- определение наличия и индикация на схеме циклически замкнутых участков сети;
- расчет «топологических островов» на графе сети;
- трассировка до нормальных токоразделов;
- трассировка до текущих токоразделов.

3. ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

3.1. Алгоритмы программы

Перед началом работы большинства из описанных ниже алгоритмов производится редукция графа топологии сети: из него удаляются все двухтерминальные элементы, не несущие дополнительной топологической информации, используемой в рамках данного алгоритма трассировки. Изначальное место расположения удаленных вершин и ребер сети при этом сохраняется в специальных словарях и может быть восстановлено после завершения основного этапа поиска. Такой подход позволяет снизить размерность задачи, выраженную в количестве вершин и ребер графа топологии, с нескольких сот тысяч до нескольких десятков тысяч и на порядок ускорить поиск.

Трассировка до нормальных и текущих токоразделов требует расчета кратчайших путей между вершинами на графе топологии сети. В текущей реализации он осуществляется с помощью алгоритма Дейкстры. Трассировка топологического острова осуществляется на основе алгоритма обхода графа в глубину (Depth-first search) с запоминанием посещенных узлов. Трассировка до ближайших КА осуществляется на основе обхода графа в глубину с запоминанием обратного пути до начального этапа поиска.

При расчете трассировок, требующих построения всех альтернативных путей до целевых элементов и при этом потенциально покрывающих в ходе поиска крупные фрагменты сети, вместо классического алгоритма обхода графа в глубину с запоминанием обратного пути используется альтернативный алгоритм на основе построения остовного дерева графа (далее – ОД), общая схема которого выглядит следующим образом:

- 1) Производится построение произвольного ОД для той компоненты связности графа топологии сети, к которой относится начальный элемент трассировки. Для построения ОД используется упрощенный алгоритм Прима, не учитывающий вес ребер.
- 2) Производится поиск путей от начального элемента до целевых на полученном ОД: согласно определению ОД, если существует путь между двумя вершинами графа, то существует также путь между данными вершинами на любом ОД данного графа.
- 3) Каждое ребро начального графа, удаленное при формировании ОД, образует ровно один фундаментальный простой цикл (далее – ФЦ) в исходном графе. Пользуясь этим свойством, а также теоремой о разложении произвольного простого цикла в графе по базису ФЦ с использованием операции циклической суммы подграфов, последний этап построения трассировки (поиск альтернативных путей до целевых узлов) можно осуществить следующим образом:
 - Найти все ФЦ графа, последовательно возвращая и вновь удаляя из него удаленные ранее ребра и выполняя поиск в глубину между вершинами каждого такого ребра;
 - Перебирать в цикле все закольцованные участки, соответствующие найденным ФЦ в поисках альтернативных путей до целевых элементов трассировки: если очередной ФЦ образует альтернативный путь, то минимум две вершины, входящие в него, должны также принадлежать уже построенному набору путей. При соблюдении этого условия данный ФЦ включается в набор построенных трасс и исключается из списка поиска. Работа алгоритма заканчивается, когда на очередной итерации алгоритма список не включенных в трассировку ФЦ оказывается пуст, либо на предыдущей итерации ни один из ФЦ не был добавлен в трассировку.

3.1.1. До нормальных токоразделов

Осуществляется поиск ближайшего нормального токораздела во все стороны от выбранного оборудования. Метрикой близости является количество КА, через которые нужно пройти, чтобы прийти до данного токораздела. Если сразу несколько нормальных токоразделов имеют минимальное и одинаковое расстояние от выбранного оборудования, то трассировка будет осуществлена до каждого из них. Трассируются только кратчайшие пути.

3.1.2. От текущих токоразделов

Осуществляется поиск ближайшего текущего токораздела в пределах топологического острова во все стороны от выбранного оборудования. Метрикой близости является количество КА, через которые нужно пройти, чтобы прийти до данного токораздела. Если сразу несколько текущих токоразделов имеют минимальное и одинаковое расстояние от выбранного оборудования, то трассировка будет осуществлена до каждого из них. Трассируются только кратчайшие пути.

3.1.3. Топологический остров

Находит все оборудование, гальванически или магнитно связанное с данным оборудованием.

3.1.4. В сторону питания

Осуществляется поиск активных источников энергии во все стороны от выбранного оборудования в пределах топологического острова. Трассируются все альтернативные пути.

3.1.5. В сторону потребления

Осуществляется поиск потребителей энергии во все стороны от выбранного оборудования в пределах топологического острова. Трассируются только те потребители, которые достижимы через терминалы выбранного оборудования, с которых не достижимы напрямую активные источники энергии, другими словами: трассируемые потребители не должны быть запитаны, минуя выбранное оборудование. Если выбранное оборудование обесточено, то трассировка также не производится. Трассируются все альтернативные пути.

3.1.6. До ближайших КА

Осуществляется поиск во все стороны до ближайших коммутационных аппаратов. Если ближайшим КА является КА в виде центрального оборудования с токоразрывающей способностью (выключатели) и двух обрамляющих разъединителей, то поиск доходит до центрального элемента. Трассируются все альтернативные пути.

3.1.7. Поиск закольцованных участков

Осуществляется поиск всех замкнутых траекторий в пределах того же топологического острова, что и выбранное оборудование.

3.2. Используемые методы

Основные методы:

- *Trace::CalcGeneratorsTrace()* – расчет трассировки до источника;
- *Trace::CalcConsumeTrace()* – расчет трассировки до потребителей;
- *Trace::CalcCaTrace()* – расчет трассировки до ближайших КА;
- *Trace::CalcCyclesTrace()* – определение наличия и индикация на схеме циклически замкнутых участков сети;
- *Trace::CalcTopologyIslandTrace()* – расчет «топологических островов» на графе сети;
- *Trace::CalcNormalCutZoneTrace()* – расчет трассировки до нормальных токоразделов;
- *Trace::CalcCurrentCutZoneTrace()* – расчет трассировки до текущих токоразделов.

3.3. Структура программы с описанием функций составных частей и связи между ними

Основная часть алгоритмов трассировки в SCADA сосредоточена в классе *Trace*, который отвечает за расчет конкретных типов трассировок, а также за ведение списка построенных, активных на схеме и скрытых трассировок. Различные методы данного класса вызываются из обработчиков событий, привязанных к различным элементам графического интерфейса пользователя SCADA и связанных с функцией трассировки (пункты главного и контекстного меню, кнопки, поля списков, элементы мнемосхемы и т.п.).

3.4. Связи программы с другими программами

В программе используется объектная модель, созданная в программе TOPAZ Model Creator. Также программа связана с программой TOPAZ DBAL.

4. ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

Приложение поддерживает аппаратные средства, основанные на АРМ-архитектуре. Необходимые требования к аппаратной части пакет приложений представлены в таблицах ниже.

Таблица 1 – Требования к серверу доступа к данным

Наименование параметра	Значение
ЦП	Не менее 4-х ядер, не менее 1,2 ГГц
ОЗУ	Не менее 4 Гб
ПЗУ (системный накопитель)	SSD, не менее 8 Гб
ПЗУ (накопитель БД)	4 x 2.5-inch HDD, емкость носителя определяется количеством

Наименование параметра	Значение
	узлов в сети и глубиной архивирования, RAID10
ОС	Операционные системы реального времени на основе ядра LINUX
Ethernet	Не менее 2 шт, 1 Гбит/с

Таблица 2 – Требования к АРМ пользователя

Наименование параметра	Значение
ЦП	Не менее 4-х ядер, не менее 1,2 ГГц
ОЗУ	Не менее 4 Гб
Видеокарта	Дискретная
ПЗУ (системный накопитель)	SSD, не менее 8 Гб
Диагональ монитора, не менее	27"
Ethernet	Не менее 2 шт, 1 Гбит/с

5. ВЫЗОВ И ЗАГРУЗКА

5.1. Способ вызова программы

Вызов программы производится пользователем из интерфейса клиента системы SCADA.

5.2. Входные точки в программу

Входные точки в программу – класс *Trace*.

6. ВХОДНЫЕ ДАННЫЕ

6.1. Характер, организация и предварительная подготовка входных данных

Входными данными являются пользовательские данные (текстовая или цифровая информация) и данные от первичных источников информации (оборудование подстанции). Организация хранения в соответствии со структурой БД системы SCADA.

7. ВЫХОДНЫЕ ДАННЫЕ

7.1. Характер, организация и предварительная подготовка выходных данных

Выходные данные – отображение режимов трассировки в интерфейсе клиента системы SCADA.