

## Содержание

Введение.....	6
1 Анализ инфраструктуры предприятия.....	8
2 Разработка технического задания на проектирование ЛВС.....	9
3 Анализ условия размещения компьютеров и другого оборудования в помещениях предприятия.....	11
4 Расчет конфигурации сети .....	12
5 Анализ условий размещения физической среды в помещениях предприятия и между ними.....	15
6 Составление структуры физической среды сети .....	17
6.1 Выбор типа кабеля для горизонтальных подсистем .....	17
6.2 Структура горизонтальной кабельной подсистемы .....	18
6.3 Выбор типа кабеля для вертикальных подсистем.....	19
6.4 Структура вертикальной кабельной подсистемы .....	19
6.5 Выбор типа кабеля для подсистемы кампуса .....	20
6.6 Структура кабельной подсистемы для кампуса .....	21
6.7 Выбор типа кабеля в зависимости от требований помехозащищенности .....	21
6.8 Выбор типа кабеля с точки зрения секретности передаваемой информации .....	22
6.9 Выбор типа кабеля в зависимости от требований пожарной безопасности.....	23
6.10 Выбор типа кабеля в зависимости от требований допустимой дальности передачи информации .....	23
6.11 Выбор типа кабеля в зависимости от его стоимости .....	23
7 Выбор активного сетевого оборудования сети.....	25
7.1 Выбор типовых сетевых адаптеров.....	25
7.2 Выбор типов репитеров.....	27

7.3 Выбор типов коммутаторов .....	27
7.4 Выбор типов концентраторов .....	29
8 Выбор пассивного оборудования .....	30
9 Выбор компьютеров и другого оборудования .....	31
9.1 Выбор компьютеров и другого оборудования в зависимости от требований производительности и запоминающих возможностей .....	31
9.2 Выбор компьютеров и другого оборудования в зависимости от их стоимости .....	32
10 Выбор сетевой операционной системы .....	33
11 Выбор протоколов сети .....	35
12 Методы передачи данных в сети .....	37
13 Разработка программного обеспечения «TWITTER-клиент» .....	39
13.1 Обзор аналогов программных средств .....	39
13.2 Постановка задачи .....	42
13.3 Разработка алгоритма и программного средства (ПС) .....	42
13.4 Разработка алгоритма работы ПС .....	43
13.5 Разработка спецификации требований к ПС.....	43
13.6 Разработка структурной схемы ПС.....	43
13.7 Обоснование выбора языка и среды программирования.....	44
13.8 Программная реализация разработанного алгоритма работы ПС (описание основных функций) .....	46
13.9 Тестирование и отладка ПС .....	46
13.10 Руководство пользователя .....	47
14 Методы и средства защиты информации в сети .....	49
Заключение .....	51
Список использованных источников .....	52
Приложение А .....	53

## **Введение**

Современная эпоха характеризуется стремительным процессом информатизации общества. Это можно обеспечить с помощью объединения компьютеров в сети, что позволяет получать и передавать сообщения прямо на рабочем месте, практически мгновенно получать необходимые сведения о любом событии из любой точки земного шара. Локальные вычислительные сети (ЛВС) являются важным компонентом в таком массовом и масштабном обмене информацией.

Под ЛВС понимают совместное подключение нескольких отдельных компьютерных рабочих мест (рабочих станций) к единому каналу передачи данных. Благодаря такому объединению мы получили возможность одновременного использования программ и баз данных несколькими пользователями. Необходимо постоянно поддерживать сети в состоянии, соответствующем требованиям времени, внедрять разнообразные современные технические, технологические и организационные новшества.

Создание сети основывается на нахождении компромисса между потребностями предприятия, его финансовыми возможностями и возможностями сетевых и информационных технологий сегодня и в будущем.

От правильно спроектированной и реализованной корпоративной сети, выбора надежного и производительного оборудования напрямую зависит работоспособность информационной системы в целом, возможность ее эффективной и длительной эксплуатации, модернизации и адаптации к меняющимся задачам.

Комплекс аппаратно–программных средств сети может быть описан многоуровневой моделью.

В основе любой сети лежит аппаратный слой, который включает компьютеры различных классов. Набор компьютеров в сети должен соответствовать набору разнообразных задач, решаемых сетью.

Второй слой составляет разнообразное сетевое оборудование, необходимое для создания локально-вычислительных сетей, и коммуникационное оборудование для связи с глобальными сетями. Коммуникационные устройства играют не менее важную роль, чем компьютеры, которые являются основными элементами по обработке данных.

Третьим слоем являются операционные системы, которые составляют программную основу сети. При построении сетевой структуры важно учитывать насколько эффективно данная операционная система может взаимодействовать с другими операционными системами сети, насколько она способна обеспечить безопасность и защиту данных и т. д.

Самым верхним слоем сетевых средств являются различные сетевые приложения, такие как сетевые базы данных, почтовые системы, средства архивирования данных и др. Важно знать совместимость различных сетевых приложений.

Использование сети приводит к совершенствованию коммуникаций, т.е. к улучшению процесса обмена информацией и взаимодействия между сотрудниками предприятия, а также его клиентами и поставщиками. Сети снижают потребность предприятий в других формах передачи информации, таких как телефон или обычная почта. Зачастую вычислительные сети на предприятии развёртываются из-за возможности организации электронной почты.

Главным доказательством эффективности вычислительных сетей является бесспорный факт их повсеместного распространения. Всё больше и больше появляются крупные сети с сотнями рабочих станций и десятками серверов.

## **1 Анализ инфраструктуры предприятия**

Предприятие - учреждение образования, состоящее из трех учебно-лабораторных корпусов. В каждом корпусе необходимо поместить соответствующее оборудование для эффективного функционирования сети. Первый корпус должен содержать пять компьютерных класса, каждый прямоугольной формы без выступов и площадью  $S=5*10=50(\text{м}^2)$ . Второй корпус должен иметь три компьютерных класса без выступов, два из которых имеют каждый площадь  $S=6*11=66(\text{м}^2)$ , а третий -  $S=4*7=28(\text{м}^2)$ . В третьем корпусе два компьютерных класса, площадь каждого составляет  $S=7*12=84(\text{м}^2)$ , в каждом корпусе все компьютерные классы — смежные, расположены посередине корпуса. Расстояние от крайнего компьютерного класса до боковой несущей стены каждого корпуса одинаково, равно  $l=10\text{м}$ . корпуса расположены последовательно. Между ними расстояние одинаково и составляет  $l=15\text{м}$ . Все классы крайних корпусов расположены на первом этаже.

В качестве сетевой технологии использовать архитектуру FDDI.

## **2 Разработка технического задания на проектирование ЛВС**

Для создания сети ЛВС здания необходимо в первую очередь подробно исследовать план здания и сделать выводы о возможностях прокладки сети. При анализе важно обращать внимание на размеры помещений, особенностей здания. С учетом этих условий, а также исходя из предъявляемых к сети требований, складывается представление о топологии будущей сети. Планирование проекта ЛВС начинается с предварительного выбора базовой сетевой технологии для проектируемой локальной сети на основании технических требований, экспертных данных и теоретического материала.

Исходя из данных к курсовому проекту, разработаем техническое задание на проектирование ЛВС. Необходимо разработать локальную сеть, которая предназначена для обмена файлами в пределах учебных лабораторий без совместного выхода в глобальную сеть Internet. Передаваемая в ЛВС информация является общедоступной. В качестве сетевых технологий будем использовать архитектуру FDDI.

FDDI (Fiber Distributed Data Interface) — стандартизованная спецификация ANSI X3T9.5 для сетевой архитектуры высокоскоростной передачи данных по оптоволоконным линиям. Скорость передачи — 100 Мбит/с. Топология — кольцо (двойное), возможна гибридная: включение звездообразных или древовидных подсетей в главную сеть через концентратор.

Элементами конфигурации сети будут являться компьютеры, коммутаторы, кабельные соединения между ними. В каждом классе размещены компьютеры и сетевые розетки. В специальном помещении имеется шкаф для сетевого оборудования, в котором стоит коммутатор корпуса. Коммутаторы соединяются с коммутаторами других корпусов через сервер корпуса. Во втором корпусе находится маршрутизатор, к которому подключаются другие серверы корпусов. Маршрутизатор обеспечит соединение внутренних сегментов сети, а также обеспечит выход во внешнюю среду. В результате меж-

ду любыми двумя компьютерами сети число коммутаторов не будет превышать четырех.

Перед тем, как приступить непосредственно к реализации сети, важно рассчитать нагрузку на сеть, используя созданный план сети, и в соответствии с этим выбрать все необходимое оборудование, которое будет наиболее оптимальным по соотношению цены и качества в условиях данной сети.

### **3 Анализ условия размещения компьютеров и другого оборудования в помещениях предприятия**

Проанализировав исходные данные к проекту, можно сделать вывод, что в помещениях, в которых предполагается размещение компьютеров и другого оборудования, отсутствуют электромагнитные помехи, источники высоких температур, и другие неблагоприятные факторы, способные привести к выходу из строя оборудования.

Первый корпус должен содержать пять компьютерных классов, прямоугольной формы без выступов и площадью  $S = 50 \text{ (м}^2\text{)}$ . В этих кабинетах будет находиться по 10 компьютеров.

Второй корпус должен иметь два компьютерных класса без выступов  $S = 66 \text{ (м}^2\text{)}$  по 10 компьютеров и один класс  $S = 28 \text{ (м}^2\text{)}$  на 6 компьютеров.

В третьем корпусе два компьютерных класса, площадь каждого составляет  $S = 84 \text{ (м}^2\text{)}$  будет находиться по 8 компьютеров. В каждом корпусе все компьютерные классы - смежные, расположены посередине корпуса.

Расстояние от крайнего компьютерного класса до боковой несущей стены каждого корпуса одинаково, равно  $l=10\text{м}$ . корпуса расположены последовательно. Между ними расстояние одинаково и составляет  $L_k=15\text{м}$ . Все компьютерные классы находятся на первом этаже.

Задача проекта спроектировать сеть в 3 корпусах. В каждом из корпусов находится по несколько компьютерных классов. Компьютеры в классах расположены вдоль стен.



## 4 Расчет конфигурации сети

Топология связей в FDDI устроена таким образом, что отказ в любом из узлов из-за выхода из строя оборудования или отключения питания не приведет к разрыву кольца, поток кадров автоматически пойдет в обход поврежденного участка.

Спецификация FDDI определяет принципы построения интерфейса, который обеспечивает передачу данных со скоростью 100 Мбит/сек. Для обеспечения большей надежности каналы передачи данных в сети FDDI дублируются, таким образом, в этой сети могут быть использованы два информационных кольца – основное (primary ring) и дополнительное (secondary ring).

В зависимости от способа подключения, станции, которые подключены к сети FDDI, могут быть двух типов:

- Станции класса А (станции двойного подключения DAS)
- Станции класса В (станции одинарного подключения SAS)

В данном случае применяется топология кольца, схема представлена на рисунке 4.1.

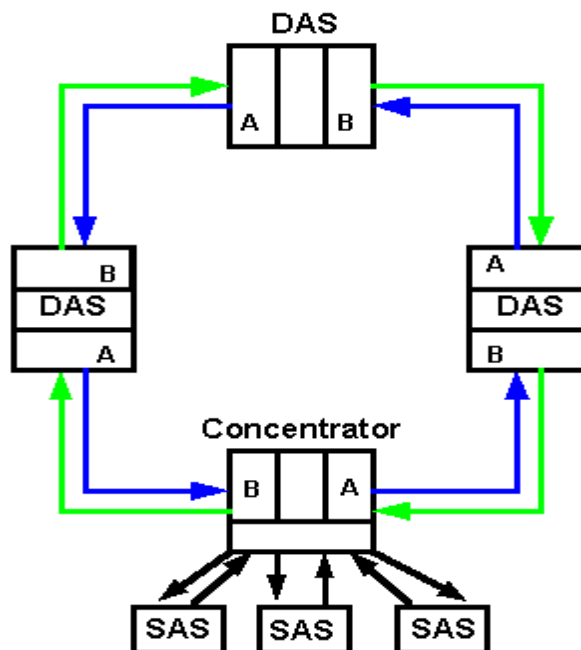


Рис. 4.1 Схема построения сети FDDI

Использование в качестве среды распространения волоконной оптики позволяет существенно расширить полосу пропускания кабеля и увеличить расстояния между сетевыми устройствами.

Метод доступа — маркерный с возможностью одновременного циркулирования множества кадров в кольце. Максимальное количество станций в сети — до 1000, расстояние между станциями до 2 км при многомодовом и до 45 км при одномодовом кабеле (затухание сигнала между станциями до 11 дБ), длина кольца до 100 км (может увеличиваться за счет применения более мощной аппаратуры). В некоторых случаях вторичное кольцо используется для повышения пропускной способности потенциально до 200 Мбит/с.

Детерминированная природа протокола FDDI (возможность предсказания максимальной задержки при передаче пакета по сети и возможность обеспечить гарантированную полосу пропускания для каждой из станций) делает его идеальным для использования в сетевых АСУ в реальном времени и в приложениях, критичных ко времени передачи информации (например, для передачи видео и звуковой информации).

Станции, подключенные к сети FDDI, подразделяются на две категории.

1. Станции класса А имеют физические подключения к первичному и вторичному кольцам (Dual Attached Station - двукратно подключенная станция);

2. Станции класса В имеют подключение только к первичному кольцу (Single Attached Station - однократно подключенная станция) и подключается только через специальные устройства, называемые концентраторами.

Порты сетевых устройств, подключаемых к сети FDDI, классифицируются на 4 категории: А порты, В порты, М порты и S порты. Портом А называется порт, принимающий данные из первичного кольца и передающий их во вторичное кольцо. Порт В - это порт, принимающий данные из вторичного кольца и передающий их в первичное кольцо. М (Master) и S (Slave) порт пе-

редают и принимают данные с одного и того же кольца. М порт используется на концентраторе для подключения Single Attached Station через S порт.

## **5 Анализ условий размещения физической среды в помещениях предприятия и между ними**

Как было сказано выше, в помещениях отсутствуют источники сильных электромагнитных помех, агрессивные среды, повышенная влажность и источники высоких температур, что позволяет не использовать для кабелей дополнительной защитной изоляции и предположить, что в таких условиях кабель прослужит долгое время. В качестве физической среды будет использоваться витая пара.

Так как компьютеры внутри классов стоят вдоль стен, то физическая среда также будет проходить вдоль стен. Удобнее всего заложить физическую среду в короба из ПВХ. Данный способ размещения физической среды хорош тем, что позволяет обойтись малыми затратами, а также эстетичным видом. Возле каждого компьютера будет выведена розетка. Розетки располагаются в аудиториях - рядами вдоль стен, либо по полу параллельно рядам рабочих мест в больших аудиториях. В других помещениях розетки располагаются вдоль стен или загоронок в количестве, зависящем не только от количества рабочих мест, но и от размеров помещения.

Прокладка физической среды передачи данных между учебными корпусами возможна по воздуху, под землей, по столбам телефонных линий. Достоинство первого способа мобильность, недостатки малая помехозащищенность (при больших помехах необходимо использовать оптоволокно или экранированные кабели), малая надежность, подверженность воздействиям окружающей среды (из-за чего защитная оболочка быстрее теряет свои защитные качества). Достоинство второго способа более надежен по сравнению с первым, недостаток вследствие близкого расположения к линиям связи, возможно искажение сигнала электромагнитным полем (при использовании оптоволокна этот недостаток исчезает). Достоинство третьего способа - менее подвержен воздействию окружающей среды, недостатки меньшая мобильность; возможно воздействие на защитную оболочку кабеля агрессивных

сред при попадании их в землю, если кабель недостаточно заглублен; не исключено воздействие электромагнитных полей, при прохождении рядом линий троллейбусов или трамваев (этот же недостаток может проявляться и в других способах).

Кабель будем улаживать в специальные, предусмотренные конструкцией здания, каналы. Каналы выполнены в виде труб различного сечения и предназначены для прокладки проводов различных коммуникаций.

## **6 Составление структуры физической среды сети**

В качестве сетевой технологии используется архитектура Fiber Distributed Data Interface, оптоволоконный интерфейс распределенных данных - это первая технология локальных сетей, в которой средой передачи данных является волоконно-оптический кабель.

Протокол рассчитан на физическую скорость передачи информации 100Мбит/с и предназначен для сетей с суммарной длиной до 100км

Топология связей в FDDI устроена таким образом, что отказ в любом из узлов из-за выхода из строя оборудования или отключения питания не приведет к разрыву кольца, поток кадров автоматически пойдет в обход поврежденного участка.

FDDI позволяет работать с кадрами размером 4500 октетов, за вычетом места, занимаемого преамбулой, остается 4470 октетов для передачи данных. RFC-1188 резервирует 256 октетов для заголовков, оставляя для данных 4096 октетов. Маршрутизатор, поддерживающий протокол FDDI должен быть способен принимать такие длинные пакеты. Посылаться же должны дейтограммы не длиннее 576 октетов, если не ясно, сможет ли адресат принимать длинные кадры.

### **6.1 Выбор типа кабеля для горизонтальных подсистем**

Горизонтальная кабельная система начинается телекоммуникационной розеткой на рабочем месте она включает в себя: розетку, горизонтальный кабель, точки терминирования и пэтч-корды (кроссировочные перемычки), представляющие собой горизонтальный кросс.

Максимальная длина горизонтального кабеля должна составлять 90 м, независимо от типа среды. Она измеряется от разъема (панели) в телекоммуникационном шкафу этажа до информационного разъема на рабочем месте.

Максимальная механическая длина абонентских, коммутационных (перемычек) и сетевых кабелей - не более 10 метров.

Стоимость установки сетей на оптоволоконном кабеле для горизонтальной подсистемы оказывается весьма высокой. Эта стоимость складывается из стоимости сетевых адаптеров и стоимости монтажных работ, которая в случае оптоволокна гораздо выше, чем при работе с другими видами кабеля.

Преобладающим кабелем для горизонтальной подсистемы является неэкранированная витая пара категории 5, но, если нужна высокая степень защищенности данных, высокая пропускная способность или устойчивость к электромагнитным помехам, волоконно-оптический кабель может использоваться в горизонтальных подсистемах.

## **6.2 Структура горизонтальной кабельной подсистемы**

Структуру кабельной системы этажа и здания характеризуется очень большим количеством ответвлений кабеля, так как его нужно провести к каждой пользовательской розетке, причем и в тех комнатах, где пока компьютеры в сеть не объединяются. Поэтому к кабелю, используемому в горизонтальной проводке, предъявляются два основных требования: удобство прокладки и удобство выполнения ответвлений.

Горизонтальная подсистема - наиболее «подвижная» часть любой локальной сети. Добавление новых пользователей, перемещение рабочих группы происходят гораздо чаще, чем изменения в вертикальных подсистемах между этажами. Кроме того, эта часть сети является самой длинной, несмотря на множество относительно коротких отрезков и содержит большое количество соединительных разъемов. Именно по этим причинам оборудование должно быть максимально надежно и просто в эксплуатации, но, вместе с тем, оказывает весьма существенное влияние на конечную стоимость всей сети.

При выборе кабеля принимаются во внимание следующие характеристики: полоса пропускания, расстояние, физическая защищенность, электромагнитная помехозащищенность, стоимость. Кроме того, при выборе кабеля нужно учитывать, какая кабельная система уже установлена на предприятии, а также какие тенденции и перспективы существуют на рынке в данный момент.

Горизонтальная подсистема - наиболее «подвижная» часть любой локальной сети. Добавление новых пользователей, перемещение рабочих группы происходят гораздо чаще, чем изменения в вертикальных подсистемах между этажами.

### **6.3 Выбор типа кабеля для вертикальных подсистем**

Кабельная система является фундаментом любой сети. Вертикальная система является, по сути, магистралью, соединяющей этажи здания. К кабелю, используемому в ней, предъявляются соответствующие требования: он должен передавать данные на большие расстояния и с более высокой скоростью по сравнению с кабелем горизонтальной подсистемы.

В общем случае для вертикальной подсистемы оптоволокно показывает отличные характеристики пропускной способности, расстояния и защиты данных; устойчивость к электромагнитным помехам; может передавать голос, видеоизображение и данные. Это, безусловно, самый привлекательный вариант.

### **6.4 Структура вертикальной кабельной подсистемы**

Кабельная система является фундаментом любой сети. Как при строительстве нельзя создать хороший дом на плохо построенном фундаменте, так и сеть, отлично работающая на плохой кабельной системе, - это явление из области ненаучной фантастики. Если в кабелях ежедневно происходят корот-



кие замыкания, контакты разъемов то отходят, то снова входят в плотное соединение, добавление новой станции приводит к необходимости тестирования десятка контактов разъемов из-за того, что документация на физические соединения не ведется, то ясно, что на основе такой кабельной системы любое, самое современное и производительное оборудование будет работать плохо. Пользователи будут недовольны большими периодами простоев и низкой производительностью сети. Причем проблем с кабельной системой становится намного больше при увеличении размеров сети.

Ответом на высокие требования к качеству кабельной системы стали структурированные кабельные системы.

### **6.5 Выбор типа кабеля для подсистемы кампуса**

Как и для вертикальных подсистем, оптоволоконный кабель является наилучшим выбором для подсистем нескольких зданий, расположенных в радиусе нескольких километров. Для этих подсистем также подходит толстый коаксиальный кабель. При выборе кабеля для кампуса нужно учитывать воздействие среды на кабель вне помещения. Для предотвращения поражения молнией лучше выбрать для внешней проводки неметаллический оптоволоконный кабель. При подземной прокладке кабель должен иметь специальную влагозащитную оболочку (от дождя и подземной влаги), а также металлический защитный слой от грызунов и вандалов. Влагозащитный кабель имеет прослойку из инертного газа между диэлектриком, экраном и внешней оболочкой.

Кабель для внешней прокладки не подходит для прокладки внутри зданий, так как он выделяет при сгорании большое количество дыма.

## **6.6 Структура кабельной подсистемы для кампуса**

Сети кампусов объединяют множество сетей различных отделов одного предприятия в пределах отдельного здания или в пределах одной территории, покрывающей площадь в несколько квадратных километров. При этом глобальные соединения в сетях кампусов не используются. Службы такой сети включают взаимодействие между сетями отделов, доступ к общим базам данных предприятия, доступ к общим факс-серверам, высокоскоростным модемам и высокоскоростным принтерам. В результате сотрудники каждого отдела предприятия получают доступ к некоторым файлам и ресурсам сетей других отделов. Важной службой, предоставляемой сетями кампусов, стал доступ к корпоративным базам данных независимо от того, на каких типах компьютеров они располагаются

Кабель подсистемы кампуса проводится на уровне потолка первого этажа. Так как кабель имеет собственный трос внутри, не требуется никаких дополнительных устройств для предотвращения обрыва кабеля.

## **6.7 Выбор типа кабеля в зависимости от требований помехозащищенности**

Оптоволоконный кабель обладает исключительными характеристиками по помехозащищенности. Никакие внешние электромагнитные помехи в принципе не способны исказить световой сигнал, а сам этот сигнал принципиально не порождает внешних электромагнитных излучений.

Типичная величина затухания сигнала в оптоволоконных кабелях на частотах, используемых в локальных сетях, составляет около 5 дБ/км, что примерно соответствует показателям электрических кабелей на низких частотах. Но в случае оптоволоконного кабеля при росте частоты передаваемого сигнала затухание увеличивается очень незначительно, и на больших ча-

стотах (особенно свыше 200 МГц) его преимущества перед электрическим кабелем неоспоримы, он просто не имеет конкурентов.

Оптоволоконный кабель менее прочен, чем электрический, и менее гибкий (типичная величина допустимого радиуса изгиба составляет около 10—20 см). Чувствителен он и к ионизирующим излучениям, из-за которых снижается прозрачность стекловолокна, то есть увеличивается затухание сигнала. Чувствителен он также к резким перепадам температуры, в результате которых стекловолокно может треснуть. В настоящее время выпускаются оптические кабели из радиационно стойкого стекла (стоят они, естественно, дороже).

Оптоволоконные кабели чувствительны также к механическим воздействиям (удары, ультразвук) — так называемый микрофонный эффект. Для его уменьшения используют мягкие звукопоглощающие оболочки.

## **6.8 Выбор типа кабеля с точки зрения секретности передаваемой информации**

Кабельная система не содержит активных или нелинейных элементов, поэтому сама по себе она не может быть источником побочных излучений. Однако кабельная система связывает между собой все элементы компьютерной сети. По ней передаются сетевые данные, но вместе с этим она является также приемником всех наводок и средой для переноса побочных электромагнитных излучений.

Современные кабели для локальных сетей имеют очень низкий уровень излучения передаваемых сигналов. В этих кабелях сигналы передаются по витой паре проводов, причем количество скруток на единицу длины, строго постоянно. Благодаря этому витая пара получается очень хорошо сбалансированной. В принципе такая система вообще не должна излучать. Более того, наличие экрана у витой пары мало влияет на уровень излучения сигналов, передаваемых по витой паре. Уровень излучения сигналов, передаваемых по

кабелю, остается довольно низким, особенно для кабелей категории выше пятой. Реально на расстоянии буквально единиц метров уже невозможно по электромагнитному излучению современного кабеля перехватить передаваемую по нему информацию

### **6.9 Выбор типа кабеля в зависимости от требований пожарной безопасности**

Кабель прокладываемый в помещениях будет укладываться в пластиковые короба. В случае его воспламенения его будет легко потушить средствами пожара тушения.

### **6.10 Выбор типа кабеля в зависимости от требований допустимой дальности передачи информации**

В качестве базовой кабельной системы определяет многомодовый волоконно-оптический кабель. Максимальная длина кабеля для одномодового волокна равна 2000 м. Предельное расстояние получается небольшим - 550 м. Это связано с особенностями распространения когерентного света в широком канале многомодового кабеля. Для присоединения лазерного трансивера к многомодовому кабелю необходимо использовать специальный адаптер. Для нашей сети этого расстояния вполне достаточно.

### **6.11 Выбор типа кабеля в зависимости от его стоимости**

Для создания сети, обычно, требуется соединить между собой компьютеры и сетевое оборудование. С этой целью используются кабели. Конечно, в настоящее время, помимо кабельных, существуют радиосети, в которых в качестве среды передачи данных выступают радиоволны, сети на основе лазерных технологий, но в данной статье речь о них не пойдет. Все же

наиболее распространенными остаются и в ближайшем будущем, видимо, останутся, сети, построенные на основе различного рода кабелей. Оценка стоимости кабелей представлена в таблице 6.1.

Таблица 6.1 – Оценка стоимости

<b>Вид кабеля</b>	<b>Стоимость</b>
Тонкий коаксиальный кабель	низкая
Толстый коаксиальный кабель	низкая
Витая пара	низкая
Экранированная витая пара	средняя
Волоконно-оптический	высокая
Одномодовый оптоволоконный	высокая

Стоимость установки сетей на оптоволоконном кабеле для горизонтальной подсистемы оказывается весьма высокой. Эта стоимость складывается из стоимости сетевых адаптеров (около тысячи долларов каждый) и стоимости монтажных работ, которая в случае оптоволокна гораздо выше, чем при работе с другими видами кабеля.

## **7 Выбор активного сетевого оборудования сети**

В нашей проектируемой сети будем использовать активное сетевое оборудование это сетевые адаптеры и коммутаторы.

Концентраторы вместе с сетевыми адаптерами, а также кабельной системой представляют тот минимум оборудования, с помощью которого можно создать локальную сеть. Такая сеть будет представлять собой общую разделяемую среду. Концентраторы и сетевые адаптеры позволяют строить небольшие базовые фрагменты сетей, которые затем должны объединяться друг с другом с помощью мостов, коммутаторов и маршрутизаторов.

Активное сетевое оборудование подразумевает под собой электронное устройство, за которым следуют некоторые "интеллектуальные" способности. К такому оборудованию, прежде всего можно отнести коммутаторы и маршрутизаторы. Коммутаторы представляют собой устройство, при помощи которого подключённые конечные пользователи сети, прежде всего, могут осуществлять обмен информацией. Так же коммутаторы применяются для соединения доменов коллизий локальной сети между собой.

Маршрутизатор - интеллектуальное устройство, работающее на третьем (сетевом) уровне сетевой модели OSI, предназначенное в основном для обеспечения выхода пользователей из локальной сети предприятия в глобальную сеть Интернет. Так же маршрутизаторы могут использоваться для организации распределённой сети компании, филиалы которой находятся на значительном удалении друг от друга и их не представляется возможным объединить физической линией связи.

### **7.1 Выбор типовых сетевых адаптеров**

Сетевые адаптеры предназначены для сопряжения сетевых устройств со средой передачи в соответствии с принятыми правилами обмена информацией. Сетевым устройством может быть компьютер пользователя, сетевой

сервер, рабочая станция и т.д. Набор выполняемых адаптером функций зависит от конкретного сетевого протокола. Ввиду того, что сетевой адаптер и в физическом, и в логическом смысле находится между устройством и сетевой средой, его функции можно разделить на функции сопряжения с сетевым устройством и функции обмена с сетью. Количественный и качественный состав функций сопряжения с сетевым устройством определяется его назначением и функциональной схемой.

Адаптеры FDDI могут использоваться на разнообразных рабочих и в устройствах межсетевого взаимодействия – мостах и маршрутизаторах. Существуют адаптеры FDDI, предназначенные для работы со всеми распространенными шинами. В сети FDDI такие устройства, как рабочие станции или мосты, подсоединяются к кольцу через адаптеры одного из двух типов: с двойным (DAS) или одиночным (SAS) подключением. Адаптеры DAS осуществляют физическое соединение устройств как с первичным, так и со вторичным кольцом, что повышает отказоустойчивость сети. Такой адаптер имеет два разъема (розетки) оптического интерфейса. Адаптеры SAS подключают рабочие станции к концентратору FDDI через одиночную оптоволоконную линию в звездообразной топологии. Эти адаптеры представляют собой плату, на которой наряду с электронными компонентами установлен оптический трансивер с разъемом (розеткой) оптического интерфейса.



Рис. 7.1 сетевой адаптер D-Link DGE-530T

Основные характеристики	
Производитель	D-Link
Тип	Сетевая карта
Максимальная скорость	1000 Мбит/сек
Скорость передачи данных	1000 Мбит/сек

## 7.2 Выбор типов репитеров

Репитеры (повторители), как уже отмечалось, ретранслируют приходящие на них (на их порты) сигналы, восстанавливают их амплитуду и форму, что позволяет увеличивать длину сети.

Конструктивно репитер может быть выполнен либо в виде отдельной конструкции со своим блоком питания, либо в виде платы, вставляемой в слот расширения материнской платы компьютера.

Один из недостатков встраиваемого в рабочую станцию репитера заключается в том, чтобы для обеспечения круглосуточной работы сети станция с репитером также должна работать круглосуточно. При выключении питания связь между сегментами сети будет нарушена.

Функции репитера заключаются в физическом разделении сегментов сети и обеспечении восстановления пакетов, передаваемых из одного сегмента сети в другой.

Репитер повышает надежность сети, так как отказ одного сегмента (например, обрыв кабеля) не сказывается на работе других сегментов. Однако через поврежденный сегмент данные проходить не могут.

Репитеры в данной сети отсутствуют.

## 7.3 Выбор типов коммутаторов

Коммутатор (switch) обеспечивает высокопроизводительную коммутацию в сетях с широким набором возможностей. Он представляет собой иде-



альное решение для сетей малого и среднего масштаба с растущими требованиями к пропускной способности. Он также прекрасно подходит для крупных предприятий, которым необходимы коммутационные решения Ethernet для центров обработки данных, серверных комплексов или кроссовых шкафов, сочетающие богатые функциональные возможности с приемлемой ценой. В настоящее время выпускаются различные по функциональности коммутаторы: с различным числом портов, дополнительными коммутационными модулями.

Для сети мы выбрали коммутатор с 16 портами 10/100Base-TX. Неуправляемый коммутатор 10/100 Мбит/с D-Link DES-1016A предназначен для повышения производительности работы относительно небольшой группы пользователей, обеспечивая при этом высокий уровень гибкости. Мощный и одновременно с этим простой в использовании, DES-1016A позволяет пользователям без труда подключить к любому порту сетевое оборудование, работающее на скоростях 10 Мбит/с или 100 Мбит/с, понизить время отклика и удовлетворить потребности в большой пропускной способности сети. Коммутатор снабжен 16 портами 10/100 Мбит/с, позволяющими рабочей группе гибко подключаться к сетям Ethernet и Fast Ethernet, а также интегрировать их. Это достигается благодаря свойству портов автоматически определять сетевую скорость, согласовывать стандарты 10Base-T и 100Base-TX, а также режим передачи полуполный дуплекс. Все порты поддерживают управление потоком 802.3х. Эта функция предотвращает пакеты от передачи, которая может привести к их потере, посредством передачи сигнала о возможном переполнении порта, буфер которого полон. Приостановка передачи пакетов продолжается до тех пор, пока буфер порта не будет готов принимать новые данные. Управление потоком реализовано для режимов полного и полудуплекса. Коммутатор может быть использован для непосредственного подключения компьютеров, так как обладает малой стоимостью подключения на порт. Это предотвращает возможность образования «узких мест» благодаря

предоставлению каждому компьютеру сети выделенной полосы пропускания.

Внешний вид коммутатора представлен на рисунке 7.2



Рис. 7.2 Коммутатор с 16 портами D-Link DES-1016A

#### **7.4 Выбор типов концентраторов**

Концентратор - сетевое устройство, предназначенное для объединения нескольких устройств Ethernet в общий сегмент сети. В настоящее время хабы почти не выпускаются — им на смену пришли сетевые коммутаторы (свитчи), выделяющие каждое подключённое устройство в отдельный сегмент. Концентраторы в данной сети отсутствуют.

## **8 Выбор пассивного оборудования**

Пассивное сетевое оборудование не относится непосредственно к созданию локальной вычислительной сети, но влияет на ее функциональность. Это кабели, короба, розетки, патч-панели, соединительные элементы и сопутствующее оборудование, которое не вносит изменения в сигналы на информационном уровне и не требует электропитания.

Розетки для каждого рабочего места выбираем двойные.

Патч-панели выбираются с расчетом на то, чтобы были отдельно на компьютерную часть сети и телефонную.

Патч-корды лучше использовать только заводского производства. Длины заводских патч-кордов бывают разные, выбирать следует исходя из необходимости конкретного соединения.

## **9 Выбор компьютеров и другого оборудования**

На сегодняшний день производители компьютерного оборудования представляют очень большой ассортимент своей продукции, и эта продукция рассчитана на широкий круг покупателей. Подбор оборудования осуществляется с учетом профессиональной деятельности сотрудников и в соответствии с системными требованиями ПО.

### **9.1 Выбор компьютеров и другого оборудования в зависимости от требований производительности и запоминающих возможностей**

При выборе рабочих станций, необходимо обратить внимание на то, каким быстродействием должен обладать компьютер пользователей. В данном случае в качестве рабочей станции нужно использовать машины с хорошей производительностью и высокой надежностью.

Рабочие станции будем собирать на основе чипсетов фирмы INTEL, как производительное и надежное решение для процессоров Intel Pentium(Celeron). Чипсеты от Intel отличаются высокой стабильностью, надежностью, функциональностью, производительностью. Тактовую частоту процессора выберем 2,2 ГГц, такая частота на данный момент способна справиться с современными офисными и другими ПО. Оперативная память выбирается равной 1 Гб, это вполне допустимый размер памяти, его достаточно для выполнения поставленных задач.

Жесткий диск из соображений экономии и оптимального варианта выбирается 250Гб. Этот размер является оптимальным для хранения временных данных и обмена информации, т.к. практически все данные хранятся на сервере. Модель монитора - 19 дюймовый Samsung 923NW. Данные мониторы являются весьма эргономичным и не очень дорогим решением для создания рабочих станций.

## **9.2 Выбор компьютеров и другого оборудования в зависимости от их стоимости**

Выбор серверных и рабочих станций в главе 9.1 изначально производился с учетом наименьшей цены за наиболее приемлемое качество для данной сети.

## 10 Выбор сетевой операционной системы

Для клиентских машин будет установлена операционная система Microsoft Windows 7. Возможности этой системы позволяют использовать ее как основную операционную систему для современных настольных компьютеров, используемых на предприятиях любого типа.

Операционная система обладает поддержкой мультитач-управления. Сетевая технология Branch Cache позволяет кэшировать содержимое интернет-трафика. Если пользователю в локальной сети потребуется файл, который уже был загружен кем-то из пользователей его сети, — он сможет получить его из локального кэш-хранилища, а не использовать канал с ограниченной пропускной способностью. Сетевой кеш может работать в двух режимах — Hosted Cache и Distributed Cache. В первом случае — файл хранится на выделенном локальном сервере под управлением Windows Server 2008 R2, во втором случае сервер не требуется, а кеш распределяется для хранения на компьютерах пользователей.

Дополнительным преимуществом Windows 7 можно считать более тесную интеграцию с производителями драйверов. Большинство из них определяются автоматически, при этом в 90 % случаев сохраняется обратная совместимость с драйверами для Windows Vista.

Также в Windows 7 реализована более гибкая настройка User Account Control (UAC), которая имеет ещё два промежуточных состояния между режимами «Всегда уведомлять» и «Никогда не уведомлять» — «Уведомлять, только при попытках программ внести изменения в компьютер» (положение по умолчанию), «Уведомлять, только при попытках программ внести изменения в компьютер (не затемнять рабочий стол)».

Функция DirectAccess позволяет устанавливать безопасное соединение с сервером в фоновом режиме, в отличие от VPN, которому требуется участие пользователя. Также DirectAccess может применять групповые политики до входа пользователя в систему.

Для серверных станций будет установлена Microsoft Windows Server 2008.

Windows Server 2008 позволяет лучше контролировать инфраструктуру серверов и сети и сконцентрироваться на решении задач первоочередной важности благодаря следующему:

- Упрощенное управление ИТ-инфраструктурой с помощью новых средств, обеспечивающих единый интерфейс для настройки и мониторинга серверов и возможность автоматизации рутинных операций.

- Оптимизация процессов установки Windows Server 2008 и управления ими за счет развертывания только нужных ролей и функций. Настройка конфигурации серверов уменьшает количество уязвимых мест и снижает потребность в обновлении программного обеспечения, что приводит к упрощению текущего обслуживания.

- Эффективное обнаружение и устранение неполадок с помощью мощных средств диагностики, дающих наглядное представление об актуальном состоянии серверной среды, как физической, так и виртуальной.

- Улучшенный контроль над удаленными серверами, например, серверами филиалов. Благодаря оптимизации процессов администрирования серверов и репликации данных вы сможете лучше обслуживать своих пользователей и избавиться от некоторых управленческих проблем.

Облегченное управление веб-серверами с помощью Internet Information Services 7.0 — мощной веб-платформы для приложений и служб. Эта модульная платформа имеет более простой интерфейс управления на основе задач и интегрированные средства управления состоянием веб-служб, обеспечивает строгий контроль над взаимодействием узлов, а также содержит ряд усовершенствований по части безопасности.

## 11 Выбор протоколов сети

Протокол внутри сети выбираем TCP/IP, так как он широко распространен и имеет высокую надежность и скорость передачи данных.

TCP/IP - это набор протоколов, которые позволяют физическим сетям объединяться вместе для образования Internet. TCP/IP соединяет индивидуальные сети для образования виртуальной вычислительной сети, в которой отдельные главные компьютеры идентифицируются не физическими адресами сетей, а IP-адресами.

В TCP/IP используется многоуровневая архитектура, которая четко описывает, за что отвечает каждый протокол. TCP и UDP обеспечивают высокоуровневые служебные функции передачи данных для сетевых программ, и оба опираются на IP при передаче пакетов данных. IP отвечает за маршрутизацию пакетов до их пункта назначения.

Данные, перемещающиеся между двумя прикладными программами, работающими на главных компьютерах Internet, "путешествуют" вверх и вниз по стекам TCP/IP на этих компьютерах. Информация, добавленная модулями TCP/IP на стороне отправителя, "разрезается" соответствующими TCP/IP-модулями на принимающем конце и используется для воссоздания исходных данных.

Стек протоколов TCP/IP основан на модели сетевого взаимодействия DOD и включает в себя протоколы четырех уровней:

- прикладного (application),
- транспортного (transport),
- сетевого (internet),
- уровня доступа к среде (network access).

Протоколы этих уровней полностью реализуют функциональные возможности модели OSI. На стеке протоколов TCP/IP построено всё взаимодействие пользователей в IP-сетях. Стек является независимым от физической среды передачи данных.



Физический уровень описывает среду передачи данных (будь то коаксиальный кабель, витая пара, оптическое волокно или радиоканал), физические характеристики такой среды и принцип передачи данных (разделение каналов, модуляцию, амплитуду сигналов, частоту сигналов, способ синхронизации передачи, время ожидания ответа и максимальное расстояние).

Канальный уровень описывает, каким образом передаются пакеты данных через физический уровень, включая кодирование (то есть специальные последовательности бит, определяющих начало и конец пакета данных). Ethernet, например, в полях заголовка пакета содержит указание того, какой машине или машинам в сети предназначен этот пакет.

Сетевой уровень изначально разработан для передачи данных из одной (под)сети в другую. Примерами такого протокола является X.25 и IP в сети ARPANET.

С развитием концепции глобальной сети в уровень были внесены дополнительные возможности по передаче из любой сети в любую сеть, независимо от протоколов нижнего уровня, а также возможность запрашивать данные от удалённой стороны, например в протоколе ICMP (используется для передачи диагностической информации IP-соединения) и IGMP (используется для управления multicast-потоками).

Пакеты сетевого протокола IP могут содержать код, указывающий, какой именно протокол следующего уровня нужно использовать, чтобы извлечь данные из пакета. Это число — уникальный IP-номер протокола. ICMP и IGMP имеют номера, соответственно, 1 и 2.

Протоколы транспортного уровня могут решать проблему негарантированной доставки сообщений («дошло ли сообщение до адресата?»), а также гарантировать правильную последовательность прихода данных. В стеке TCP/IP транспортные протоколы определяют, для какого именно приложения предназначены эти данные.

## 12 Методы передачи данных в сети

В сетях Ethernet используется метод доступа к среде передачи данных, называемый методом коллективного доступа с опознаванием несущей и обнаружением коллизий (carrier-sense-multiply-access with collision detection, CSMA/CD).

Этот метод используется исключительно в сетях с общей шиной (к которым относятся и радиосети, породившие этот метод). Все компьютеры такой сети имеют непосредственный доступ к общей шине, поэтому она может быть использована для передачи данных между любыми двумя узлами сети. Простота схемы подключения - это один из факторов, определивших успех стандарта Ethernet. Говорят, что кабель, к которому подключены все станции, работает в режиме коллективного доступа (multiply-access, МА).

Все данные, передаваемые по сети, помещаются в кадры определенной структуры и снабжаются уникальным адресом станции назначения. Затем кадр передается по кабелю. Все станции, подключенные к кабелю, могут распознать факт передачи кадра, и та станция, которая узнает собственный адрес в заголовках кадра, записывает его содержимое в свой внутренний буфер, обрабатывает полученные данные и посылает по кабелю кадр-ответ. Адрес станции-источника также включен в исходный кадр, поэтому станция-получатель знает, кому нужно послать ответ.

При описанном подходе возможна ситуация, когда две станции одновременно пытаются передать кадр данных по общему кабелю. Для уменьшения вероятности этой ситуации непосредственно перед отправкой кадра передающая станция слушает кабель (то есть принимает и анализирует возникающие на нем электрические сигналы), чтобы обнаружить, не передается ли уже по кабелю кадр данных от другой станции. Если опознается несущая (carrier-sense, CS), то станция откладывает передачу своего кадра до окончания чужой передачи, и только потом пытается вновь его передать. Но даже при таком алгоритме две станции одновременно могут решить, что по шине

в данный момент времени нет передачи, и начать одновременно передавать свои кадры. Говорят, что при этом происходит коллизия, так как содержимое обоих кадров сталкивается на общем кабеле, что приводит к искажению информации.

Чтобы корректно обработать коллизию, все станции одновременно наблюдают за возникающими на кабеле сигналами. Если передаваемые и наблюдаемые сигналы отличаются, то фиксируется обнаружение коллизии (collision detection, CD). Для увеличения вероятности немедленного обнаружения коллизии всеми станциями сети, ситуация коллизии усиливается посылкой в сеть станциями, начавшими передачу своих кадров, специальной последовательности битов, называемой jam-последовательностью.

После обнаружения коллизии передающая станция обязана прекратить передачу и ожидать в течение короткого случайного интервала времени, а затем может снова сделать попытку передачи кадра.

Из описания метода доступа видно, что он носит вероятностный характер, и вероятность успешного получения в свое распоряжение общей среды зависит от загруженности сети, то есть от интенсивности возникновения в станциях потребности передачи кадров. При разработке этого метода предполагалось, что скорость передачи данных в 10 Мб/с очень высока по сравнению с потребностями компьютеров во взаимном обмене данными, поэтому загрузка сети будет всегда небольшой. Это предположение остается часто справедливым и по сей день, однако уже появились приложения, работающие в реальном масштабе времени с мультимедийной информацией, для которых требуются гораздо более высокие скорости передачи данных. Поэтому наряду с классическим Ethernet'ом растет потребность и в новых высокоскоростных технологиях.

## **13 Разработка программного обеспечения «TWITTER-клиент»**

«Твиттер» (Twitter, от англ. to twitter – «чирикать, щебетать, болтать») – социальная сеть для публичного обмена сообщениями при помощи веб-интерфейса, SMS, средств мгновенного обмена сообщениями или сторонних программ-клиентов для пользователей интернета любого возраста. Публикация коротких заметок в формате блога получила название «микроблогинг». Пользование сервисом бесплатно. Пользование посредством SMS тарифицируется оператором согласно тарифному плану пользователя.

Работать с сервисом Twitter можно через Web-интерфейс с использованием браузера, но также возможно для этой цели использовать специальные программы, так называемые Twitter-клиенты. Работа через них считается более безопасной, чем через Web-интерфейс. Такие клиенты есть под разные платформы.

### **13.1 Обзор аналогов программных средств**

За последние несколько лет Twitter стал нашим постоянным спутником. Кто-то регулярно проверяет ленту в дороге, кто-то переписывается дома, а для кого-то это уже не только не средство общения, а возможность быстро получить необходимую информацию в сжатом виде. Наибольшей популярностью пользуются твиттер-клиенты для операционной системы Android.

#### **1) Twitter (официальное приложение).**

В последнее время официальное приложение Twitter получило новые интересные возможности, что сделало его несколько привлекательнее для пользователей. Речь идет о возможности прикреплять другие твиты к своим твитам, а также о поддержке изображений формата GIF. Стоит отметить, что загрузка и просмотр столь популярных сегодня анимированных изображений доступны только в официальном клиенте.

Кроме того, вы можете фильтровать результаты по типам контента (фото, видео и т.д.). Также доступны прямые сообщения, возможность добавлять фоловеров, импортировать контакты и, конечно, просматривать твиты. Это приложение подойдет вам для большинства основных задач, необходимых для управления вашим Twitter-аккаунтом. Если же у вас их несколько, вы можете добавить и переключаться между ними в одном приложении.

## 2) TweetCaster.

Конечно, TweetCaster предлагает базовую функциональность Twitter-клиента. Однако он превосходит официальное приложение по многим параметрам. Вы можете разделять людей и темы по группам или спискам, доступен точный поиск по категориям, но самой полезной возможностью, на мой взгляд, является просмотр статистики твитов и людей, которых вы фоловите. Чьи твиты вы репостите чаще всего? Как часто вы твитите? На эти и многие другие вопросы вам ответит статистика в TweetCaster.

Прочие приятности: двенадцать тем на выбор; настройка одновременных постов в Twitter и Facebook; фильтры и эффекты для фотографий. В бесплатной версии вы будете вынуждены наблюдать рекламные баннеры. Избавиться от них можно с помощью покупки TweetCaster Pro за 153 российских рубля

## 3) Tweedle.

Разработчик приложения заявляет: «Созданный с заботой о простоте и функциональности, Tweedle — это красивый и в то же время настраиваемый клиент Twitter».

Точнее это приложение описать сложно. В нем имеются отдельные секции для вашей ленты, уведомлений, упоминаний и прямых сообщений. Tweedle действительно дает довольно широкие возможности для персонализации. Вы можете менять 24-часовой формат, настраивать размер шрифта и использовать различн

## 4) Hootsuite.

С помощью Hootsuite вы можете пользоваться Twitter, Facebook,

LinkedIn, Google+ и Foursquare. Этот клиент предназначен для профессиональных медиа, которые управляют несколькими социальными аккаунтами.

В приложении доступна статистика, сообщающая, например, сколько просмотров получили ваши твиты.

При обновлении до версии Pro (\$9.99 за месяц или \$109.99 за год) вы сможете добавлять и управлять неограниченным числом аккаунтов, не отвлекаясь при этом на рекламные баннеры.

#### 5) Talon.

Основным отличием этого приложения является пользовательский интерфейс. Несмотря на его изначальную красоту, вы можете поменять шрифт или тему.

Доступна и настройка того, что отображается у вас в ленте. Например, если вы не хотите видеть в ней ретвиты, вы можете с легкостью от них избавиться. Это поможет вам навести долгожданный порядок в своей ленте. Вы можете написать твит практически с любого экрана приложения, нажав на иконку в виде бумажного самолета.

#### 6) Plume.

В Plume доступно управление несколькими аккаунтами, а также сокращение URL-ссылок с помощью Bit.ly.

Интересна возможность просматривать наиболее популярные тренды в Twitter на основе вашего местоположения.

В двух словах, Plume – это бесплатный, простой в использовании клиент с хорошим внешним видом.

Как и в аналогах, для того, чтобы избавиться от рекламы, вам придется купить версию Premium за 177 российских рублей.

#### 7) Fenix.

Fenix быстро завоевал любовь пользователей за простой и интуитивно-понятный интерфейс. Вы можете открывать ссылки прямо из ленты без лишних кликов. Чтение диалогов выполнено максимально удобно. О доступе с различных аккаунтов тоже не забыли.

Поддерживается в Fenix и беззвучный режим. В наличии есть несколько дополнительных тем.

### **13.2 Постановка задачи**

Необходимо создать простой, графический клиент на основе REST API с использованием STREAMING API, в котором отображаются сообщения от друзей в Twitter в соответствии с хронологией и который позволяет просматривать и обновлять состояние в Twitter.

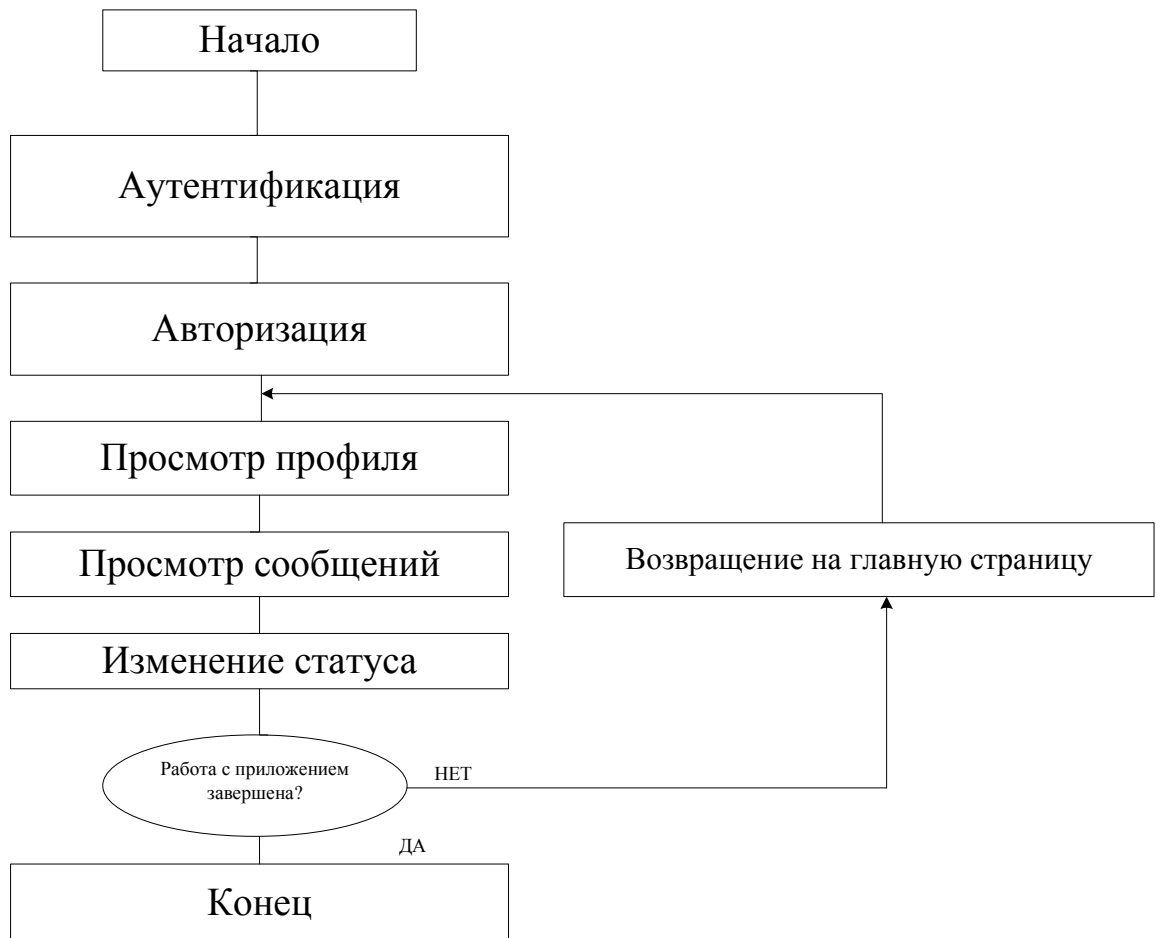
### **13.3 Разработка алгоритма и программного средства (ПС)**

При авторизации, аутентификации пользователя и переходе к соответствующим разделам приложения пользователю предоставлена возможность следующих действий:

- просмотр сообщений от всех друзей в Twitter в соответствии с хронологией;
- просмотр сообщений от конкретного пользователя в Twitter в соответствии с хронологией;
- обновление своего состояния в Twitter.

Подробный алгоритм работы системы представлен схемой алгоритма работы ПС в следующей главе.

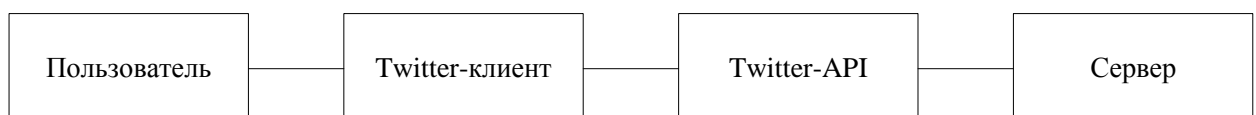
### 13.4 Разработка алгоритма работы ПС



### 13.5 Разработка спецификации требований к ПС

Проектирование данного программного средства не предусматривают наличие спецификаций.

### 13.6 Разработка структурной схемы ПС





### 13.7 Обоснование выбора языка и среды программирования

Java — строго типизированный объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Приложения Java обычно транслируются в специальный байт-код, поэтому они могут работать на любой компьютерной архитектуре, с помощью виртуальной Java-машины. Дата официального выпуска — 23 мая 1995 года.

Язык Java активно используется для создания мобильных приложений под операционную систему Android. При этом программы компилируются в нестандартный байт-код, для использования их виртуальной машиной Dalvik (начиная с Android 5.0 Lollipop виртуальная машина заменена на ART). Для такой компиляции используется дополнительный инструмент, а именно Android SDK (Software Development Kit), разработанный компанией Google.

Разработку приложений можно вести в среде Android Studio, NetBeans, в среде Eclipse, используя при этом плагин Android Development Tools (ADT), или в IntelliJ IDEA. Версия JDK при этом должна быть 5.0 или выше.

8 декабря 2014 года Android Studio признана компанией Google официальной средой разработки под ОС Android.

Android Studio — это интегрированная среда разработки (IDE) для работы с платформой Android, анонсированная 16 мая 2013 года на конференции Google I/O.

IDE находилась в свободном доступе начиная с версии 0.1, опубликованной в мае 2013, а затем перешла в стадию бета-тестирования, начиная с версии 0.8, которая была выпущена в июне 2014 года. Первая стабильная версия 1.0 была выпущена в декабре 2014 года, тогда же прекратилась поддержка плагина Android Development Tools (ADT) для Eclipse.

Android Studio, основанная на программном обеспечении IntelliJ IDEA от компании JetBrains, официальное средство разработки Android приложений. Данная среда разработки доступна для Windows, OS X и Linux.

## Особенности

- Новые функции появляются с каждой новой версией Android Studio.

На данный момент доступны следующие функции:

- Расширенный редактор макетов: WYSIWYG, способность работать с UI компонентами при помощи Drag-and-Drop, функция предпросмотра макета на нескольких конфигурациях экрана.
- Сборка приложений, основанная на Gradle.
- Различные виды сборок и генерация нескольких .apk файлов
- Рефакторинг кода
- Статический анализатор кода (Lint), позволяющий находить проблемы производительности, несовместимости версий и другое.
- Встроенный ProGuard и утилита для подписывания приложений.
- Шаблоны основных макетов и компонентов Android.
- Поддержка разработки приложений для Android Wear и Android TV.
- Встроенная поддержка Google Cloud Platform, которая включает в себя интеграцию с сервисами Google Cloud Messaging и App Engine.
- Android Studio 2.1 поддерживает Android N Preview SDK, а это значит, что разработчики смогут начать работу по созданию приложения для новой программной платформы.
- Новая версия Android Studio 2.1 способна работать с обновленным компилятором Jack, а также получила улучшенную поддержку Java 8 и усовершенствованную функцию Instant Run.
- Platform-tools 23.1.0 для Linux без объявления стала исключительно 64-разрядной, даже при попытке установить 32-разрядную версию. Иными словами Android Studio больше НЕ работает (выдаёт неустраняемые ошибки) в 32-разрядных версиях Linux.

## 13.8 Программная реализация разработанного алгоритма работы ПС (описание основных функций)

MainActivity – главная форма программы, с которой начинается работа, позволяет пользователю выбрать функцию для работы с программой: просмотр сообщений друзей или конкретного пользователя, просмотр своего профиля, смена состояния.

- функция получения данных о пользователе (twitter4j.User) организована с помощью технологии REST API;
- функция изменения состояния (twitter4j.Status) организована с помощью технологии STREAMING API;
- функция просмотра сообщений пользователей (twitter4j.Twitter) организована с помощью технологии REST API, она позволяет просматривать как сообщения всех друзей, так и конкретного пользователя.

LoggedIn – данная форма предоставляет функции для аутентификации и авторизации пользователя:

- функция аутентификации (twitter4j.auth.AccessToken) организована с помощью технологии OAuth1.0, код для авторизации формируется автоматически, чтобы избежать ошибок, правильность введенных данных проверяется;
- функция авторизации (twitter4j.auth.RequestToken) организована с помощью технологии OAuth1.0, правильность введенных данных проверяется.

## 13.9 Тестирование и отладка ПС

Функция	Входные данные	Ожидаемый результат	Полученный результат
twitter4j.auth.AccessToken	Имя пользователя и пароль	Прохождение аутентификации, получение кода для авто-	Прохождение аутентификации, получение кода для авто-

		ризации	ризации
twitter4j.auth.RequestToken	Код авторизации	Прохождение авторизации	Прохождение авторизации
twitter4j.User		Отображение данных пользователя	Отображение данных пользователя
twitter4j.Status	Текст	Изменение состояния	Изменение состояния
twitter4j.Twitter	Имя пользователя	Просмотр сообщений друзей или конкретного пользователя	Просмотр сообщений друзей или конкретного пользователя

### 13.10 Руководство пользователя

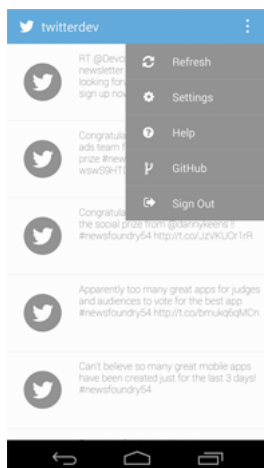
При отсутствии учетной записи Twitter перейдите по ссылке [twitter.com](https://twitter.com) и создайте ее. Чтобы зарегистрироваться, зайдите на сайт [twitter.com](https://twitter.com) и сразу увидите форму для регистрации на сервисе. Введите ваше имя и фамилию, адрес электронной почты и пароль.

Во время регистрации у вас может возникнуть проблемы с доступностью именем, оно может быть занято. Проверьте, все ли данные введены корректно, выберите свободно имя и нажмите кнопку «создать учетную запись».

Далее на ваш почтовый ящик придет уведомление, перейдите по ссылке в письме, и вы активируете ваш личный аккаунт. После этого откройте приложение и введите имя пользователя и пароль в соответствующие поля.



Далее вы можете пользоваться всеми возможностями приложения, перейдя по соответствующим пунктам меню: просмотреть свой профиль, изменить статус, просмотреть сообщения друзей или конкретного пользователя.



## **14 Методы и средства защиты информации в сети**

Как правило, принято различать пять основных средств защиты информации: технические, программные, криптографические, организационные, законодательные. Дадим определения некоторым этим средствам.

Технические средства защиты это механические, электромеханические, оптические, радио, радиолокационные, электронные и другие устройства и системы, способные выполнять самостоятельно или в комплексе с другими средствами функции защиты данных.

Программные средства защиты это специальные программы, включаемые в состав программного обеспечения системы, для обеспечения самостоятельно или в комплексе с другими средствами, функций защиты данных.

Криптографические средства защиты - это методы специального шифрования данных, в результате которого их содержание становится недоступным без применения некоторой специальной информации и обратного преобразования.

Организационные средства защиты специальные организационно-технические и организационно-правовые мероприятия, акты и правила, осуществляемые в процессе создания и эксплуатации системы для организации и обеспечения защиты информации.

Законодательные средства защиты – это законодательные акты, которые регламентируют правила использования и обработки информации, и устанавливают ответственность, и санкции за нарушение этих правил.

Основной особенностью любой сетевой системы является то, что её компоненты распределены в пространстве и связь между ними физически осуществляется при помощи сетевых соединений (коаксиальный кабель, витая пара, оптоволокно и т. п.) и программно при помощи механизма сообщений. При этом все управляющие сообщения и данные, пересылаемые между объектами распределенной вычислительной системы, передаются по сетевым соединениям в виде пакетов обмена.

Основные понятия компьютерной безопасности:

Угроза безопасности компьютерной системы - это потенциально возможное: происшествие, которое может оказать нежелательное воздействие на саму систему, а также на информацию, хранящуюся в ней.

Уязвимость компьютерной системы - это некая ее неудачная характеристика, которая делает возможным возникновение угрозы.

Атака на компьютерную систему - это действие, предпринимаемое злоумышленником, которое заключается в поиске и использовании той или иной уязвимости.

Наиболее простыми и дешёвыми являются административные методы защиты, как то: использование в сети стойкой криптографии, статических ARP-таблиц, hosts файлов вместо выделенных DNS-серверов, использование или неиспользование определённых операционных систем и другие методы.

Программные средства и методы защиты активнее и шире других применяются для защиты информации в персональных компьютерах и компьютерных сетях, реализуя такие функции защиты, как разграничение и контроль доступа к ресурсам; регистрация и анализ протекающих процессов, событий, пользователей; предотвращение возможных разрушительных воздействий на ресурсы; криптографическая защита информации; идентификация и аутентификация пользователей и процессов.

## **Заключение**

На сегодняшний день разработка и внедрение локальных информационных систем является одной из самых интересных и важных задач в области информационных технологий. Все больше возрастает стоимость информации и зависимость предприятий от оперативной и достоверной информации. В связи с этим появляется потребность в использовании новейших технологий передачи информации.

В ходе проекта была разработана сеть, объединившая 3 корпуса с возможностью обмена данными. Реализована технология качества сетевых технологий использовать архитектуру FDDI. Для удобства прокладки кабеля и его структуризации используется структурированная кабельная система. Имеется возможность расширения сети, т.к. у коммутатора остаются незадействованные порты. При необходимости можно предусмотреть дополнительные места подключения рабочих станций (дополнительные розетки), так что подключение рабочих станций к сети будет определяться временем настройки сетевого программного обеспечения.

Было выбрано активное и пассивное сетевое оборудование, среды передачи данных, конфигурации рабочих станций и серверов сети, сетевые протоколы, сетевая операционная система, разработана защита сети от несанкционированного внутреннего и внешнего доступа, так же предусмотрены меры предосторожности и профилактики при работе с рабочими и серверными станциями.



## Список использованных источников

- 1) Н.А. Олифер, В.Г. Олифер. Базовые технологии локальных сетей Центр Информационных Технологий.
- 2) В.Г. Олифер, Н.А. Олифер. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 2-ое издание. - СПб.: Питер. 2005. — 864с.:ил
- 3) Танненбаум Э. Компьютерные сети. – СПб.: Питер, 2002. – 992с.
- 4) Сети ЭВМ: протоколы стандарты, интерфейсы. Ю. Блэк; перев. с англ. - М.: Мир, 1990.
- 5) Протоколы Internet. С. Золотов. - СПб.: BHV - Санкт-Петербург, 1998,
- 6) Вычислительные системы, сети и телекоммуникации. Пятибратов и др. - ФИС, 1998.
- 7) Основы построения сетей. Учебное руководство для специалистов MCSE (+CD-ROM). Дж. Челлис, Ч. Перкинс, М. Стриб; перевод с англ. - Лори, 1997.
- 8) Толковый словарь по вычислительной технике; перев. с англ. - М.: Издательский отдел «Русская редакция» ТОО «Channel Trading Ltd.», 1995.
- 9) Филиппов М.В. Вычислительные машины, компьютерные сети и системы телекоммуникации // учебное пособие // Волгоград, ВФ МУПК, 2002.
- 10) В. Э. Фигурнов «IBM PC для пользователя».- М.: ИНФРА-М, 2003.
- 11) А. В. Гаврилов «Локальные сети ЭВМ».-М.: Издательство «Мир», 2004.
- 12) С. И. Казаков «Основы сетевых технологий», 2003 г.
- 13) А.Н. Глебов, Р.В. Густев, учебный материал «Организация сети», «Преимущества локальной сети», январь 2002 г.

## Приложение А

```
package com.twitterdev.twitter4j_sample_app.app;

import android.app.SearchManager;
import android.content.Context;
import android.os.AsyncTask;
import android.os.Build;
import android.os.Handler;

import android.support.v4.view.MenuItemCompat;
import android.view.Menu;

import android.view.MenuInflater;
import android.view.MenuItem;

import android.content.Intent;
import android.content.SharedPreferences;
import android.content.SharedPreferences.Editor;
import android.net.Uri;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.view.inputmethod.InputMethodManager;
import android.widget.ImageButton;
import android.widget.ImageView;

import android.widget.SearchView;
import android.widget.TextView;
import android.widget.Toast;

import twitter4j.Status;
import twitter4j.Twitter;
```

```

import twitter4j.TwitterException;
import twitter4j.User;
import twitter4j.TwitterFactory;
import twitter4j.auth.AccessToken;
import twitter4j.auth.RequestToken;
import twitter4j.conf.Configuration;
import twitter4j.conf.ConfigurationBuilder;

public class LoggedIn extends ActionBarActivity {

    private static SharedPreferences mSharedPreferences;
    private static Twitter twitter;
    private static RequestToken requestToken;
    private AccessToken accessToken;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // set full screen
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);
        // set context
        setContentView(R.layout.activity_home);

        // define variable for the settings
        mSharedPreferences = getApplicationContext().getSharedPreferences(
            "twitter4j-sample", 0);

        Log.v("LoggedIn", "getUser().execute(<String>);");
        // process getting a user @<:handle>
        new getUser().execute("twitterdev");
    }

    private class getUser extends AsyncTask<String, Void, String> {
        // set the last_tweet this is displayed on the device in a TextView
        private String last_tweet;
    }

```

```

@Override
protected String doInBackground(String... params) {
    Log.v("LoggedIn", "getUser..doInBackground");
    ConfigurationBuilder builder = new ConfigurationBuilder();
    builder.setOAuthConsumerKey(Constants.API_KEY);
    builder.setOAuthConsumerSecret(Constants.API_SECRET);

    // // Setup preferences
    mSharedPreferences = getApplicationCon-
text().getSharedPreferences(
        "twitter4j-sample", 0);

    // Access Token
    String access_token = mSharedPrefer-
ences.getString(Constants.PREF_KEY_OAUTH_TOKEN, "");
    // Access Token Secret
    String access_token_secret = mSharedPrefer-
ences.getString(Constants.PREF_KEY_OAUTH_SECRET, "");
    // Setup Access Token
    AccessToken accessToken = new AccessToken(access_token, ac-
cess_token_secret);
    // Setup instance of twitter to perform requests e.g. twit-
ter.showUser(<:handle>);
    Twitter twitter = new TwitterFacto-
ry(builder.build()).getInstance(accessToken);
    // Define the twitter_handle
    final String twitter_handle = params[0];
    // Try and make magic
    try {

        twitter4j.User user = twitter.showUser(twitter_handle);
        twitter4j.Status s = user.getStatus();
        final String last_tweet = s.getText();
        runOnUiThread(new Runnable() {

            public void run() {
                Log.v("LoggedIn", "running on UI thread update text view");
                TextView lt = (TextView) findViewById(R.id.last_tweet);
                lt.setText(last_tweet);
            }
        });
    } catch (Exception e) {
        Log.e("LoggedIn", "Error: " + e.getMessage());
    }
}

```

```

        TextView h = (TextView) findViewById(R.id.handleId);
        h.setText("@ " + twitter_handle);

    }
});
//Log.v("Last Tweet", s.getText());
} catch (TwitterException e) {
    Toast.makeText(getApplicationContext(), "Oops something bad
happened...", Toast.LENGTH_SHORT).show();
    e.printStackTrace();
}
return null;
}
}

```

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    //Used to put dark icons on light action bar
    final Menu m = menu;
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.main, menu);
    final MenuItem searchItem = menu.findItem(R.id.search_icon);
    final SearchView searchView = (SearchView) searchItem.getActionView();
    //searchView.setSubmitButtonEnabled(true);
    assert searchView != null;

    searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
        @Override
        public boolean onQueryTextSubmit(String query) {
            // TODO Auto-generated method stub
            Log.v("search", query);
            searchView.setIconified(true);
            searchView.clearFocus();
            searchView.setQuery("", false);
            new getUser().execute(query);
        }
    });
}

```

```

        // This method does not exist
        searchView.setIconified(true);
        return false;
    }

    @Override
    public boolean onQueryTextChange(String newText) {
        // TODO Auto-generated method stub

        return false;
    }

});
return super.onCreateOptionsMenu(menu);

}

/*@Override
public boolean onCreateOptionsMenu(Menu menu) {

    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}*/

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();
    if (id == R.id.menu_logout) {
        mSharedPreferences = getApplicationCon-
text().getSharedPreferences(
        "twitter4j-sample", 0);
        SharedPreferences.Editor editor = mSharedPreferences.edit();
        editor.clear();
        editor.commit();
        Intent myIntent = new Intent(getBaseContext(), MainActivity.class);

```

```

        startActivity(myIntent);
        return true;
    } else if (id == R.id.refresh){
        Intent refresh = new Intent(getApplicationContext(), LoggedIn.class);
        startActivity(refresh);
    }
    return super.onOptionsItemSelected(item);
}
}

```

```

package com.twitterdev.twitter4j_sample_app.app;

```

```

import android.content.Intent;
import android.content.SharedPreferences;
import android.content.SharedPreferences.Editor;
import android.net.Uri;
import android.os.AsyncTask;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.Toast;

```

```

import twitter4j.Twitter;
import twitter4j.TwitterFactory;
import twitter4j.auth.AccessToken;
import twitter4j.auth.RequestToken;
import twitter4j.conf.Configuration;
import twitter4j.conf.ConfigurationBuilder;

```

```

public class MainActivity extends ActionBarActivity {

```

```

private static SharedPreferences mSharedPreferences;
private static Twitter twitter;
private static RequestToken requestToken;
private AccessToken accessToken;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    // set full screen
    requestWindowFeature(Window.FEATURE_NO_TITLE);
    setContentView(R.layout.activity_main);
    getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
        WindowManager.LayoutParams.FLAG_FULLSCREEN);
    ImageView imgV = (ImageView) findViewById(R.id.imageView);
    imgV.setScaleType(ImageView.ScaleType.FIT_XY);

    //
    mSharedPreferences = getApplicationContext().getSharedPreferences(
        "twitter4j-sample", 0);

    // handle on click
    ImageButton btnLoginTwitter = (ImageButton) findViewById(R.id.btnLogin);
    btnLoginTwitter.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View arg0) {
            // Call login twitter function
            loginToTwitter();
        }
    });

    // capture uri
    if (!isTwitterLoggedInAlready()) {
        Uri uri = getIntent().getData();
        if (uri != null &&
            uri.toString().startsWith(Constants.CALLBACKURL)) {

```



```

        // OAuth verifier
        final String verifier = uri

.getQueryParameter(Constants.URL_TWITTER_OAUTH_VERIFIER);

        try {

            Thread thread = new Thread(new Runnable(){
                @Override
                public void run() {
                    try {

                        // Get the access token
                        accessToken = twitter.getOAuthAccessToken(
                            requestToken, verifier);
                        Log.v("accessToken", accessToken.getToken());
                        // Shared Preferences
                        mSharedPreferences = getApplicationCon-
text().getSharedPreferences(
                            "twitter4j-sample", 0);
                        Editor e = mSharedPreferences.edit();

                        // After getting access token, access token secret
                        // store them in application preferences
                        e.putString(Constants.PREF_KEY_OAUTH_TOKEN,
accessToken.getToken());
                        e.putString(Constants.PREF_KEY_OAUTH_SECRET,
accessToken.getTokenSecret());
                        // Store login status - true

                        e.putBoolean(Constants.PREF_KEY_TWITTER_LOGIN, true);
                        e.commit(); // save changes

                        Log.e("Twitter OAuth Token", "> " + ac-
cessToken.getToken());
                        Intent myIntent = new Intent(getBaseContext(),
LoggedIn.class);
                        startActivity(myIntent);
                    }
                }
            });
        }
    }
}

```

```

        // Hide login button
    } catch (Exception e) {
        e.printStackTrace();
    }
}
});
thread.start();

} catch (Exception e) {
    // Check log for login errors
    Log.e("Twitter Login Error", "> " + e.getMessage());
    e.printStackTrace();
}
}
} else {
    Intent myIntent = new Intent(getBaseContext(), LoggedIn.class);
    startActivity(myIntent);

}
}

private boolean isTwitterLoggedInAlready() {
    // return twitter login status from Shared Preferences
    return mSharedPreferences.getBoolean(Constants.PREF_KEY_TWITTER_LOGIN, false);
}

private void loginToTwitter() {
    // Check if already logged in
    if (!isTwitterLoggedInAlready()) {
        // Setup builder
        ConfigurationBuilder builder = new ConfigurationBuilder();
        // Get key and secret from Constants.java
        builder.setOAuthConsumerKey(Constants.API_KEY);
        builder.setOAuthConsumerSecret(Constants.API_SECRET);

        // Build
        Configuration configuration = builder.build();
        TwitterFactory factory = new TwitterFactory(configuration);

```

```

        twitter = factory.getInstance();

        // Start new thread for activity (you can't do too much work on the
        UI/Main thread.
        Thread thread = new Thread(new Runnable(){
            @Override
            public void run() {
                try {

                    requestToken = twitter
                        .getOAuthRequestToken(Constants.CALLBACKURL);
                    MainActivity.this.startActivity(new In-
tent(Intent.ACTION_VIEW, Uri
                        .parse(requestToken.getAuthenticationURL())));

                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
        thread.start();
    } else {
        Intent myIntent = new Intent(getBaseContext(), LoggedIn.class);
        startActivity(myIntent);
    }
}

```