

Федеральное агентство по образованию
Государственное образовательное учреждение
высшего профессионального образования
Петрозаводский государственный университет
Математический факультет

Кафедра информатики
и математического обеспечения

СПЕЦИФИКАЦИЯ ТРЕБОВАНИЙ
для расширения программной системы
WEB-SYNDIC

Заказчик:
к. ф.-м. н.,
доцент Корзун Д. Ж.

Последняя модификация:
Срд Апр 4 02:13:38 MSD 2007

Петрозаводск — 2007

Содержание

Описание назначения программной системы	4
1 Первичный список требований	4
2 Модель предметной области	5
2.1 Глоссарий терминов	5
2.2 Системы диофантовых уравнений	6
2.3 Система WEB-SYNDIC	7
2.3.1 Архитектура текущей версии WEB-SYNDIC	7
2.4 Моделирование p2p-сетей	8
2.4.1 Протокол Chord	8
2.4.2 Генерация топологии	8
2.4.3 Грамматика маршрутизации	9
2.4.4 Построение систем АНЛДУ	9
2.5 Обработка потоков сетевого трафика	9
3 Модели пользователей системы	10
4 Функциональная модель	10
4.1 Детальное описание функций системы	11
4.1.1 Модификация AlgorithmServer	11
4.2 Подсистема моделирования и анализа p2p-сетей	11
4.2.1 Клиент сервера потоков (Flow Client)	12
4.2.2 Сервер потоков (Flow Server)	12
4.3 Сценарии работы с точки зрения пользователя	13
4.3.1 Анализ топологии сети	13
4.3.2 Анализ данных сетевого трафика	13
4.4 Сценарии работы с точки зрения системы	14
4.4.1 Подсистема моделирования и анализа p2p-сетей	14
4.4.2 Сценарий работы сервера потоков	16
4.5 Высокоуровневая архитектура системы	16
4.6 Изменения архитектуры WEB-SYNDIC	16
4.6.1 Высокоуровневая архитектура сервера потоков (Flow Server)	17
4.7 Внешние алгоритмы сервера потоков	17
4.7.1 Алгоритм разбиения	17
4.7.2 Алгоритм построения переходов	18
4.8 Алгоритм построения систем АНЛДУ	18
5 Критерии аттестации системы	18
5.1 Тесты подсистемы анализа топологии p2p-сети	18
5.2 Тесты подсистемы анализа данных сетевого трафика	20
5.2.1 Общие тесты	20
5.2.2 Тесты сервера потоков	20

Список источников

Описание назначения программной системы

Разрабатываемая программная система должна представлять собой дальнейшее развитие проекта WEB-SYNDIC по следующим направлениям: поддержка моделей сетевого трафика на уровне потоков и моделей маршрутизации в р2р-сети. Предварительно должен быть произведен рефакторинг существующей системы WEB-SYNDIC, направленный на унификацию работы с внешними алгоритмами подсистемы «Algorithm Server».

1 Первичный список требований

1) Модификация архитектуры WEB-SYNDIC

(a) Унификация работы с внешними алгоритмами:

- i. решателями
- ii. генераторами систем АНЛДУ
- iii. генераторами топологий сети
- iv. прочими алгоритмами (например, обработки статистики)

(b) Унификация понятия «задача»:

- i. решения системы
- ii. генерации системы
- iii. построения модели сети и т. п.

2) Моделирование топологии р2р-сети

Необходимо расширить функциональность WEB-SYNDIC таким образом, чтобы пользователь получил возможность моделировать топологию Chord-подобных и других одноранговых сетей [1] и производить анализ такой топологии (поиск циклических маршрутов, выявление наиболее загруженных участков сети и т. п.).

Модель маршрутизации формулируется в виде системы АНЛДУ, построенной на основе топологии сети. Решения этой системы определяют маршруты в сети.

(a) Требования

Система должна:

- i. Генерировать различные варианты Chord-подобных топологий
- ii. На основе топологии (сгенерированной, либо заданной пользователем) построить систему АНЛДУ.
- iii. Поддерживать различные варианты построения систем АНЛДУ
- iv. Осуществлять визуализацию глобальной топологии и суграфов маршрутов (найденных в результате решения систем АНЛДУ)

(b) Ограничения

- i. Необходима возможность подключения различных генераторов топологий сети.
 - ii. В качестве исходных данных для построения системы АНЛДУ должны выступать данные, полученные при генерации топологии, либо заданные пользователем
 - iii. Генерируемая система АНЛДУ должна поддерживаться WEB-SYNdic
- 3) Анализ данных потоков сетевого трафика
- Разрабатываемое расширение должно предоставлять пользователю возможность моделирования структуры потоков сетевого трафика, при этом должны выявляться устойчивые закономерности, имеющиеся в распределении потоков данных на заданном отрезке времени.
- (a) Требования
- Система должна:
- i. На основе заданных параметров производить выборку данных о потоках сетевого трафика
 - ii. На основе полученной выборки строить систему АНЛДУ.
 - iii. По требованию пользователя помимо системы АНЛДУ и ее решения выводить построенную систему переходов
 - iv. Предоставлять пользователю возможность самостоятельно редактировать систему переходов
 - v. Осуществлять визуальное представление решения системы АНЛДУ (в терминах системы переходов)
- (b) Ограничения
- i. Входными данными для анализа трафика должны быть файлы в формате Cisco NetFlow.
 - ii. Генерируемая система АНЛДУ должна поддерживаться WEB-SYNdic.
 - iii. Модуль, осуществляющий обработку трафика, должен быть реализован в виде отдельного сервера, взаимодействующего с остальной системой посредством сетевого протокола.

2 Модель предметной области

2.1 Глоссарий терминов

Одноранговая, пиринговая или P2P-сеть — сеть без четкого централизованного контроля или строгой иерархии. В такой сети специальные сервера отсутствуют, или выполняют вспомогательные функции, каждый узел может выступать как в роли клиента, так и в роли сервера.

Топология сети — описание компьютерной сети, схема взаимодействия сетевых устройств. Классифицируется на физическую, описывающую реальное расположение и связи между узлами, и логическую, описывающую хождение сигнала в пределах физической топологии. В данном проекте под понятием «топология» полагается логическая топология, если не указано иначе.

Граф топологии сети — граф, описывающий топологию сети. Вершинам графа соответствуют узлы, ребрам — каналы связи между ними. В случаях, когда связь односторонняя, т. е. узел a «знает» и может установить связь с узлом b , но не наоборот (как в системе Chord), используется ориентированный граф.

Chord — проект по разработке надежной масштабируемой распределенной системы, основанной на принципах р2р сети. Основой для разработки является сетевой протокол поиска Chord.

Трафик — объем информации, переданной по сети.

NetFlow — название системы учета трафика фирмы Cisco Systems и формата хранения данных этой системы.

Диофантово уравнение, уравнение в неотрицательных целых числах — уравнение с целыми коэффициентами или неизвестными, которые могут принимать только неотрицательные целые значения.

Система АНЛДУ — ассоциированная с КС-грамматикой система неотрицательных линейных диофантовых уравнений.

КС-грамматика — частный случай грамматики Хомского, грамматика, у которой левые части всех продукций являются нетерминалами. Смысл термина «контекстно-свободная» заключается в том, что возможность применить продукцию к нетерминалу, в отличие от общего случая грамматики Хомского, не зависит от контекста этого нетерминала.

Неразложимое решение системы АНЛДУ — решение системы АНЛДУ, которое нельзя представить в виде суммы двух ненулевых решений.

Решение системы АНЛДУ — вектор значений переменных, при подстановке в систему уравнений превращающий ее в набор верных тождеств.

Базис Гильберта — множество всех неразложимых решений системы АНЛДУ.

2.2 Системы диофантовых уравнений

Определение 1. Однородной системой неотрицательных линейных диофантовых уравнений (одНЛДУ) называется система уравнений, коэффициенты которых — целые числа, а компоненты решений могут принимать лишь целые неотрицательные значения.

Определение 2. Ненулевое решение называется неразложимым, если оно не может быть представлено в виде суммы двух ненулевых решений этой же системы.

Определение 3. Базисом Гильберта однородной системы НЛДУ называется множество всех неразложимых решений этой системы.

Одним из практических приложений задач решения систем АНЛДУ (определение совместимости, нахождение частных решений и базиса Гильберта) являются задачи моделирования маршрутизации вычислительных сетей и сетевого трафика.

2.3 Система WEB-SYNDIC

WEB-SYNDIC — это система, предназначенная для демонстрации и тестирования синтаксических методов нахождения неотрицательных целочисленных решений диофантовых уравнений. Текущая версия не поддерживает работу с диофантовыми моделями сетевых систем.

2.3.1 Архитектура текущей версии WEB-SYNDIC

Схема архитектуры текущей версии WEB-SYNDIC изображена на рис. 1. Ниже приведено краткое описание каждого из модулей.

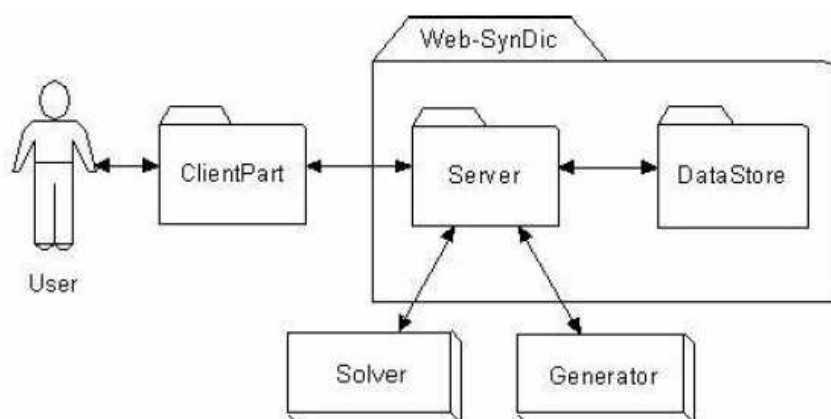


Рис. 1: Высокоуровневая архитектура WEB-SYNDIC

ClientPart — стандартный обозреватель интернета.

Server — основная часть системы, осуществляющая координацию между другими подсистемами. Основные составляющие: Web Server, Session Processing, Activity Statistics, Algorithm Server, Management.

DataStore — подсистема, предназначенная для хранения данных по профилям и активности пользователей.

Generator — реализация алгоритмов генерации тестовых систем АНЛДУ и их базисов Гильберта.

Solver — реализация алгоритмов решения систем одАНЛДУ.

Пользователь непосредственно взаимодействует только с ClientPart, через которую он задает параметры генерации системы одАНЛДУ и получает отчет о решении.

2.4 Моделирование p2p-сетей

Задача моделирования маршрутизации возникает при исследовании вычислительных сетей.

2.4.1 Протокол Chord

Протокол Chord [1, 8, 9] поддерживает одну основную операцию: получив ключ, ставит ему в соответствие определенный узел. Для оптимизации поиска, узлы и ключи располагаются на т. н. кольце идентификаторов. Согласно теоретическим исследованиям и практическим испытаниям, сложность алгоритма поиска составляет $O(\log_2 N)$, где N — количество узлов в инфраструктуре.

Для генерации графа топологии, соответствующей инфраструктуре Chord, требуется задать единственный параметр — количество узлов в инфраструктуре. Задача генерации решалась в рамках курсовой работы О. Прасол [6].

Пример топологии Chord приведен на рис. 2. Ключи и узлы располагаются на кольце размером 2^m . Владельцем ключа k назначается первый узел, идентификатор которого больше либо равен k . В терминах системы этот узел называют *преемником* (*successor*) ключа k . Если передвигаться по кольцу идентификаторов в положительном направлении, то $successor(k)$ — первый узел, встречающийся после ключа k .

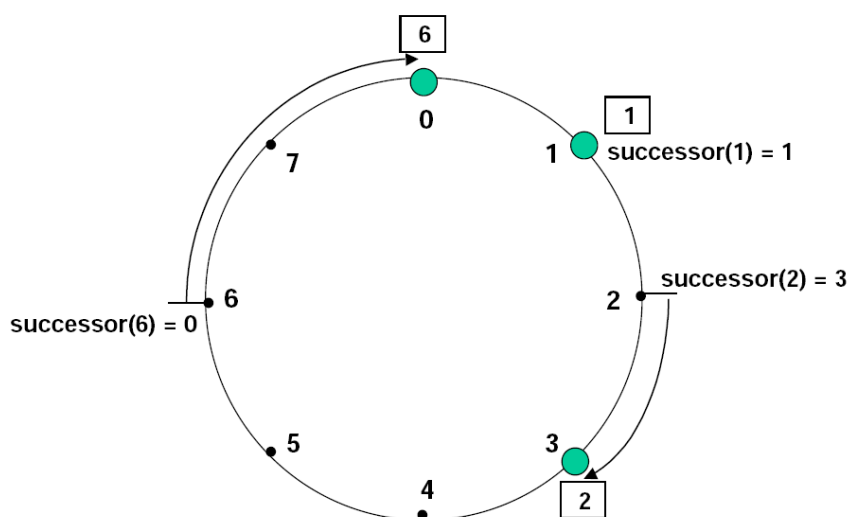


Рис. 2: Пример топологии Chord

2.4.2 Генерация топологии

Задача генерации топологии сети была реализована в [6].

2.4.3 Грамматика маршрутизации

Для моделирования маршрутизации используется подход, основанный на применении формальных грамматик, линейных диофантовых уравнений и базисов Гильберта.

Рассмотрим инфраструктуру из N узлов $S = \{S_1, \dots, S_n\}$. Каждый узел имеет свою таблицу маршрутизации с несколькими записями и может отправить сообщение любому из узлов, указанных в этой таблице. Пусть узлом s получено сообщение, и этот узел не является адресатом. Тогда в таблице маршрутизации выбирается в точности один (в более сложных случаях — несколько) узел, и сообщение перенаправляется туда. Запишем все варианты отправки в виде набора правил:

$$s \rightarrow v^{k_v}, \quad v \in T_s, \quad s \in S$$

Здесь T_s — таблица маршрутизации узла s , k_v — число попыток передачи. Если в правой части формулы находится пустая цепочка, такое правило описывает завершение пути следования сообщения.

Пусть R_s — множество всех таких правил для узла s . Получаем коммутативную КС-грамматику без начального символа:

$$G = (S, \Sigma, R, \cdot)$$

2.4.4 Построение систем АНЛДУ

Построим по полученной ранее грамматике следующую систему одАНЛДУ:

$$E(R)x = Ax$$

Здесь каждому узлу s соответствует уравнение, каждому правилу — переменная. Каждому решению системы соответствует контур. Каждому базисному решению — простой контур. Значения переменных x_r равны числу применений правила r . $E(R)$ — матрица, определяющая распределение правил по узлам инфраструктуры: $E_{sr} = 1$ тогда и только тогда, когда узел s использует правило r , иначе $E_{sr} = 0$. Матрица A строится в соответствии с правыми частями правил грамматики. Строки соответствуют узлам, столбцы — правилам. Значение элемента столбца равно числу появлений узла, определяемого строкой, в данном правиле. В простейших случаях матрица A является бинарной. Алгоритм построения систем АНЛДУ, основанных на графе топологии сети, был реализован ранее в [5].

2.5 Обработка потоков сетевого трафика

Определение 1. Поток сетевого трафика — однонаправленная последовательность пакетов между двумя точками вычислительной сети, характеристики которых удовлетворяют заданным условиям. Совокупность этих условий называется определением потока.

По потокам может быть разделен трафик практически любого сетевого или транспортного протокола.

Формат *Cisco NetFlow* был разработан компанией Cisco Systems, и является частью одноименной системы учета сетевого трафика. Данные о потоках трафика генерируются маршрутизаторами Cisco и сохраняются в этом формате. Эти данные содержат сведения, позволяющие идентифицировать поток, а также служебную информацию.

В работе системы обработки потоков трафика можно выделить следующие основные этапы:

- 1) Полученные потоки разбиваются на группы по времени их прохождения.
- 2) Строятся переходы между полученными группами потоков.
- 3) На основании полученных данных строится система АНЛДУ

Задачи разбиения потоков и построения переходов решались в ходе курсовых работ И. Шмарова [7], А. Ломова [3] и С. Караваева [4]. При этом активно применялся программный комплекс ТСРсопап.

Задачи разбиения потоков и построения переходов решались в ходе курсовых работ [4, 3, 7]. При этом активно применялся программный комплекс ТСРсопап.

3 Модели пользователей системы

Система WEB-SYNDIC поддерживает две разновидности пользователей:

- 1) Обычные пользователи — исследователи предметной области системы. В свою очередь, подразделяются на гостей и зарегистрированных пользователей. К возможностям обычных пользователей относятся генерация и решение систем АНЛДУ, задание параметров в рамках текущей сессии. После модификации системы к этим возможностям добавятся построение моделей маршрутизации в р2р-сетях и моделей структуры сетевого трафика на уровне потоков.
- 2) Администраторы системы, осуществляющие настройку системы, задание базовых параметров, недоступных для остальных пользователей и т. п.

Управление пользователями производится существующими средствами WEB-SYNDIC. Модификация подсистемы управления пользователями не планируется. Кроме того, добавится еще один вид пользователя — администратор сервера потоков. Этот пользователь будет иметь возможность непосредственно редактировать файл конфигурации сервера потоков, задавая основные параметры этого сервера.

4 Функциональная модель

В соответствии с требованиями заказчика, схему работы с системой (с точки зрения пользователя) можно представить следующей диаграммой (3):

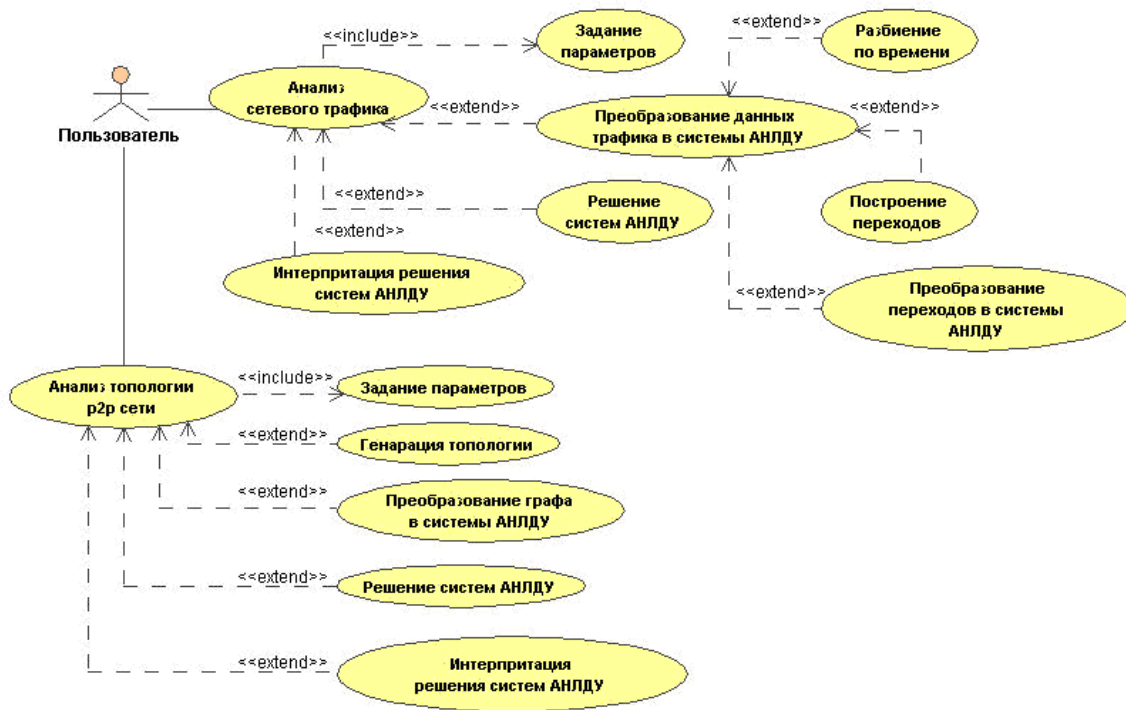


Рис. 3: Сценарий использования разрабатываемой системы

4.1 Детальное описание функций системы

4.1.1 Модификация AlgorithmServer

Основное назначение алгоритм-сервера — осуществление взаимодействия с внешними алгоритмами. В настоящий момент сервер поддерживает две разновидности алгоритмов: генерации систем АНЛДУ и решения таких систем (или их наборов). В соответствии с этим, существуют два класса задач: задачи генерации систем и задачи решения систем. Архитектура подсистемы спроектирована таким образом, что эти два вида задач хотя и являются частью одного механизма, но по сути представляют собой разные subsystemы алгоритм-сервера. В основном это связано с тем, что различные задачи для своего выполнения требуют задания различных параметров и, соответственно, выдают разные результаты.

Подобное положение дел затрудняет модификацию системы, направленную на расширение круга задач, выполняемых алгоритм-сервером. В частности, это относится и к функциональностям, описанным в разделе основных требований.

В связи с этим, предлагается в корне модифицировать структуру алгоритм-сервера таким образом, чтобы унифицировать понятия «алгоритм» и «задача».

4.2 Подсистема моделирования и анализа р2р-сетей

В соответствии со сформулированными выше требованиями к алгоритм-серверу, данная подсистема должна будет представлять собой не что иное, как модуль алгоритм-сервера, решающий задачи моделирования и анализа одноранговых (р2р) сетей. В соответствии с этим, можно сформулировать детальные требования

к подсистеме:

1) Должны вводиться три новых класса задач для алгоритм-сервера:

- задача генерации модели р2р-сети
- задача построения системы АНЛДУ на основе модели сети
- задача интерпретации решений системы АНЛДУ в контексте анализа р2р-сети

Задачи должны быть реализованы таким образом, чтобы выходные данные одних можно было использовать в качестве входных данных других.

2) Для соответствующих задач должны быть созданы внешние алгоритмы

Кроме того, для осуществления взаимодействия с пользователем, должны быть добавлены соответствующие элементы (страницы) в web-интерфейс системы.

4.2.1 Клиент сервера потоков (Flow Client)

Данная подсистема будет, как и предыдущая, представлять собой набор задач для алгоритм-сервера. С точки зрения последнего, существование сервера потоков неочевидно, он должен восприниматься как рядовой внешний алгоритм. Предполагается наличие задач:

- Задача выборки данных о потоках трафика и построения системы переходов
- Задача построения системы АНЛДУ на основе этих данных
- Задача интерпретации решений АНЛДУ в терминах системы переходов

Как и для задачи моделирования/анализа р2р-сети, должен быть реализован соответствующий модуль web-интерфейса системы.

Подсистема должна реализовывать клиентскую часть сетевого протокола взаимодействия с сервером потоков и осуществлять передачу параметров для анализа потоков серверу и получения от сервера результатов анализа.

4.2.2 Сервер потоков (Flow Server)

1) Программно должен представлять собой демон UNIX (разработка и тестирование будут производиться в ОС SUSE GNU/Linux).

2) Должен реализовывать следующие функциональности:

- (a) Получение параметров операции разбиения потоков.
- (b) Выборка данных из базы потоков в соответствии с параметрами разбиения.
- (c) Построение систем АНЛДУ на основе данных выборки.
- (d) Возврат системы АНЛДУ пользователю.

- 3) На время выполнения задачи сервер должен блокироваться и не принимать новые задачи до окончания выполнения текущей. Должна быть предусмотрена возможность дальнейшего расширения сервера с поддержкой выполнения параллельных задач.
- 4) Должен обеспечивать связь с пользователем посредством специального сетевого протокола. Основные требования к протоколу перечислены ниже:
 - (a) Роль сервера должен выполнять сервер потоков.
 - (b) Протокол должен быть ориентирован на установление соединения.
 - (c) Сервер должен поддерживать работу с одним клиентом одновременно. В случае попытки соединения с занятым сервером, должно отправляться соответствующее сообщение, а клиент ставиться в очередь на выполнение задачи.
 - (d) Протокол должен поддерживать операции установления сессии связи между сервером потоков и клиентом, отправку параметров выборки и разбиения от клиента к серверу, отправку результатов работы от сервера клиенту.

4.3 Сценарии работы с точки зрения пользователя

4.3.1 Анализ топологии сети

- 1) Задание параметров генерации графа.
- 2) Запуск задачи на выполнение.
- 3) Генерация графа.
- 4) Просмотр результатов генерации.
- 5) Задание параметров анализа.
- 6) Запуск задачи на выполнение.
- 7) Ожидание завершения выполнения задачи (возможно, получение промежуточных результатов).
- 8) Получение результатов работы системы.

4.3.2 Анализ данных сетевого трафика

- 1) Задание параметров выборки и разбиения данных трафика
- 2) Запуск задачи на выполнение
- 3) Ожидание завершения выполнения задачи (возможно, получение промежуточных результатов).

4) Получение результатов работы системы

Эти сценарии описываются диаграммами (4) и (??).

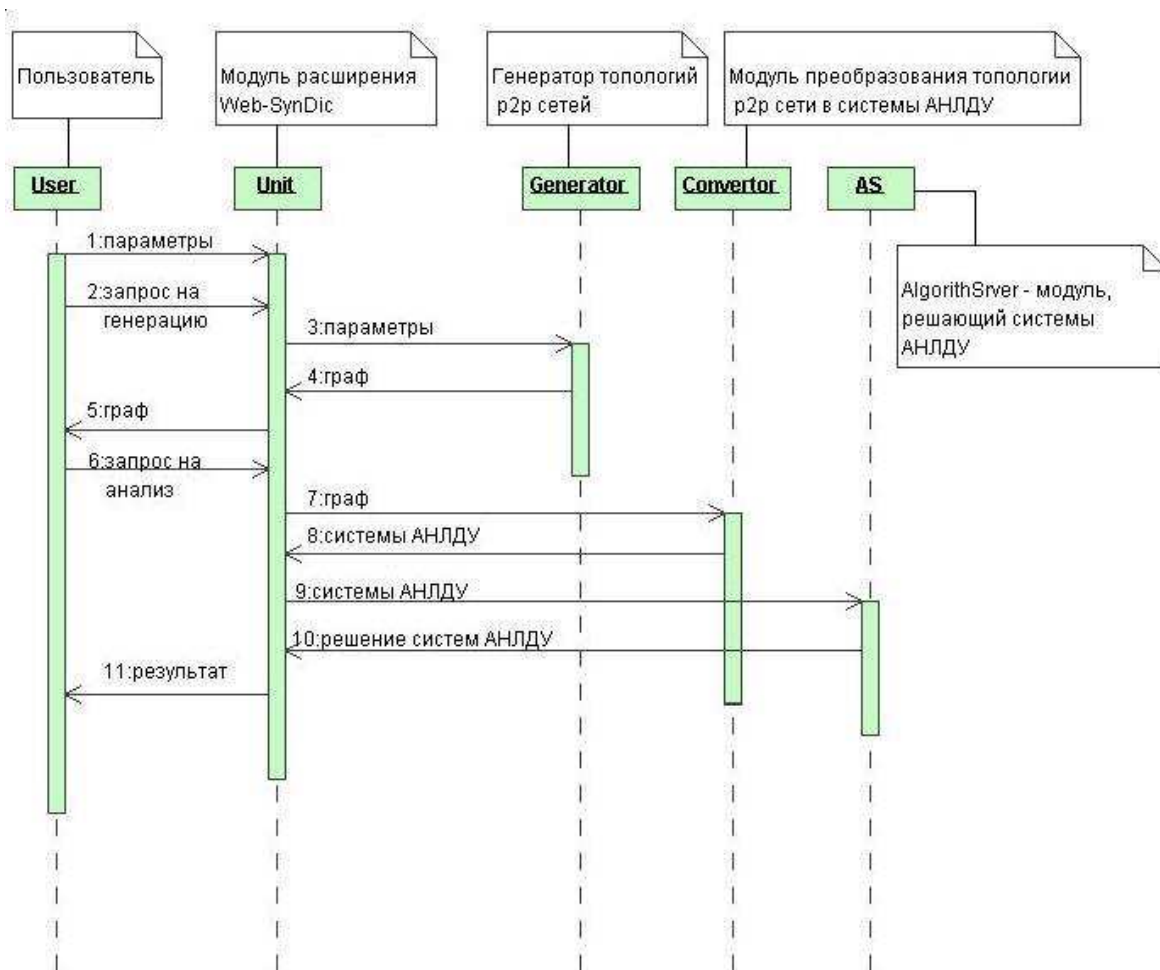


Рис. 4: Сценарий работы подсистемы анализа топологии р2р-сети

4.4 Сценарии работы с точки зрения системы

4.4.1 Подсистема моделирования и анализа р2р-сетей

- 1) **Получение параметров генерации.** На этом этапе от пользователя получается размер (количество узлов) генерируемой Chord-подобной топологии. Также на этом этапе выбирается используемый генератор (из списка доступных)
- 2) **Генерация графа топологии.** Вызывается внешний генератор графа топологии сети, которому передаются параметры, полученные на предыдущем шаге. Выбранный генератор возвращает граф в формате списка ребер.

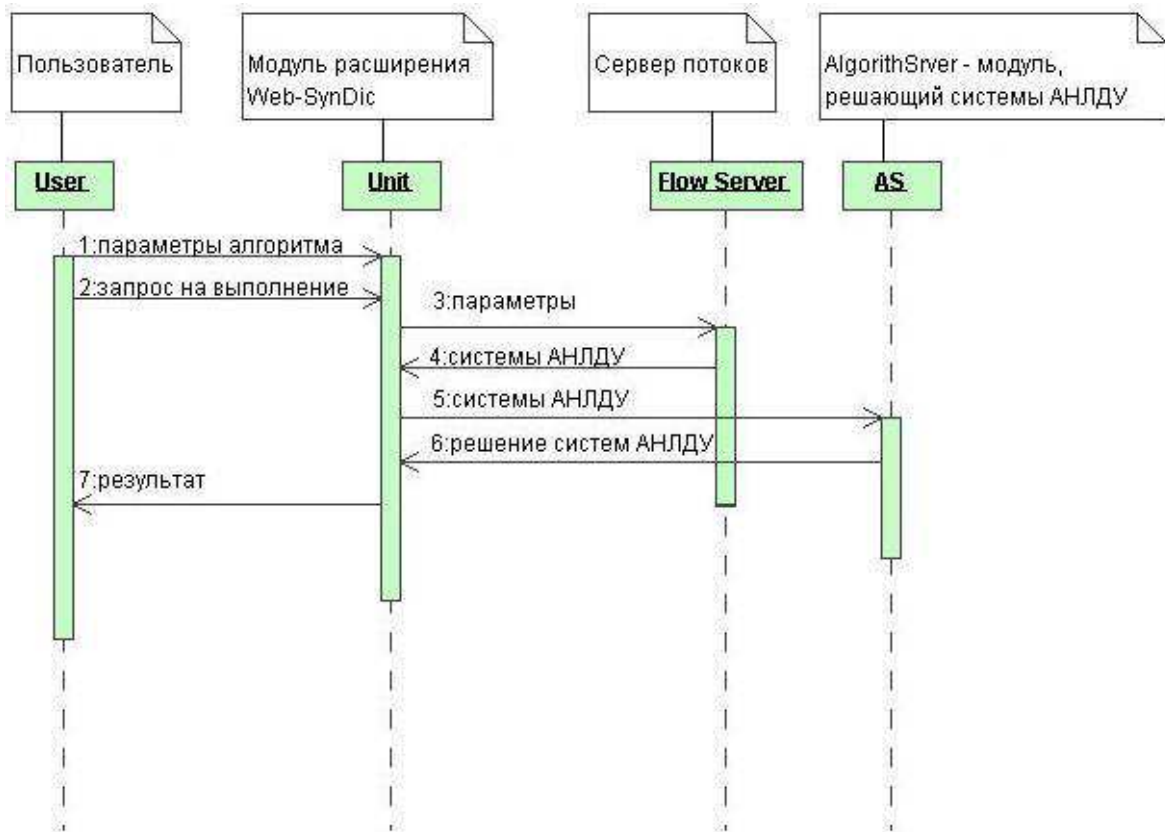


Рис. 5: Сценарий работы подсистемы анализа данных сетевого трафика

- 3) **Визуализация топологии сети.** Пользователю предоставляется сгенерированный граф топологии. Должна быть предоставлена возможность сохранения результата в файл. Также по запросу может быть выведено наглядное представление графа. Для реализации этой задачи предполагается использовать одну из существующих библиотек визуализации, выбор которой будет осуществлен в процессе конструирования системы.
- 4) **Преобразование графа в систему АНЛДУ.** После получения сгенерированного графа, пользователю предоставляется возможность выбрать один из доступных генераторов систем АНЛДУ на основе графа сети. После запуска задачи на выполнение, система осуществляет вызов генератора и выводит полученный результат пользователю.
- 5) **Решение системы АНЛДУ.** Сгенерированная система АНЛДУ отправляется на решение с помощью существующих алгоритмов решения WEB-SYNDic.
- 6) **Отображение результатов.** Результаты выводятся пользователю через Web-интерфейс. Должна быть предусмотрена возможность сохранения результатов в файл. Кроме того, должна осуществляться полная или частичная визуализация решения: например, отображение циклических маршрутов.

4.4.2 Сценарий работы сервера потоков

- 1) Ожидание запроса от клиента
- 2) Получение запроса на выполнение задачи с параметрами
- 3) Если в данный момент выполняется хоть одна задача, помещение задачи в очередь, иначе на шаг 5
- 4) Вызов модулем SessionManager функции модуля AlgorithmManager на выполнение преобразования
- 5) Если задача на построение системы АНЛДУ на основе данных о потоках сетевого трафика, то на шаг 7. Иначе на шаг 9 (работа с системой переходов).
- 6) Модуль AlgorithmManager с учетом параметров вызывает выполнение алгоритма разбиений
- 7) Модуль AlgorithmManager вызывает на выполнение алгоритма переходов, в качестве входных данных посылается результат выполнения алгоритма разбиений
- 8) Модуль AlgorithmManager вызывает на выполнение алгоритма построения АНЛДУ, в качестве входных данных посылается результат выполнения алгоритма переходов
- 9) Передача полученных систем АНЛДУ модулем SessionManager клиенту
- 10) Если есть еще задачи в очереди, то извлечение задачи из начала очереди и переход на шаг 5.
- 11) Возврат к шагу 1

4.5 Высокоуровневая архитектура системы

4.6 Изменения архитектуры WEB-SYNDIC

Изменение структуры WEB-SYNDIC состоит в том, чтобы унифицировать работу с внешними решателями и генераторами систем АНЛДУ. Т. е. главные изменения коснутся подсистемы «Server». Необходимо изменить работу с внешними модулями таким образом, чтобы подключение новых внешних модулей не затрагивало систему в целом. В связи с тем, что в качестве внешних модулей могут выступать программы, несущие на себе различную функциональную нагрузку, необходимо придумать простой и эффективный способ взаимодействия между самой системой и внешними и модулями.

4.6.1 Высокоуровневая архитектура сервера потоков (Flow Server)

1) Модуль SessionManager

- (a) Должен отвечать за реализацию протокола взаимодействия с клиентом
- (b) Должен вызывать функции модуля алгоритмов для преобразования данных сетевого трафика в системы АНЛДУ
- (c) Должен предоставлять возможность постановки нескольких задач на выполнение
- (d) Должен ставить задачи в очередь на выполнение
- (e) В каждый момент времени выполняется только одна задача, остальные находятся в очереди на выполнение

2) Модуль AlgorithmManager

- (a) Должен выполнять преобразования данных сетевого трафика в системы АНЛДУ, используя вызовы алгоритмов разбиения, переходы и построения АНЛДУ в соответствии с параметрами преобразования
- (b) Должен предоставлять функции для преобразования данных сетевого трафика в системы АНЛДУ, параметрами которых должны являться параметры преобразования, а возвращаемыми значениями должны быть системы АНЛДУ

3) Алгоритм разбиения

Это должен быть модуль, содержащий реализации алгоритмов разбиения, рассмотренных в [7] и [3]

Входными данными алгоритма должны быть файлы с данными сетевого трафика в формате NetFlow из хранилища, выбранные с учетом параметров преобразования.

4) Алгоритм переходов

Это должен быть модуль, содержащий алгоритм переходов, рассмотренный в [4].

5) Алгоритм построения АНЛДУ

Это должен быть модуль, содержащий преобразование результата выполнения алгоритма переходов в систему АНЛДУ.

4.7 Внешние алгоритмы сервера потоков

Каждый из трех указанных на схеме алгоритмов реализуется в виде самостоятельных программ, вызываемых сервером потоков.

4.7.1 Алгоритм разбиения

Данный алгоритм был реализован ранее С. Караваевым в рамках курсовой работы [4]. Разработанная утилита производит разбиение промежутка времени на интервалы, а затем определяет для каждого потока один или несколько интервалов времени, которым принадлежит данный поток.

Выходные данные программы — последовательность чисел (состояний), соответствующих интервалам времени, которым принадлежали потоки.

4.7.2 Алгоритм построения переходов

Формат входных данных этого алгоритма соответствует формату выходных данных предыдущего. В ходе работы подсчитывается количество переходов от одного состояния к другому, выясняется, какие переходы были наиболее частыми.

4.8 Алгоритм построения систем АНЛДУ

Данный алгоритм должен быть реализован на текущей итерации. Основные требования:

- 1) Формат входных данных этого алгоритма должен совпадать с форматом выходных данных предыдущего.
- 2) Выходными данными должна являться система АНЛДУ в формате, приемлемом для решения с помощью WEB-SYNDIC.

5 Критерии аттестации системы

Критерии аттестации системы включают в себя тесты каждой из заявленных выше функциональностей. Их перечень будет расширяться по мере уточнения требований к системе.

5.1 Тесты подсистемы анализа топологии р2р-сети

1) Задание параметров

- (a) Возможность задания размера генерируемой инфраструктуры Chord. В ходе тестов должны передаваться различные значения параметра, система должна генерировать инфраструктуру Chord заданного размера.
- (b) Реакция системы на попытку ввода неверных параметров. В этой ситуации система должна выдавать пользователю соответствующее сообщение.
- (c) В случае реализации возможности задания пользовательского графа топологии — проверка синтаксической и семантической правильности вводимых данных.

2) Генерация топологии сети

- (a) Тесты, проверяющие правильность вызова внешнего генератора, а также поведение системы в случае невозможности вызова последнего, или ошибки взаимодействия с ним на всех стадиях. Если в процессе взаимодействия произойдет ошибка, система должна выводить пользователю соответствующее сообщение.

- (b) Проверка синтаксической и семантической правильности генерируемого графа для различных параметров. Поведение системы в случае неверной генерации.
- (c) Тесты на работу системы в вырожденных случаях (пустой граф) и в случаях слишком большого размера графа.
- (d) Временные тесты: оценка времени генерации графа для различных размеров топологии.

3) Визуализация топологии сети

- (a) Тесты на возможность сохранения результатов в файл. Сохраненная система должна соответствовать отображаемой. В случае, если файл существует, должно выводиться предупреждение о его перезаписи.
- (b) Проверка отображаемого наглядного отображения графа. Осуществляется путем демонстрации пользователям изображений, в процессе демонстрации их замечания фиксируются в протоколе тестирования, после чего осуществляется доработка подсистемы в соответствии с протоколом.

4) Генерация системы АНЛДУ на основе графа топологии

- (a) Тесты, проверяющие правильность вызова внешнего генератора, а также поведение системы в случае невозможности вызова последнего, или ошибки взаимодействия с ним на всех стадиях. Если в процессе взаимодействия происходит ошибка, система должна выводить пользователю соответствующее сообщение.
- (b) Тесты, проверяющие синтаксическую и семантическую правильность сгенерированной системы АНЛДУ, соответствие ее формату, пригодному для решения в WEB-SYNDIC.
- (c) Временные тесты: проверка времени генерации систем уравнений для различных размеров графа.

5) Визуализация сгенерированных систем АНЛДУ

- (a) Тесты, проверяющие соответствие отображаемой системы сгенерированной.
- (b) Тесты, проверяющие возможность сохранения сгенерированной системы в файл.

6) Решение систем АНЛДУ средствами WEB-SYNDIC

- (a) Тесты, проверяющие возможность решения любой сгенерированной системы (если ее решение возможно с использованием существующих алгоритмов).

7) Интерпретация решений систем АНЛДУ

- (a) Более подробно эти тесты будут определены по мере уточнения требований к подсистеме интерпретации решений. Формулировки требований можно будет найти в плане тестирования.

5.2 Тесты подсистемы анализа данных сетевого трафика

5.2.1 Общие тесты

1) Задание параметров

Система должна адекватно реагировать на заданные параметры. В связи с этим должны быть проведены следующие тесты:

- (a) Проверка возможности задания любого допустимого значения каждого из параметров.
- (b) Проверка реакции системы на попытку ввода некорректных параметров.
- (c) Проверка работы системы при корректном задании пользователем системы переходов и при задании системы переходов, на основании которой не может быть построена система АНЛДУ.

2) Взаимодействие с сервером потоков

- (a) Проверка поведения системы в случае недоступности сервера потоков на всех этапах.
- (b) Проверка реализации клиентской части протокола взаимодействия с сервером потоков.

3) Решение систем АНЛДУ средствами WEB-SYNdic

- (a) Проверка возможности решения любой сгенерированной системы средствами текущей версии WEB-SYNdic

4) Интерпретация решений систем АНЛДУ

Тесты будут определены после уточнения требований к подсистеме интерпретации решений.

5.2.2 Тесты сервера потоков

1) Взаимодействие с клиентом

- (a) Проверка поведения сервера в случае возникновения ошибок взаимодействия с клиентом.
- (b) Проверка реализации серверной части протокола сервера потоков.
- (c) Проверка поведения сервера потоков в случае получения некорректных данных.

2) Управление заданиями и вызовы менеджера алгоритмов

3) Взаимодействие с хранилищем данных NetFlow

- (a) Проверка работы сервера в случае недоступности модуля machine программного комплекса tcrcpan.
- (b) Проверка работы сервера в случае возникновения ошибок в работе модуля machine.
- (c) Проверка работы сервера в случае некорректно заданных имен файлов NetFlow.

4) Распределение потоков по интервалам

- (a) Проверка работы алгоритма при некорректно заданных параметрах.
- (b) Проверка работы алгоритма распределения потоков при различных размерах окна.
- (c) Проверка работы алгоритма распределения потоков при различной точности округления.
- (d) Проверка подключения каждого из алгоритмов распределения потоков.

5) Построение системы переходов

- (a) Проверка работы алгоритма при некорректно заданных параметрах.
- (b) Проверка работы алгоритма построения переходов при различной точности определения состояний.
- (c) Проверка работы алгоритма построения переходов при задании различных минимальных значимых количеств появлений переходов.

6) Трансляция системы переходов в системы АНЛДУ

- (a) Проверка работы транслятора при некорректном входном тексте.
- (b) Проверка пригодности сгенерированных систем для решения их средствами Web-SynDic.

Библиографический список использованных источников

Список литературы

- [1] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan, *Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications*, ACM SIGCOMM 2001, San Deigo, CA, August 2001, pp. 149-160., http://pdos.csail.mit.edu/papers/chord:sigcomm01/chord_sigcomm.pdf
- [2] KARGER, D., LEHMAN, E., LEIGHTON, F., LEVINE, M., LEWIN, D., AND PANIGRAHY, R. Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the World Wide Web. In Proceedings of the 29th Annual ACM Symposium on Theory of Computing (El Paso, TX, May 1997), pp. 654-663.
- [3] Ломов А. А. *Задача разбиения потоков по времени*. Курсовая работа. <http://cs.karelia.ru/p/2005/22205/lomov/>
- [4] Караваев С. В. *Анализ зависимости между объемами потоков в сетевом трафике*. Курсовая работа. <http://cs.karelia.ru/p/2005/2204/skaravaev/>
- [5] Пашков Г. А. *Грамматика маршрутизации в P2P-системе*. Курсовая работа. <http://cs.karelia.ru/p/2005/22206/gpashkov/>
- [6] Прасол О. А. *Генерация топологии P2P-системы*. Курсовая работа. <http://cs.karelia.ru/p/2005/22203/prasol>
- [7] Шмаров И. И. *Задача разбиения для анализа сетевого трафика*. Курсовая работа. <http://cs.karelia.ru/p/2005/22206/shmarov/>
- [8] Stoica, I, Morris R., Karger, D., Kaashoek, M. F., AND Balakrishnan, H. *Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications*. Tech. Rep. TR-819, MIT LCS, March 2001. <http://www.pdos.lcs.mit.edu/chord/papers/>.
- [9] *The Chord/DHash Project*, <http://pdos.csail.mit.edu/chord/>