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

Кафедра информатики
и математического обеспечения

Техническое задание

на разработку программной системы, реализующей словарь для базы данных на внешнем запоминающем устройстве.

Заказчик:
зав. кафедрой ИМО, к.т.н.,
доцент Ю. А. Богоявленский

Представлен "____"_____ 2006г.

Петрозаводск — 2006

Оглавление

1 Описание предметной области. 3
1.1 Введение. 3
1.2 Технология анализа трафика. 3
1.3 Основные звенья при осуществлении анализа. 4
1.4 Описание назначения программной системы. 6
1.4.1 Классическая схема обработки базы потоков. 6
1.4.2 Предлагаемая схема работы. 6

2 Первичный список требований к программной системе. 8
2.1 Функциональные требования: 8
2.2 Ограничения: 9

3 Модель требований. 10

4 Архитектура программной системы. 14

5 Критерии аттестации системы. 16

6 Глоссарий. 17

Глава 1

Описание предметной области.

1.1 Введение.

В соответствии с ростом и развитием современных компьютерных сетей объем трафика в них увеличивается взрывными темпами. В связи с этим перед организациями различных уровней возникает необходимость в средствах определения и анализа характеристик сетевого трафика. Это вызвано важностью решения следующих административных задач:

- повышение эффективности политики маршрутизации
- анализ сетевых угроз и атак
- осуществление расчётов за предоставленные услуги сетевого доступа
- проверка корректности работы сети
- ограничение трафика с нежелательным содержанием
- прогнозирование сетевой нагрузки
- наблюдение за активностью некоторых приложений или пользователей
- планирование развития сети

1.2 Технология анализа трафика.

Актуальность проблемы учета трафика и перспективность работы в этом направлении побуждает различные компании к разработке широкого спектра программных и аппаратных решений в данной области.

Одним из признанных лидеров здесь является корпорация Cisco Systems, внедряющая свою технологию учета трафика - NetFlow. Обработка и сбор данных

в формате NetFlow поддерживается большим количеством приложений от сторонних разработчиков. В этой технологии используется распространённый подход при анализе трафика - разбиение его на потоки. Компания Juniper Networks предоставляет аналогичную технологию на своих маршрутизаторах - cflow.

Признанное лидерство и широкая распространённость NetFlow стали причиной использования его как основы разрабатываемого стандарта сбора информации о трафике - Internet Protocol Information Export (IPFIX).

Поток - однонаправленная последовательность пакетов между отправителем и получателем. Поток идентифицируется по IP-адресам, портам источника и конечной точки, протоколу транспортного уровня, полю ToS (Type of Service) IP-пакета, номеру интерфейса, на котором был принят поток. Конечно, такая информация будет не столь подробна, как tcpdump, но в комплексе представляет довольно подробную статистику. Сбор информации о потоках может осуществляется маршрутизаторами компании Cisco Systems, а так же с помощью специализированного программного обеспечения на компьютерах, подключённых к сети.

NetFlow - особенность Internetworking OS (IOS), а также название открытого, но проприетарного протокола для сбора информации о IP-трафике. Маршрутизаторы Cisco могут генерировать записи NetFlow, которые экспортируются в UDP пакетах для специального сборщика данных (поэтому запись о потоке может быть безвозвратно потерянна при перегрузке сети). Маршрутизатор сформирует и отправит запись о потоке только по его окончании. Запись о потоке помимо данных, по которым поток идентифицируется, содержит номер версии протокола, отметки о времени начала и конца потока, объем трафика в потоке. Формат записей о потоках менялся и развивался, широко применялись несколько его версий. Обработка всех потоков на маршрутизаторе может перегрузить его, поэтому существует возможность избирательного сбора данных о потоках, с пропуском некоторого заданного числа потоков.

1.3 Основные звенья при осуществлении анализа.

В работе Netflow используются три основных компонента: сенсор, коллектор и система отображения данных.

- Сенсор - демон, который слушает сеть и фиксирует потоки. Сенсор должен иметь возможность подключиться к хабу (многопортовому повторителю), "зеркалированному"порту коммутатора или любому другому устройству, для просмотра сетевого трафика. Если вы используете систему пакетной фильтрации на базе BSD или Linux, то это превосходное место для сенсора Netflow, так как весь трафик будет проходить через эту точку. Сенсор будет собирать информацию о сеансах и сбрасывать её в коллектор.

- Коллектор - второй демон, который слушает на UDP порту, указанному вами и осуществляет сбор информации от сенсора. Полученные данные он сбрасывает в файл для дальнейшей обработки. Различные коллекторы сохраняют данные в различных форматах.
- Наконец, система обработки читает эти файлы и генерирует отчёты в форме, более удобной для человека. Эта система должна быть совместима с форматом данных, предоставляемых коллектором.

Программное обеспечение сенсоров, коллекторов и систем отображения весьма

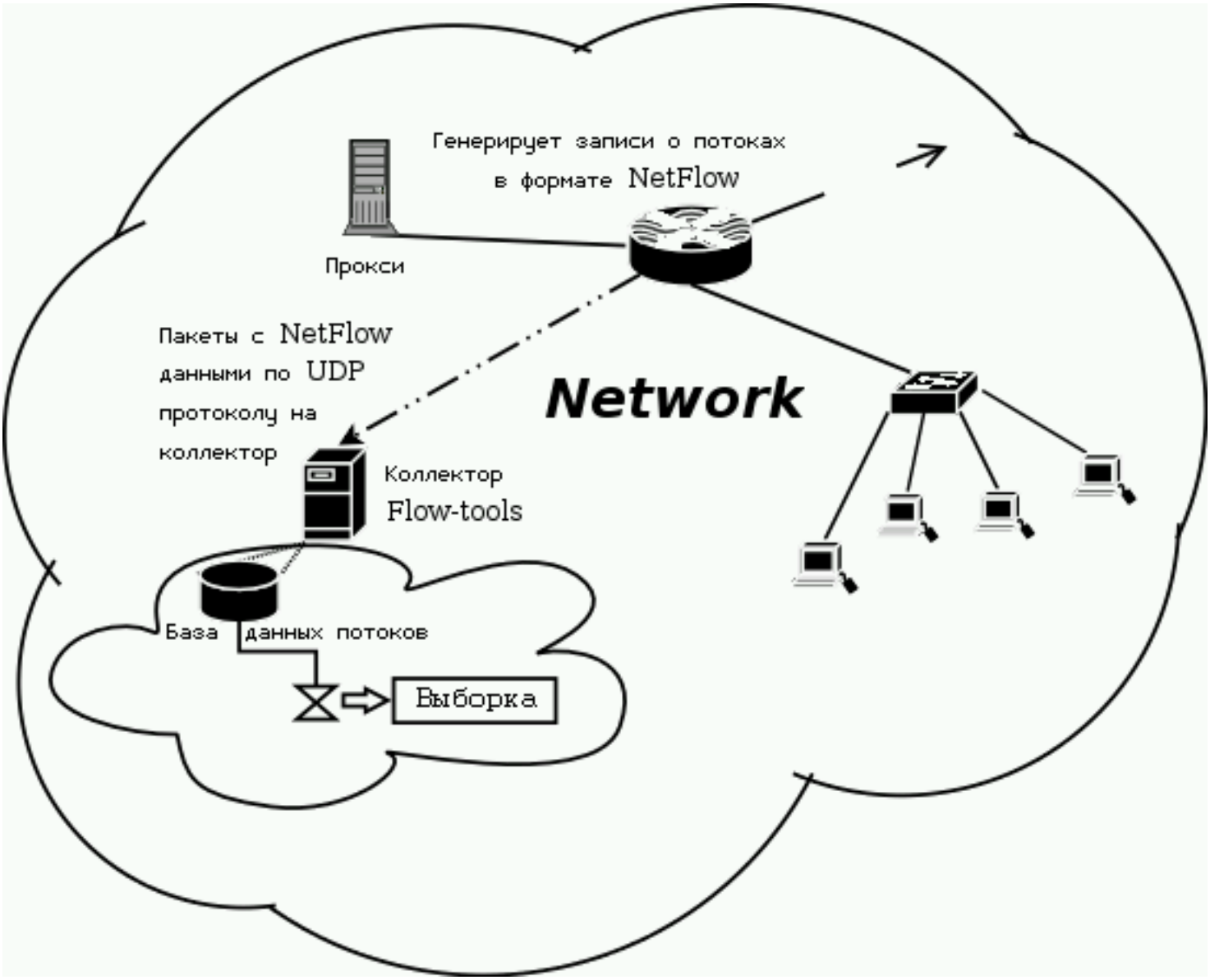

Рис. 1.1: Диаграмма предметной области.

1.4 Описание назначения программной системы.

Программная система, разрабатываемая нашей группой, является третьим компонентом в выше описанной схеме анализе трафика, то есть призвана взять на себя роль обработки и выборки записей из базы потоков. Основной причиной выбора данного направления работы является необходимость ускорения операций производимых на этом этапе за счёт использования более эффективной организации данных.

1.4.1 Классическая схема обработки базы потоков.

Записи о потоках хранятся в виде файлов, каждый из которых соответствует определённому периоду. В этих файлах записи хранятся в порядке их получения. Обычно при выборке эти файлы обрабатываются фильтром последовательно, что неэффективно с точки зрения скорости, так как для получения результатов необходим полная проверка каждой записи.

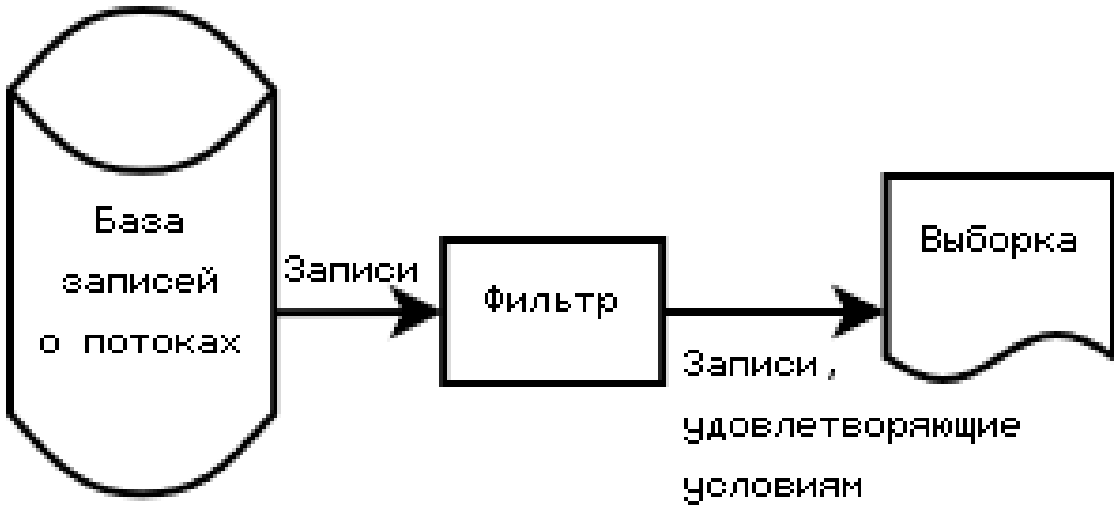

Рис. 1.2: Классическая схема обработки базы потоков.

1.4.2 Предлагаемая схема работы.

С целью ускорения процесса выборки необходимо организовать индексный файл, поиск по которому будет занимать меньше времени. Таким образом процесс выборки будет разбит на два этапа: построение индексного файла, осуществление выборки по условиям с помощью индексного файла и извлечение необходимых записей из базы потоков. Рассмотрим базу потоков как базу данных сверхбольшого размера на внешнем запоминающем устройстве. Записи содержат множество

разнотипных полей. Эффективность поиска основана на организации индексного файла как структуры, представляющей из себя древовидный словарь сортировки, построенный с использованием полей записи в качестве ключей. Обход данного словаря позволит исключить часть записей из обработки.

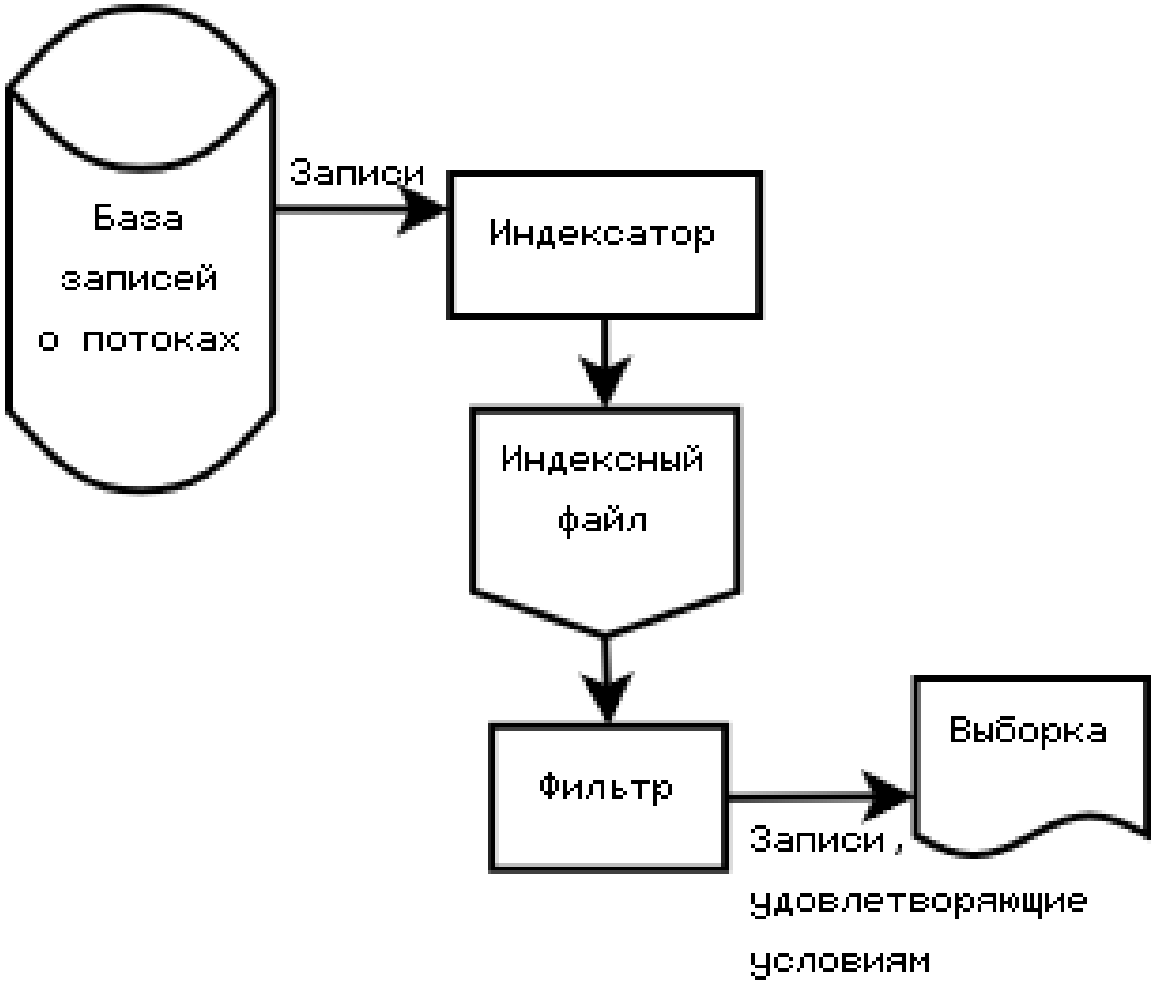

Рис. 1.3: Предлагаемая схема работы.

Глава 2

Первичный список требований к программной системе.

2.1 Функциональные требования:

- F1. Программная система должна реализовывать построение индекса по файлам, содержащим информацию о потоках.
- F2. Программная система должна производить поиск записей базы данных по построенному индексному файлу.
- F3. Поиск должен осуществляться по точному значению полей и в ограниченной области значений.
- F4. Параметры поиска задаются в файле условий.
- F5. Индексация должна происходить по всем хранимым полям записи. Желательно наличие возможности построения дополнительных индексов по отдельным полям записи, что позволит ускорить обработку запросов фильтрации.
- F6. Скорость осуществления операций поиска должна хотя бы на порядок превышать скорости работы существующих на данный момент решений. (В первую очередь flow-tools)
- F7. Программная система должна осуществлять индексацию базы потоков в формате NetFlow v5, созданной коллектором из набора flow-tools, с возможностью простого расширения для поддержки других форматов.

2.2 Ограничения:

- C1. Оформление кода программной системы и документации к ней должно соответствовать стилям, принятым при разработке проектов GNU.
- C2. Программная система должна быть реализована в виде набора утилит, написанных на языке C в соответствии со стандартом ANSI. (Сборка должна осуществляться gcc с флагами -pedantic -ansi.)
- C3. Система должна быть достаточно гибкой для её расширения в будущем с целью поддержки формата NetFlow v9.
- C4. Использование программной системы должно быть описано на странице справочного руководства ('man page' в стиле, принятом в операционных системах Unix), представленного на двух языках: английском и русском.
- C5. Комментарии в исходном коде программы должны быть на английском.
- C6. Документация проекта, представляемая для аттестации должна быть представлена в формате TeX.
- C7. Время на разработку программной системы: 15 недель.
- C8. Количество разработчиков: 7 человек.

Глава 3

Модель требований.

Программная система должна реализовывать построение индексного файла по файлам, содержащим информацию о потоках в формате NetFlow v5 и осуществлять фильтрацию на основе построенного справочника.

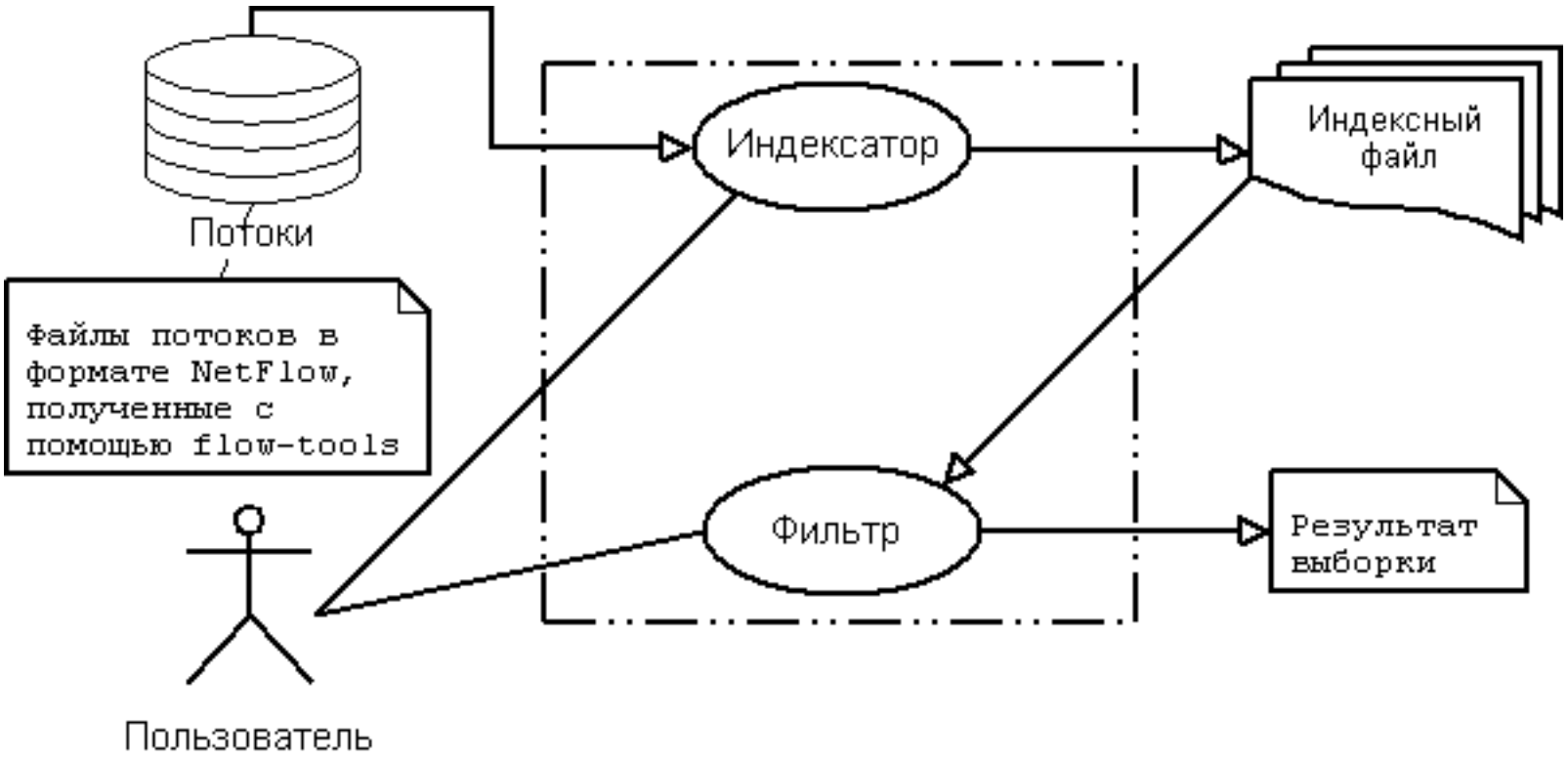

Рис. 3.1: Модель требований.

Разрабатываемая система представляет из себя два модуля: индексатор и фильтр. Внешние объекты - пользователь, файлы, хранящие потоки в формате NetFlow v5, индексный файл, результаты выборки.

Пользователь - администратор системы, должен иметь представление о потоках NetFlow.

Разрабатываемая система состоит из двух основных частей-функций:

1) Создание индексного файла - справочника, на основе файлов flow-tools с записями о всех потоках (использование исходного кода из пакета flow-tools).

2) Анализ условий и фильтрация данных о потоках на основе индексного файла.

Последовательность операций, которые необходимо провести для получения выборки из имеющихся данных (полный сценарий использования 3.2):

1) Запуск индексатора

2) Ввод данных - указание файлов базы потоков которые надо индексировать

3) Построение индексных файлов и завершение работы индексатора

4) Запуск фильтра

5) Ввод данных - указание критериев поиска

6) Получение выборки, вывод на экран или в файл

Возможно разделение основного сценария на два отдельных этапа:
Создание индексных файлов на основе базы данных о потоках (3.3).

Фильтрация потоков по заданным параметрам (3.4).

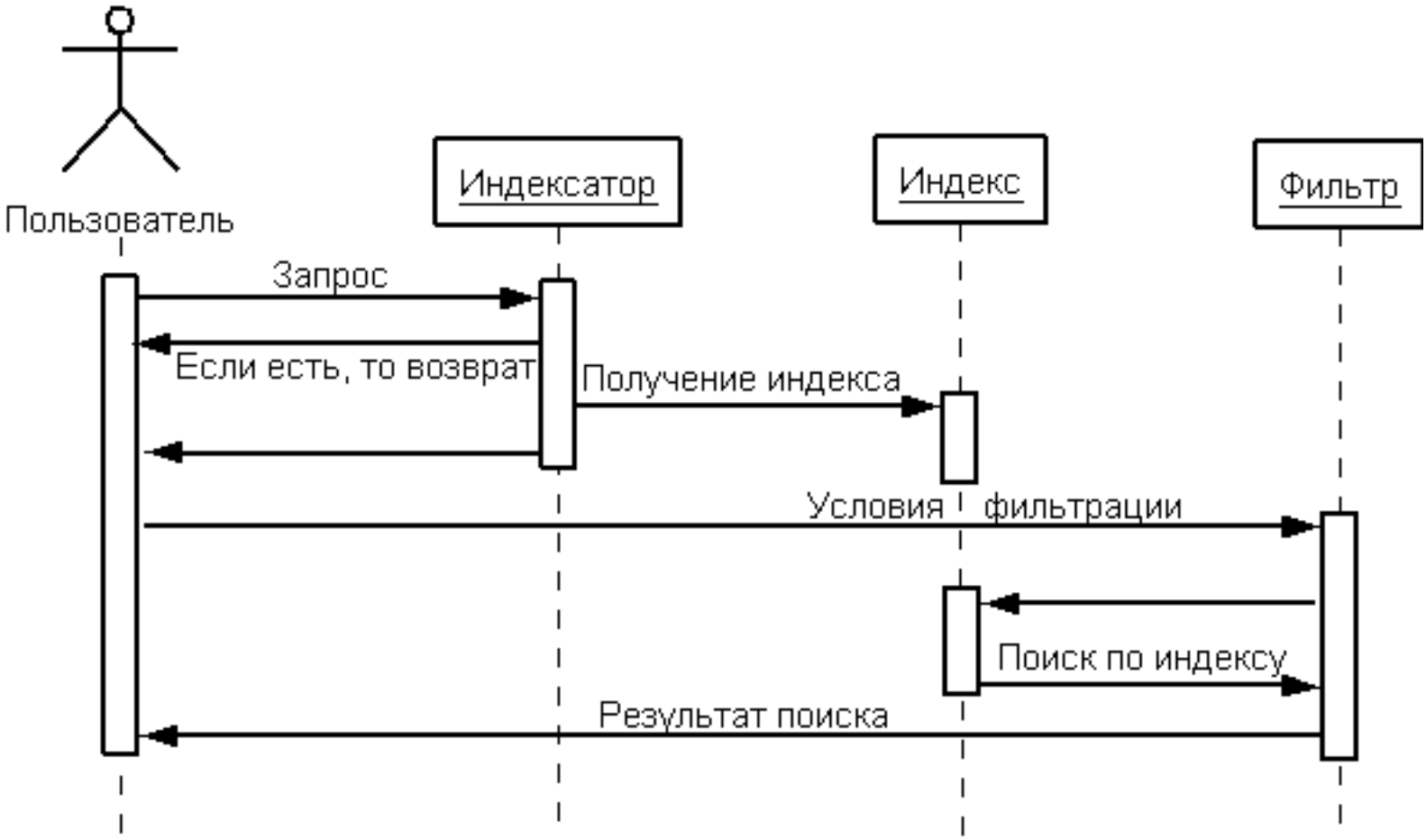

Рис. 3.2: Полный сценарий использования.

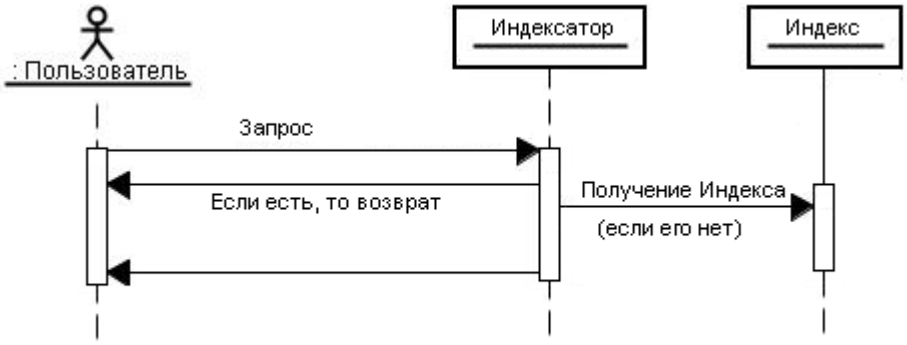

Рис. 3.3: Создание индексных файлов на основе базы данных о потоках.

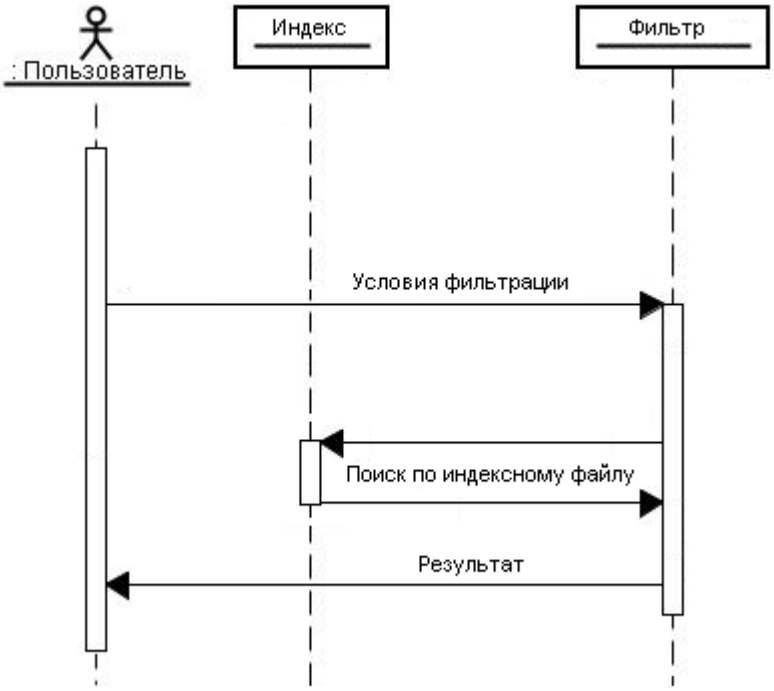

Рис. 3.4: Фильтрация потоков по заданным параметрам.

Глава 4

Архитектура программной системы.

Программная система представляет из себя две консольные утилиты:

- индексатор. (программа построения индексного файла)
- фильтр. (программа для осуществления выборки потоков по условиям из индексного файла)

Программная система использует (может использовать) при работе следующие файлы:

- файлы, содержащие индексируемые данные.
- индексный файл.
- шаблон, описывающий формат внутреннего представления записи в разрабатываемой программной системе.
- файл, описывающий условия поиска.
- файл для вывода результата выборки.

Подсистемы индексатора:

- чтения и анализа шаблона записи.
- доступа к базе потоков (отдельная для каждого поддерживаемого формата представления этой базы).

    1) данные могут быть представлены в нескольких файлах. (соблюдение временной непрерывности и последовательности данных не играет определяющей роли, т.к. для их хранения в индексном файле используются сбалансированные деревья и при добавлении записей, данные автоматически располагаются в правильном порядке)

2) максимальный размер файла определяется ограничениями файловой системы.

3) формат представления данных в обрабатываемых файлах соответствует принятому в flow-tools для хранения записей NetFlow v5.

4) последовательный доступ к данным.

- представления записи во внутренней структуре, не привязанной к формату базы потоков. (данная структура должна обеспечивать гибкое представление различных по характеру и количеству полей записей, однако, для каждого индексного файла добавляемые записи должны быть однородными)

  1) общий вид записи описывается шаблоном.

  2) шаблон для NetFlow v5 фиксирован.

- добавления записей в индексный файл.

  1) число записей может быть очень велико (миллионы).

  2) добавление осуществляется при однократной обработке данных (расширение индексного файла не предусмотрено).

Подсистемы фильтра:

- чтения условий поиска.

  1) условие может точно идентифицировать запись.

  2) условие может идентифицировать запись не по всем полям точными значениями.

  3) условие может идентифицировать запись по подмножеству полей, где для каждого поля возможно задание диапазона значений.

- доступа к индексному файлу.

- обхода индексного файла по условиям.

- представления записи во внутренней структуре.

- вывода выборки записей.

  1) вывод всех удовлетворяющих условиям записей.

Глава 5

Критерии аттестации системы.

- Тесты функциональности:
  1) Проверка способности осуществлять индексацию базы потоков в формате NetFlow v5.
  2) Проверка производительности в сравнении с утилитой flow-filter.
  3) Поиск информации по заданным для программы фильтра параметрам.
  4) Способность производить выборку данных о потоках по точному значению и ограниченной области.
  5) Формирование индексного файла по базе данных записей о потоках для поиска.
- Тесты на оговорённые в спецификации опциональные возможности:
  1) Возможности индексации по подмножеству полей записи.
  2) Возможность расширения ПС для обработки потоков в формате NetFlow v9.
- Тесты на поведение в нестандартных ситуациях:
  1) Проверка устойчивости программной системы к некорректным входным данным.
  2) Ввод неверных команд и параметров.
  3) Работа ПС с большими объёмами данных.

Глава 6

Глоссарий.

- Демон - определённая разновидность системных процессов. Демоны отличаются от обыкновенных процессов тем, что они не связаны с консолью. Обычно демоны выполняют специфические системные действия, например администрирование и управление в сетях.
- Индексный файл - файл, предназначенный для увеличения скорости доступа к данным, в котором содержится 1 или несколько полей из таблицы базы данных (индексные поля) и ссылки на записи в таблице базы данных, включающие эти поля.
- Коллектор - демон NetFlow, который слушает на UDP порту и осуществляет сбор информации от сенсора. Полученные данные он сбрасывает в файл для дальнейшей обработки.
- Пакет - блок данных, имеющий строго определённую структуру, включающую заголовок и поле данных. В Internet данные разбиваются на маленькие части, которые заключаются в пакеты; каждый пакет пересекает сеть отдельно от других.
- Поток - однонаправленная последовательность пакетов между отправителем и получателем в рамках одного сеанса. Поток идентифицируется по IP-адресам, портам источника и конечной точки, протоколу транспортного уровня, полю ToS (Type of Service) ip-пакета, номеру интерфейса, на котором был принят поток.
- Процесс (в ОС UNIX) - это программа, выполняемая в собственном виртуальном адресном пространстве. Когда пользователь входит в систему, автоматически создается процесс, в котором выполняется программа командного интерпретатора. Если командному интерпретатору встречается команда,

соответствующая выполняемому файлу, то он создает новый процесс и запускает в нем соответствующую программу, начиная с функции main.

- Сенсор - демон NetFlow, который слушает сеть и фиксирует данные сеанса.
- Система обработки - компонент NetFlow, который читает файлы, созданные коллектором и генерирует отчёты в форме, более удобной для человека. Эта система должна быть совместима с форматом данных, предоставляемых коллектором.
- Файлы NetFlow - файлы, которые создает коллектор. Каждая строка этого файла отображает информацию о потоках трафика на уровне сеансов. Она включает в себя:
  - адрес источника и назначения;
  - тип транзакции;
  - IP-адрес отправителя и получателя;
  - номера портов протоколов TCP и UDP для приложений отправителя и получателя;
  - тип протокола (TCP, UDP и т. д.);
  - тип сервиса (Type of Service) - используется для группировки приложения по классам. Наиболее часто используется при передаче голоса и видео, которые передаются с использованием UDP протокола. Без маркировки данного трафика с помощью поля ToS, его не возможно отследить.
  - номер входящего физического интерфейса - технология NetFlow позволяет учитывать только входящий трафик. Тем не менее, ReporterAnalayzer использует свои собственные алгоритмы, получивший название "OUT calculation"с помощью которого можно получить полную статистику в обоих направлениях (IN и OUT) для всех интерфейсов, мониторинг которых осуществляется.
- Фильтр - утилита, позволяющая делать выборки записей из файлов NetFlow по определённым критериям (например, flow-nfilter).
- Netflow - система учета трафика, которая имеет три основных компонента: сенсор, коллектор и систему отображения данных.