

УЗБЕКСКОЕ АГЕНСТВО СВЯЗИ И ИНФОРМАТИЗАЦИИ
ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

К защите допустить
Зав. Кафедрой ТС и СК
к.т.н. доцент Эшмурадов А.М.

_____2009__г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

На тему Разработка виртуального практического
занятия «Построение сетей на базе протокола N.323»
дисциплины NGN

Выпускник	_____	<u>Шатуновский А. И.</u>
Руководитель	_____	<u>Еркинбаева Л. Т.</u>
Рецензент	_____	_____
Консультант ОТ и ТБ	_____	<u>Абдуллаева С. М.</u>

Ташкент-2009

АННОТАЦИЯ

В данной выпускной квалификационной работе рассмотрены вопросы построения сетей на базе протокола H.323. В нескольких словах рассмотрены основные преимущества сетей IP-телефонии. Описано назначение и основные возможности виртуального практического занятия. Дан краткий обзор структуры построения программы. Так же описаны возможности развития данной программы. Рассмотрены вопросы охраны труда и техники безопасности.

SUMMARY

In this graduation qualification work problems of building H.323 based networks are described. Some words about advantages of VoIP networks are given. Also this work is spoken about functions and basic potential of this virtual practical training. Short review of program structure is given. Development potential of this program is described. Problems of protection of labour and safety measures are described.

АННОТАЦИЯ

Мазкур малакавий битирув ишида H.323 протокол асосида тармоқ қурилиши кўрсатилган. IP- телефон тармоғининг асосий афзалликлари кўрилган. Виртуал амалиёт машғулотининг вазибалари ва асосий имкониятлари ёзилган. Дастурнинг тузилиш структураси ва кенгайтириш имкониятлари келтирилган. Техника хавфсизлиги ва меҳнатни муҳофаза қилиш масалалари ҳам кўрсатилган.

ОГЛАВЛЕНИЕ

АННОТАЦИЯ.....	5
ОГЛАВЛЕНИЕ	7
ВВЕДЕНИЕ	9
1. СТРУКТУРА СЕТИ НА БАЗЕ Н.323	11
1.1. Основные элементы и назначение.....	11
1.1.1. Терминал Н.323.....	13
1.1.2. Шлюз Н.323	15
1.1.3. Привратник.....	17
1.1.4. Устройство управления конференциями	18
1.2. Сигнализация Н.323.....	22
1.2.1. Семейство протоколов Н.323	22
1.2.2. Протокол RAS.....	22
1.2.3. Сигнальный канал Н.225.0	23
1.2.4. Управляющий канал Н.245.....	24
Выводы.....	25
2. РАЗРАБОТКА МЕТОДИЧЕСКОГО УКАЗАНИЯ В ПЕЧАТНОМ ВИДЕ.....	27
2.1. Разработка условия задачи.....	27
2.2. Разработка вариантов заданий.....	27
2.3. Формирование алгоритма решения задачи	30
2.4. Разработка контрольных вопросов.....	30
Выводы.....	31
3. РАЗРАБОТКА ВИРТУАЛЬНОГО ПРАКТИЧЕСКОГО ЗАНЯТИЯ «ПОСТРОЕНИЕ СТЕЙ НА БАЗЕ ПРОТОКОЛА Н.323» ДИСЦИПЛИНЫ NGN	36
3.1. Описание средств разработки приложения.....	32
3.2. Разработка структуры программы.....	34

3.3.	Разработка блок-схемы программы	35
3.4.	Разработка графического интерфейса.....	41
3.5.	Пример решения задачи	42
	Выводы.....	46
4.	ОХРАНА ТРУДА И ТЕХНИКА БЕЗОПАСНОСТИ.....	47
4.1.	Меры безопасности при работе за компьютером	47
	Введение.....	47
4.1.1.	Зрительная работа за компьютером и ее последствия.....	47
4.1.2.	Чистота.....	48
4.1.3.	Микротравмы	48
4.1.4.	Технические методы увеличения безопасности работы за компьютером.....	49
4.1.5.	Устройства ввода информации	50
4.1.6.	Эргономичная организация рабочего места	50
4.1.7.	Рабочее пространство.....	51
4.1.8.	Работа с клавиатурой.	51
4.1.9.	Расположение монитора.	51
4.1.10.	Внутренний объем.....	51
4.1.11.	Кресло	51
4.1.12.	Положение за компьютером.....	52
4.2.	Выводы.....	52
	ЗАКЛЮЧЕНИЕ.....	53
	СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....	54
	ПРИЛОЖЕНИЕ 1	55
	ПРИЛОЖЕНИЕ 2	96

ВВЕДЕНИЕ

В настоящее время отрасль телекоммуникации развивается очень интенсивно по всему миру. Активно внедряются новые технологии, увеличивается пропускная способность магистральных каналов и объем переданного трафика. По данным на конец 2008 года количество пользователей сети Интернет достигло 1,5 миллиарда. Вместе с этим наблюдается увеличение доли мультимедийного трафика, и рост количества предоставляемых услуг. Учитывая темпы развития телекоммуникационных услуг, достаточно непросто создать общую инфраструктуру для качественного предоставления всех современных услуг. Для решения данной проблемы была разработана концепция NGN – сетей следующего поколения.

Главной идеей концепции NGN является предоставление неограниченного набора услуг и приложений различных режимов и контекста (услуги режима реального времени, услуги, не требующие поддержки режима реального времени, услуги односторонней, многосторонней и широкоэмитальной передачи, мультимедийные услуги и т.д.), независимо от используемой технологической платформы. Это достигается благодаря гибкости, функциональности и набору так называемых сервисных конфигураций сети телекоммуникаций следующего поколения.

Основная идея сети телекоммуникаций следующего поколения предполагает построение единой сетевой инфраструктуры для оказания любых видов услуг, что фактически предполагает работу с любым видом трафика на одной сетевой инфраструктуре.

Среди услуг, предоставляемых современными телекоммуникационными компаниями, достаточно популярной и востребованной является услуга IP – телефонии или VoIP на базе стандарта H.323. Данный сервис предусматривает передачу голосового и видео трафика по сетям с пакетной коммутацией, что существенно позволяет сократить стоимость международных переговоров за счет более эффективного использования полосы пропускания каналов.

Данный сервис является востребованным особенно в период мирового финансового кризиса, когда все организации ищут возможности для экономичного использования средств.

Президент Республики Узбекистан И.А.Каримов в своей книге «Мировой финансовый экономический кризис, пути и меры его преодоления в условиях Узбекистана» пишет: «Узбекистан за истекший период своего независимого развития всегда придерживался принципов по отказу от краткосрочных спекулятивных кредитов и привлечения иностранных инвестиций в основном на долгосрочной основе и по льготным процентным ставкам». Таким образом, развитие IP – телефонии является приоритетной задачей, поскольку это экономически выгодное капиталовложение на длительный срок, особенно в условиях мирового финансового кризиса.

Услуги IP – телефонии, предоставляемые на базе рекомендации H.323, позволяют не только сэкономить за счет международных переговоров, но и за счет командировок для ведения переговоров, поскольку есть возможность организовать видеоконференцию для этих целей, что обойдется гораздо дешевле. Все это делает решения на базе стандарта H.323 перспективными и конкурентоспособными.

1. СТРУКТУРА СЕТИ НА БАЗЕ H.323

1.1. Основные элементы и назначение

Рекомендация H.323 предусматривает применение различных алгоритмов сжатия речевой информации, что позволяет использовать полосу пропускания гораздо более эффективно, чем в сетях с коммутацией каналов. Оконечные устройства H.323 поддерживают передачу информации в режиме многоадресной рассылки, что позволяет организовывать конференции без дорогостоящих устройств управления конференциями (MCU), хотя на сегодняшний день без MCU не обойтись, т.к. режим многоадресной рассылки, как правило, IP-сетями не поддерживается.

Рекомендация Международного союза электросвязи H.323 является первой зонтичной спецификацией систем мультимедийной связи для работы в сетях с коммутацией пакетов, не обеспечивающих гарантированное качество обслуживания. В рекомендациях, входящих в семейство H.323, определены протоколы, методы и сетевые элементы, необходимые для организации мультимедийной связи между двумя или более пользователями.

Сети, построенные на базе протоколов H.323, ориентированы на интеграцию с телефонными сетями и могут рассматриваться как сети ISDN, наложенные на сети передачи данных. В частности, процедура установления соединения в таких сетях IP-телефонии базируется на рекомендации ITU-T Q.931 и практически идентична той же процедуре в сетях ISDN.

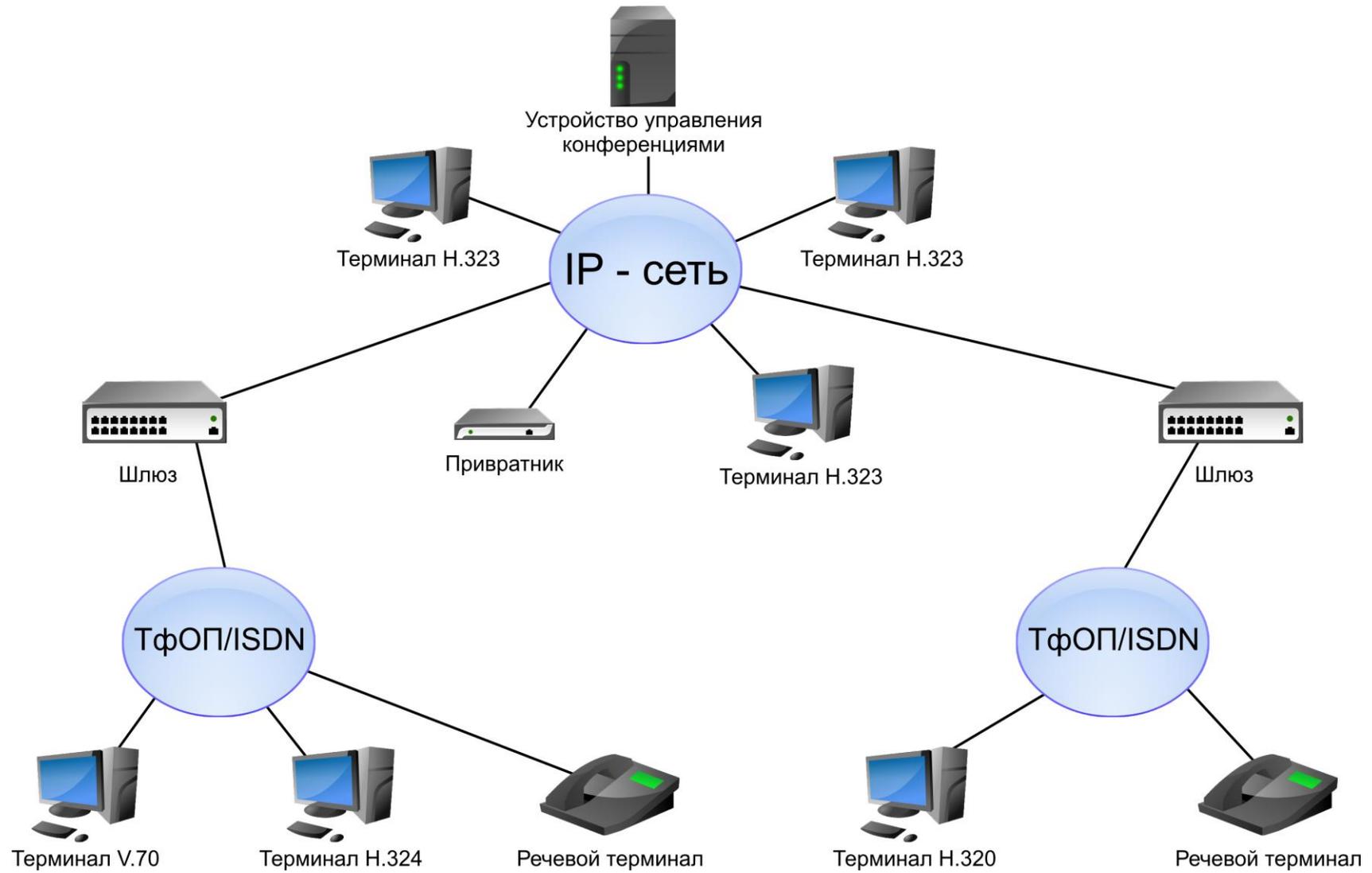


Рис. 1.1. Архитектура сети H.323

Этот вариант построения сетей IP-телефонии ориентирован на операторов местной телефонной связи (или на компании, владеющие транспортными сетями), которые желают использовать сети с маршрутизацией пакетов IP для предоставления услуг междугородной и международной связи. Протокол RAS, входящий в семейство протоколов H.323, предоставляет операторам высокий уровень контроля использования сетевых ресурсов и обеспечивает поддержку аутентификации пользователей и начисления платы за предоставленные услуги.

На рис. 1.1 изображена архитектура сети, построенной на базе рекомендации H.323.

Основными устройствами сети являются: терминал, шлюз, привратник и устройство управления конференциями.

1.1.1. Терминал H.323

Терминал H.323 - это оконечное устройство сети IP-телефонии, обеспечивающее двухстороннюю речевую или мультимедийную связь с другим терминалом, шлюзом или устройством управления конференциями. Структурная схема терминала H.323 приведена на рис. 1.2.

Пользовательский интерфейс управления системой дает пользователю возможность создавать и принимать вызовы, а также конфигурировать систему и контролировать ее работу.

Модуль управления поддерживает три вида сигнализации: H.225, H.245 и RAS.

Телематические приложения обеспечивают передачу пользовательских данных, неподвижных изображений и файлов, доступ к базам данных и т.п. Стандартным протоколом для поддержки таких приложений является протокол T.120.

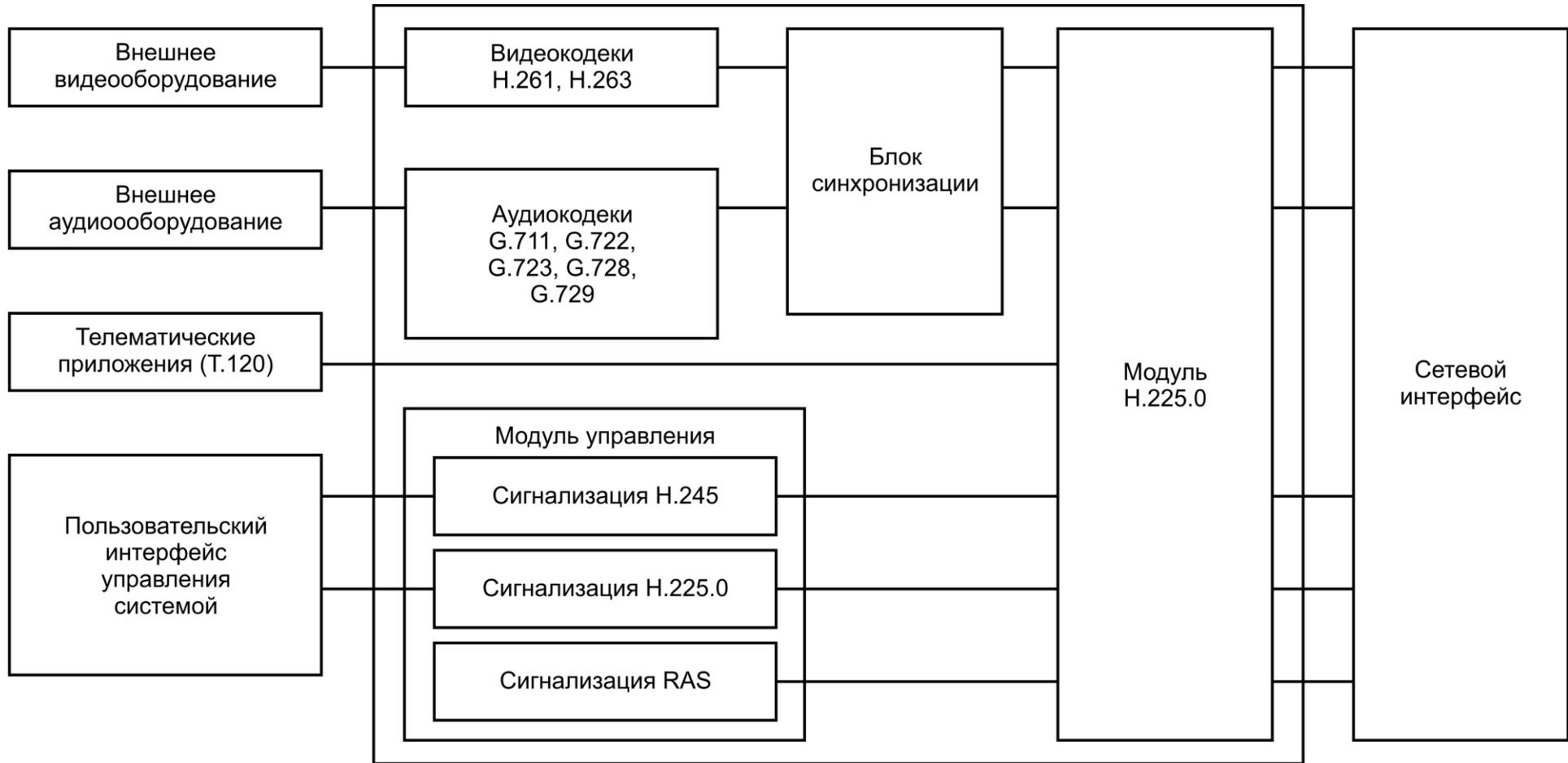


Рис. 1.2. Терминал H.323

Модуль Н.225.0 отвечает за преобразование видеоинформации, речи, данных и сигнальной информации в вид, пригодный для передачи по сетям с маршрутизацией пакетов IP, и за обратное преобразование. Кроме того, функциями модуля являются разбиение информации на логические кадры, нумерация последовательно передаваемых кадров, выявление и коррекция ошибок.

Сетевой интерфейс обеспечивает гарантированную передачу управляющих сообщений Н.245, сигнальных сообщений Н.225.0 (Q.931) и пользовательских данных при помощи протокола TCP и негарантированную передачу речевой и видеоинформации, а также сообщений RAS, при помощи протокола UDP.

Блок синхронизации вносит задержку на приемной стороне с целью обеспечить синхронизацию источника информации с ее приемником, согласование речевых и видеоканалов или сглаживание вариации задержки информации.

Видеокодеки кодируют видеоинформацию, поступающую от внешнего источника видеосигналов (видеокамеры или видеомагнитофона), для ее передачи по сети с маршрутизацией пакетов IP и декодируют сигналы, поступающие из сети, для последующего отображения видеоинформации на мониторе или телевизоре.

Аудиокодеки кодируют аудиоинформацию, поступающую от микрофона (или других источников аудиоинформации), для ее передачи по сети с маршрутизацией пакетов IP и декодируют сигналы, поступающие из сети, для последующего воспроизведения.

1.1.2. Шлюз Н.323

Основной функцией шлюза является преобразование речевой (мультимедийной) информации, поступающей со стороны ТФОП с постоянной скоростью, в вид, пригодный для передачи по IP-сетям, т.е. кодирование информации, подавление пауз в разговоре, упаковка информации в пакеты RTP/UDP/IP, а также обратное преобразование.

Кроме того, шлюз должен уметь поддерживать обмен сигнальными сообщениями как с коммутационным или терминальным оборудованием ТФОП, так и с привратником или оконечным устройством сети Н.323. Таким образом, шлюз должен преобразовывать аналоговую абонентскую сигнализацию, сигнализацию по 2ВСК, сигнальные сообщения систем сигнализации DSS1 и ОКС7 [6,7] в сигнальные сообщения Н.323. Спецификации преобразования сигнализации Q.931 (DSS1, QSIG) и ОКС7 в сигнализацию Н.225.0, основанную на Q.931, приведены в рекомендации ITU-T Н.246. Для поддержки дополнительных услуг в шлюзе может быть обеспечена прозрачная передача сигнальных сообщений Q.932 и Н.450.

Желательно, чтобы шлюз мог генерировать и распознавать сигналы DTMF на стороне ТФОП и передавать сигналы DTMF в сообщениях Н.245 userInputIndication по сети с маршрутизацией пакетов IP.

При отсутствии в сети привратника должна быть реализована еще одна функция шлюза - преобразование номера ТФОП в транспортный адрес IP-сети.

Со стороны сетей с маршрутизацией пакетов IP, так же, как и со стороны ТфОП, шлюз может участвовать в соединениях в качестве терминала или устройства управления конференциями (рис. 1.3).

Примечательно, что шлюз может изначально участвовать в соединении в качестве терминала, а затем, при помощи сигнализации H.245, продолжить работу в качестве устройства управления конференциями.

. На рис. 1.4 представлены некоторые услуги, которые могут быть реализованы в прикладном программном обеспечении на базовой платформе аппаратных и программных средств шлюза IP-телефонии.



Рис. 1.3. Возможные конфигурации шлюза



Рис. 1.4. Платформа и услуги шлюза IP-телефонии

1.1.3. Привратник

В привратнике сосредоточен весь интеллект сетей IP-телефонии, базирующихся на рекомендации ITU H.323. Сеть H.323 имеет зонную архитектуру (рис. 1.5). Привратник выполняет функции управления зоной сети IP-телефонии, в которую входят терминалы, шлюзы и устройства управления конференциями, зарегистрированные у этого привратника. Разные участки зоны сети H.323 могут быть территориально разнесены и соединяться друг с другом через маршрутизаторы. Следует обратить внимание на то, что коммутаторы кадров Ethernet и маршрутизаторы пакетов IP не являются сетевыми элементами H.323, так как они работают на звеньевом или сетевом уровнях соответственно, в то время как оборудование H.323 работает на прикладном уровне стека протоколов TCP/IP.

В число наиболее важных функций, выполняемых привратником, входят:

- преобразование имени абонента, телефонного номера, адреса электронной почты и др. в транспортный адрес сети с маршрутизацией пакетов IP (IP адрес и номер порта TCP);
- контроль доступа пользователей системы к услугам IP-телефонии при помощи сигнализации RAS (используются сообщения ARQ/ACF/ARJ);
- контроль, управление и резервирование пропускной способности сети;
- маршрутизация сигнальных сообщений между терминалами, расположенными в одной зоне; привратник может организовывать сигнальный канал непосредственно между терминалами или ретранслировать сигнальные сообщения от одного терминала к другому.

В том случае, когда вызывающий абонент знает IP-адрес терминала вызываемого абонента, соединение между двумя устройствами может быть установлено без помощи привратника. Это означает, что наличие в сети H.323 привратника не обязательно. Но, в то же время, следует отметить, что при наличии привратника обеспечивается мобильность абонентов, т.е. способность пользователя получить доступ к услугам с любого терминала в любом месте сети и способность сети идентифицировать пользователей при их перемещении из одного места в другое.

При отсутствии в сети привратника преобразование адреса вызываемого абонента, поступающего со стороны ТФОП в формате E. 164, в транспортный адрес IP-сети должно выполняться шлюзом.

1.1.4. Устройство управления конференциями

Рекомендация H.323 предусматривает три вида конференций (рис. 1.6).

Первый вид - централизованная конференция, в которой оконечные устройства соединяются в режиме точка-точка с устройством управления конференциями (MCU), контролирующим процесс создания и завершения конференции, а также обрабатывающим потоки пользовательской информации.

Второй вид - децентрализованная конференция, в которой каждый ее участник соединяется с остальными участниками в режиме точка - группа точек, и оконечные устройства сами обрабатывают (переключают или смешивают) потоки информации, поступающие от других участников конференции.

Третий вид - смешанная конференция, т.е. комбинация двух предыдущих видов.

Преимущество централизованной конференции - сравнительно простые требования к терминальному оборудованию, недостаток - большая стоимость устройства управления конференциями.



Рис. 1.5. Зона сети H.323



Рис. 1.6. Разные виды конференции в сети H.323

Для децентрализованной конференции требуется более сложное терминальное оборудование, кроме того, желательно, чтобы в сети поддерживалась передача пакетов IP в режиме многоадресной рассылки (IP multicasting).

Устройство управления конференциями MCU содержит один обязательный элемент - контроллер многоточечных соединений - Multipoint controller (MC). Кроме того, MCU может содержать один или более процессоров для обработки информации пользователей при многоточечных соединениях - Multipoint processor (MP). Контроллер может быть физически совмещен с привратником, со шлюзом или с MCU, а MCU, в свою очередь, может быть совмещено со шлюзом или с привратником (рис. 1.7).

Контроллер конференций должен использоваться для организации конференции любого вида. Он организует обмен между участниками конференции данными о функциональных возможностях (capabilities) их терминалов, указывает, в каком режиме (с использованием каких кодеков) участники конференции могут передавать информацию, причем этот режим может изменяться в ходе конференции, например, при подключении к ней нового участника. Таким образом, контроллер MC определяет режим конференции (selected communication mode - SCM), который может быть общим для всех участников конференции или отдельным для каждого из них.

Так как в сети может быть несколько контроллеров MC, то для каждой вновь создаваемой конференции должна проводиться процедура определения ведущего и ведомого оборудования, для того, чтобы выявить тот из контроллеров MC, который будет управлять данной конференцией.

При организации централизованной конференции, кроме контроллера MC, должен использоваться процессор MP, обрабатывающий пользовательскую информацию и отвечающий за переключение или смешивание речевых потоков, видеoinформации и данных. При организации децентрализованной конференции процессор MP не используется.

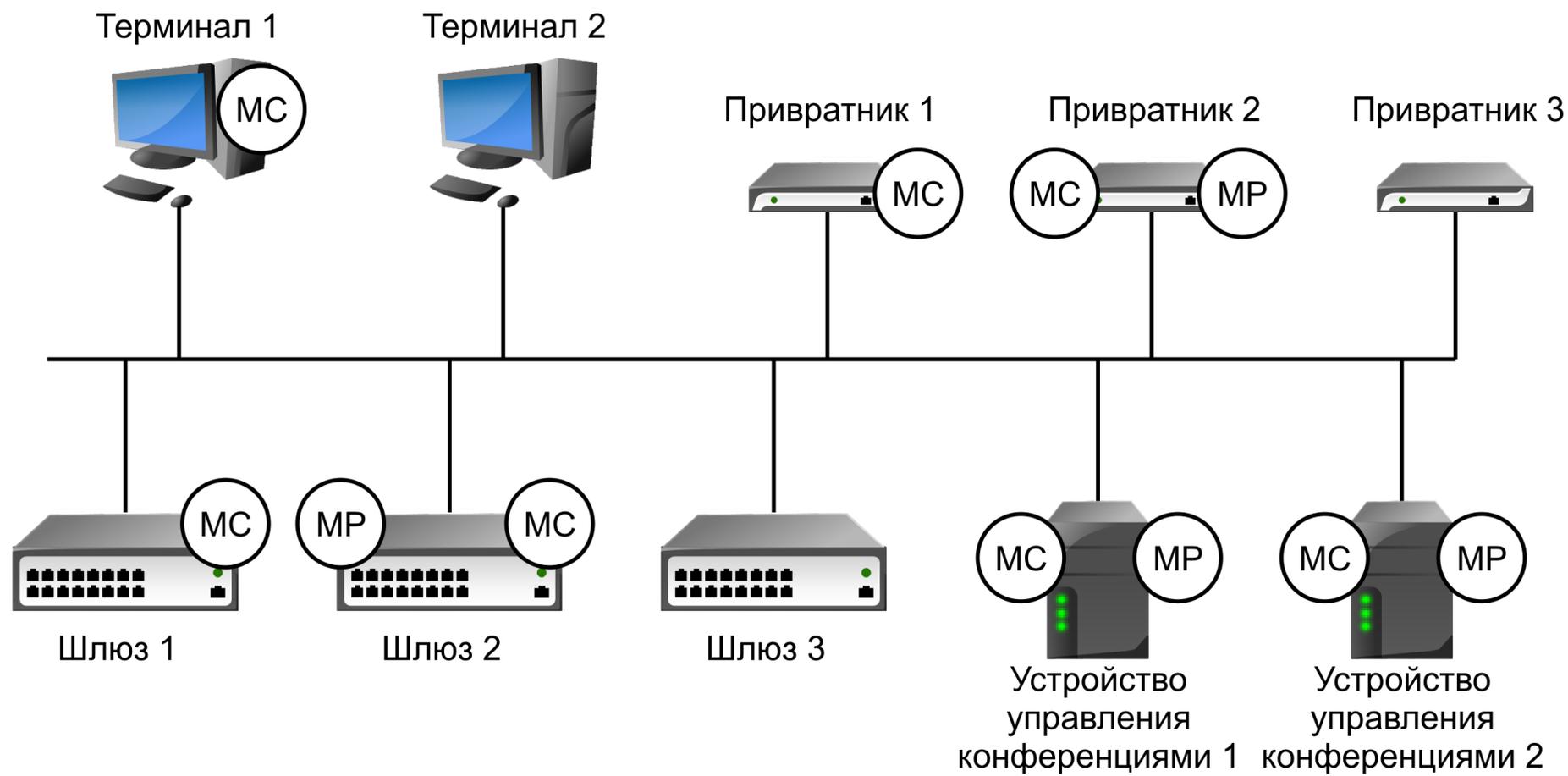


Рис. 1.7. Возможная реализация MC и MP в оборудовании H.323

1.2. Сигнализация H.323

1.2.1. Семейство протоколов H.323

Семейство протоколов H.323 включает в себя три основных протокола: протокол взаимодействия оконечного оборудования с привратником - RAS, протокол управления соединениями - H.225 и протокол управления логическими каналами - H.245.

Эти три протокола, совместно с Интернет-протоколами TCP/IP, UDP, RTP и RTCP, а также с протоколом Q.931, представлены на рис.1.8. Суть изображенной на этом рисунке иерархии заключается в следующем. Для переноса сигнальных сообщений H.225 и управляющих сообщений H.245 используется протокол с установлением соединения и с гарантированной доставкой информации - TCP. Сигнальные сообщения RAS переносятся протоколом с негарантированной доставкой информации - UDP. Для переноса речевой и видеоинформации используется протокол передачи информации в реальном времени - RTP. Контроль переноса пользовательской информации производится протоколом RTCP.

1.2.2. Протокол RAS

Международный союз электросвязи в рекомендации H.225.0 определил протокол взаимодействия рассмотренных в предыдущей главе компонентов сети H.323: оконечного оборудования (терминалов, шлюзов, устройств управления конференциями) с привратником. Этот протокол получил название RAS (Registration, Admission and Status).

Таблица. 1.1

Семейство протоколов H.323

Гарантированная доставка информации по протоколу TCP		Негарантированная доставка информации по протоколу UDP		
H.245	H.225		Потоки речи и видеоинформации	
	Управление соединением (Q.931)	RAS	RTCP	RTP
TCP		UDP		

IP
Канальный уровень
Физический уровень

Основными процедурами, выполняемыми оконечным оборудованием и привратником с помощью протокола RAS, являются:

1. Обнаружение привратника.
2. Регистрация оконечного оборудования у привратника.
3. Контроль доступа оконечного оборудования к сетевым ресурсам.
4. Определение местоположения оконечного оборудования в сети.
5. Изменение полосы пропускания в процессе обслуживания вызова.
6. Опрос и индикация текущего состояния оконечного оборудования.
7. Оповещение привратника об освобождении полосы пропускания, ранее занимавшейся оборудованием.

Выполнение первых трех процедур, предусмотренных протоколом RAS, является начальной фазой установления соединения с использованием сигнализации H.323. Далее следуют фаза сигнализации H.225.0 (Q.931) и обмен управляющими сообщениями H.245. Разъединение происходит в обратной последовательности: в первую очередь закрывается управляющий канал H.245 и сигнальный канал H.225.0, после чего по каналу RAS привратник оповещается об освобождении ранее занимавшейся оконечным оборудованием полосы пропускания.

Для переноса сообщений протокола RAS используется протокол негарантированной доставки информации UDP. В связи с этим ITU-T рекомендовал передавать повторно те сообщения RAS, получение которых не было подтверждено в течение установленного промежутка времени. Оконечное оборудование или привратник, не имеющие возможности в текущий момент времени ответить на полученный запрос, могут передавать сообщение RIP (Request in Progress) для индикации того, что запрос находится в стадии обработки. При приеме сообщения RIP привратник и оконечное оборудование должны перезапустить свои таймеры.

Важно отметить, что в сети без привратника сигнальный канал RAS вообще не используется.

1.2.3. Сигнальный канал H.225.0

Процедуры управления соединениями в сетях H.323 специфицированы Международным союзом электросвязи в рекомендации H.225.0. Данные процедуры предусматривают использование в базовом процессе обслуживания вызова ряда сигнальных сообщений Q.931, причем должен быть реализован симметричный обмен сигнальными сообщениями в соответствии с приложением D к

рекомендации Q.931. Это требование не распространяется на взаимодействие шлюза с сетью коммутации каналов.

Для реализации дополнительных услуг в соответствии с рекомендацией H.450 в сетях, построенных по рекомендации H.323, привлекаются сигнальные сообщения Q.932. В дан ном. параграфе рассматриваются наиболее часто используемые сигнальные сообщения.

Сообщение Setup передается вызывающим оборудованием с целью установить соединение. Это сообщение передается на общеизвестный TCP порт 1720 вызываемого оборудования.

Сообщение Call Proceeding передается вызываемому оборудованию, чтобы известить его о том, что вызов принят к обслуживанию.

Сообщение Alerting передается вызываемому оборудованию и информирует его о том, что вызываемое оборудование не занято, и что пользователю подается сигнал о входящем вызове.

Сообщение Connect передается вызываемому оборудованию и информирует его о том, что вызываемый пользователь принял входящий вызов. Сообщение Connect может содержать транспортный адрес управляющего канала H.245.

Сообщение Release Complete передается вызывающим или вызываемым оборудованием с целью завершить соединение. Это сообщение передается только в том случае, когда открыт сигнальный канал.

Сообщение Q.932 Facility используется для обращения к дополнительным услугам в соответствии с Рекомендациями ITU H.450.X.

Транспортировку сигнальных сообщений обеспечивает протокол с установлением соединения и с гарантированной доставкой информации -Transport Control Protocol (TCP). В соответствии с первой и второй версиями рекомендации H.323 для каждого нового вызова открывается отдельный сигнальный канал. Начиная с третьей версии рекомендации H.323, один сигнальный канал H.225.0 может переносить сообщения, относящиеся к разным вызовам и имеющие разные метки соединения (call reference). Наличие такой возможности позволяет значительно уменьшить время установления соединения с участием шлюзов и объем передаваемой служебной информации.

1.2.4. Управляющий канал H.245

В рекомендации ITU-T H.245 определен ряд независимых процедур, которые должны выполняться для управления информационными каналами. К ним относятся процедуры:

- определения ведущего и ведомого устройств (Master/slave determination);
- обмена данными о функциональных возможностях (Capability Exchange);
- открытия и закрытия однонаправленных логических каналов (Logical Channel Signalling);
- открытия и закрытия двунаправленных логических каналов (Bidirectional Logical Channel Signalling);
- закрытия логических каналов (Close Logical Channel Signalling);

- определения задержки, возникающей при передаче информации от источника к приемнику и в обратном направлении (Round Trip Delay Determination);
- выбора режима обработки информации (Mode Request);
- сигнализации по петле, создаваемой для целей технического обслуживания оборудования (Maintenance Loop Signalling).

Для выполнения вышеуказанных процедур между оконечными устройствами или между оконечным оборудованием и устройством управления конференциями или привратником организуется управляющий канал H.245. При этом оконечное оборудование должно открывать один (и только один) управляющий канал для каждого соединения, в котором оно участвует. Примечательно, что терминалы, устройства управления конференциями, шлюзы и привратники могут участвовать одновременно в нескольких соединениях и, следовательно, открывать несколько управляющих каналов.

Перенос управляющей информации H.245 осуществляется протоколом TCP по нулевому логическому каналу, который должен быть постоянно открытым с момента организации канала H.245 и вплоть до его ликвидации. Следует отметить, что нормальные процедуры открытия и закрытия логических каналов, описываемые в этой главе, для управления нулевым логическим каналом не применяются.

По управляющему каналу H.245 передаются сообщения четырех категорий: запросы, ответы, команды и индикации. Получив сообщение-запрос, оборудование должно выполнить определенное действие и немедленно передать обратно сообщение-ответ. Получив сообщение-команду, оборудование также должно выполнить определенное действие, но отвечать на команду не должно. Сообщение-индикация служит для того, чтобы информировать о чем-либо получателя, но не требует от него ни ответа, ни каких бы то ни было действий.

Выводы

Рекомендации ITU-T, входящие в стандарт H.323, определяют порядок функционирования абонентских терминалов в сетях с разделяемым ресурсом, не гарантирующих качества обслуживания (QoS). Стандарт H.323 не связан с протоколом IP, однако, большинство реализаций основано на этом протоколе. Набор рекомендаций определяет сетевые компоненты, протоколы и процедуры, позволяющие организовать мультимедиа-связь в пакетных сетях. Стандарт H.323 определяет четыре основных компонента, которые вместе с сетевой структурой позволяют проводить двусторонние (точка-точка) и многосторонние (точка — много точек) мультимедиа-конференции.

Рекомендация H.323 предусматривает довольно сложный набор протоколов, который предназначен не просто для передачи речевой информации по IP-сетям с коммутацией пакетов. Его цель – обеспечить работу мультимедийных приложений в сетях с негарантированным качеством обслуживания. Речевой трафик – это только одно из приложений H.323, наряду с видеоинформацией и данными. Достижение совместимости H.323 с различными мультимедийными приложениями требует значительных усилий.

Сети на базе протоколов H.323 ориентированы на интеграцию с телефонными сетями и могут рассматриваться как сети ISDN, наложенные на сети передачи данных. Процедура установления соединения в таких сетях IP-телефонии базируется на рекомендации Q.931 и аналогична процедуре, используемой в сетях ISDN.

2. РАЗРАБОТКА МЕТОДИЧЕСКОГО УКАЗАНИЯ В ПЕЧАТНОМ ВИДЕ

2.1. Разработка условия задачи

Для закрепления студентами знаний по дисциплине “NGN” в области построения сетей на базе стандарта H.323, разработано практическое занятие, заключающееся в решении следующей задачи.

Предприятие имеет 3 филиала. В каждом из филиалов имеется некоторое количество телефонных аппаратов, персональных компьютеров, а так же IP видеотелефонов для проведения видеоконференций между филиалами (все данные приведены в таблице 2.1). При этом первый и второй филиал находятся в одном и том же городе на определенном расстоянии друг от друга, а третий филиал находится в регионе. Необходимо построить сеть IP телефонии предприятия на базе стандарта H.323. Перечислить услуги, которые предоставляются сетью. Указать транспортные технологии, которые будут использоваться для связи между филиалами.

2.2. Разработка вариантов заданий

В качестве предприятия в данной задаче будет ТУИТ и его филиалы. Первый филиал во всех вариантах - ТУИТ. Второй филиал – его подразделения в городе Ташкенте, и третий филиал – подразделения в регионе. Все данные приведены в таблице 2.1.

Таблица 2.1

Варианты исходных данных.

№	Филиал 1 (ТУИТ)			Филиал 2 (город Ташкент)				Филиал 3			
	Кол-во ТА	Кол-во ПК	Кол-во IP видео-телефонов	Положение 2-го Ф. относительно 1-го	Кол-во ТА	Кол-во ПК	Кол-во IP видео-телефонов	Город	Кол-во ТА	Кол-во ПК	Кол-во IP видео-телефонов
1	8	8	4	В пределах 1 АТС	4	5	2	Самарканд	3	5	-
2	6	9	2	В пределах 1 АТС	3	6	-	Ургенч	4	2	1
3	8	4	-	Различные АТС г. Ташкента	5	3	-	Нукус	2	5	-
4	5	7	3	Различные АТС г. Ташкента	6	4	1	Фергана	3	4	-
5	4	8	-	В пределах 1 АТС	3	3	-	Нукус	5	3	-
6	9	5	2	В пределах 1 АТС	5	4	-	Карши	4	3	1
7	6	6	4	Различные АТС г. Ташкента	4	6	2	Ургенч	3	6	-

Таблица 2.1

Варианты исходных данных (продолжение)

№	Филиал 1 (ТУИТ)			Филиал 2 (город Ташкент)				Филиал 3			
	Кол-во ТА	Кол-во ПК	Кол-во IP видео- телефонов	Положение 2-го Ф. относительно 1-го	Кол-во ТА	Кол-во ПК	Кол-во IP видео- телефонов	Город	Кол-во ТА	Кол-во ПК	Кол-во IP видео- телефонов
8	8	6	-	Различные АТС г. Ташкента	6	2	-	Фергана	5	4	-
9	5	9	2	В пределах 1 АТС	2	5	-	Карши	3	3	1
10	6	4	-	В пределах 1 АТС	4	3	-	Самарканд	4	4	-

2.3. Формирование алгоритма решения задачи

Для решения данной задачи необходимо построить сеть внутри каждого филиала, затем выбрать технологии, соединяющие данные сети. Сеть внутри филиала будет строиться на базе технологии Ethernet, поскольку данная технология использует протокол IP, недорогая в построении и эксплуатации. Сеть Ethernet будет строиться по топологии «звезда» с использованием Хаба. К нему подключаются следующие устройства: персональные компьютеры, видеотелефоны, маршрутизаторы для связи с другими филиалами, привратник, сервер видеоконференций, и шлюз IP/ТфОП. Привратник подключается централизованно в первом филиале. Сервер видеоконференций, так же подключается централизованно в первом филиале, если существует необходимость в предоставлении услуг видеоконференций. Шлюз IP/ТфОП используется в случае использования обычных телефонных аппаратов на сети H.323.

Для связи между филиалами могут быть использованы различные технологии. В случае небольшого удаления филиалов друг от друга можно использовать технологию Ethernet. Если филиалы находятся в пределах одного города, но на большом расстоянии (свыше 1 – 2 км), можно задействовать кабельные сооружения ГТС при помощи кабельного маршрутизатора, используя поток E1 для передачи информации. Для связи между регионами целесообразно использовать оптоволоконные линии.

2.4. Разработка контрольных вопросов

Для закрепления навыков студентов по данному практическому занятию и усвоению теоретического материала необходимо составить контрольные вопросы.

Контрольные вопросы:

1. Укажите основные преимущества IP телефонии перед обычной телефонией.
2. Перечислите основные устройства сети стандарта H.323.
3. Перечислите основные протоколы стандарта H.323.
4. Какое основное назначение устройств MCU?
5. Какие функции может выполнять шлюз?
6. Какие алгоритмы кодирования аудио и видео применяются в терминале H.323?
7. Укажите основные функции привратника.
8. Объясните назначение MC и MP.
9. Укажите основные функции протокола RAS.
10. Укажите наиболее часто используемые сообщения сигнального канала H.225.0.
11. Укажите основные процедуры, выполняемые управляющим каналом H.245.

Выводы

Задача построения сети IP-телефонии на основе стандарта H.323, является актуальной для современного рынка телекоммуникационных услуг республики. Она призвана подготовить молодых специалистов к работе по проектированию сетей на производстве. Она способствует закреплению теоретического материала, полученного на лекциях. Так же практические занятия призваны помочь преподавателю оценить степень усвоения студентом теоретического материала.

Привязка к конкретным организациям и филиалам позволяет студентам лучше понять принципы построения сети IP-телефонии в зависимости от расположения филиалов друг относительно друга. Так же важным фактором является обучение правильному построению сетей внутри филиала в зависимости от конкретных задач, и понимание принципов функционирования данной сети.

В данной задаче так же нет привязки к оборудованию конкретных производителей, задача решается в общем виде, что так же способствует более общему охвату принципов построения сети на базе стандарта H.323. В дальнейшем можно развить задачу исходя из конкретных примеров оборудования, но сделать это нужно лишь после решения задачи в общем виде. А в примере с конкретным оборудованием можно включить экономическую составляющую, т.е. выбор наиболее оптимальной конфигурации оборудования при минимальном бюджете.

3. РАЗРАБОТКА ВИРТУАЛЬНОГО ПРАКТИЧЕСКОГО ЗАНЯТИЯ «ПОСТРОЕНИЕ СТЕЙ НА БАЗЕ ПРОТОКОЛА H.323» ДИСЦИПЛИНЫ NGN

3.1. Описание средств разработки приложения

Разработка мультимедийного практического занятия по построению сети H.323 выполняется в программе Adobe Flash CS3. Данная программа используется для разработки приложений, ориентированных на использование в сети Интернет. Данная среда разработки очень удобно сочетает в себе возможности графического редактора и языка программирования. Это позволяет одновременно, не меняя среду разработки, заниматься составлением алгоритма работы программы и оформлением графического интерфейса.

Программа Adobe Flash CS3 имеет встроенный язык программирования Action Script 3, который является языком программирования, оптимизированный для работы с мультимедиа и имеющий сходство с C# и Java Script. Интерфейс программы изображен на рис. 3.1.

Рабочая область программы служит для размещения на ней графических элементов, текста, а также элементов управления. Она разделена на слои и кадры. Слои могут содержать в себе множество различных графических элементов, и при этом верхние слои перекрывают нижние. Кадры служат для анимации и могут быть обычными и ключевыми. В обычных кадрах невозможно изменение объектов. Они могут служить в качестве промежуточных при плавном изменении свойств символа. В ключевых кадрах можно менять свойства символов, содержимое слоя, а также размещать текст программы. Слои и кадры существуют как в общем проекте, так и в каждом символе отдельно. Помимо этого проект разделен на сцены, которые идут друг за другом в указанном порядке.

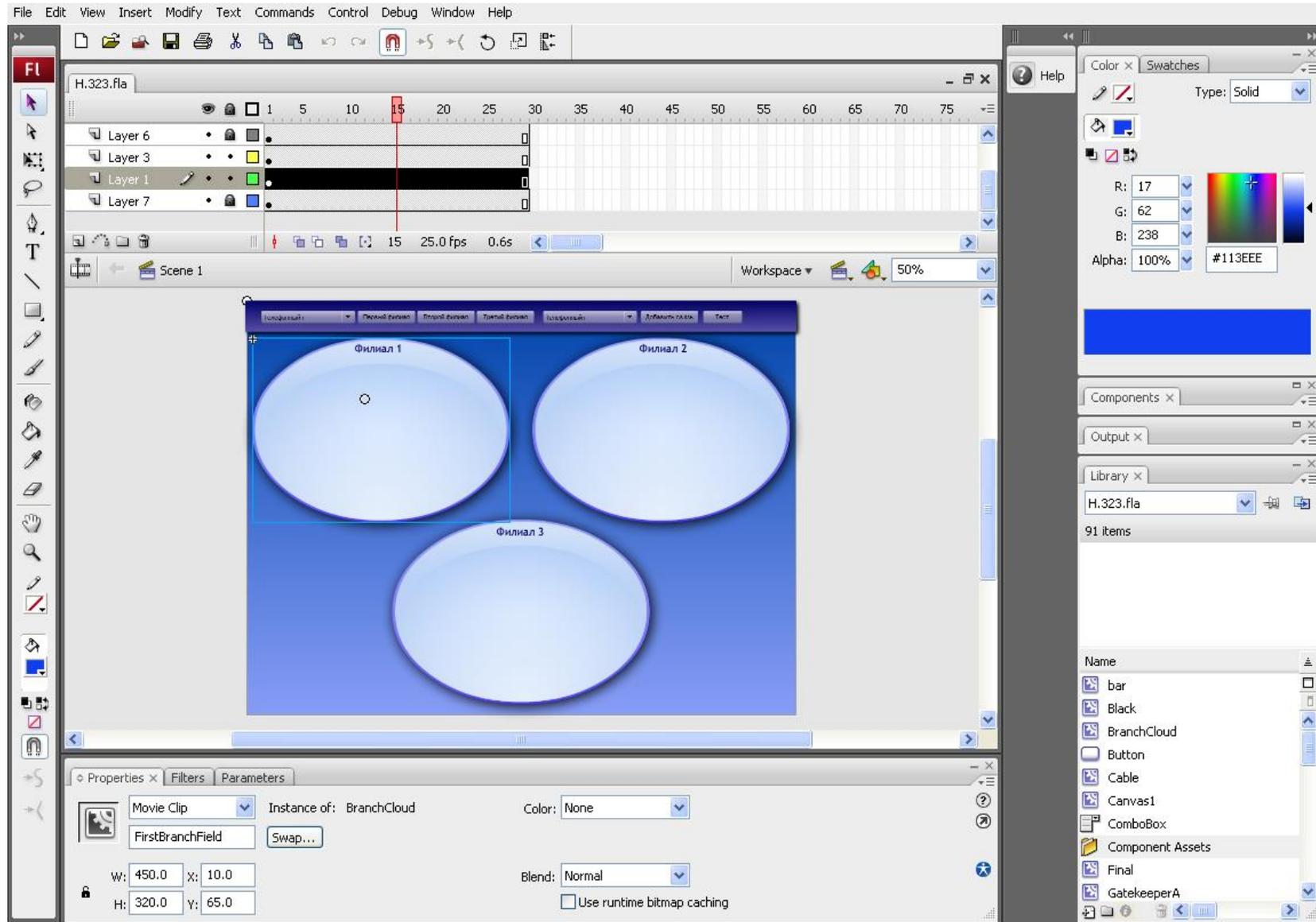


Рис. 3.1. Интерфейс программы Adobe Flash

Графические элементы в программе называются символами. Символы хранятся в библиотеке, прикрепленной к каждому файлу проекта Flash (файлы имеют расширение .FLA). Символ может быть как просто графическим или анимационным элементом, так и интерактивным, если он имеет программный код.

В более ранних версиях Adobe Flash текст программы, отвечающей за работу символа мог быть размещен на самом символе, что вызывало определенные трудности при отладке программы, поскольку нужно было искать в сценах все символы. Начиная с версии CS3, тексты программ в основном размещаются в отдельных файлах с расширением .AS, которые ассоциируются с символами при задании их свойств и обычно являются описанием целого класса символов, как это принято в языках программирования высокого уровня.

Большим преимуществом программ, написанных на Flash, является их интерактивность, небольшой размер при качественном графическом оформлении за счет использования векторной графики и возможности публикации их в сети Интернет непосредственно на веб-странице и работа с ними в on-line режиме, поскольку все современные браузеры имеют Flash Player, необходимый для воспроизведения файлов .SWF (скомпилированный проект Flash). Так же возможна компиляция проекта сразу в исполняемый .EXE файл с интегрированным Flash Player. В этом случае размер файла будет увеличен на 2,2 Мб, но при этом сможет работать на любом компьютере с операционной системой Windows 2000, XP, Vista, 7.

3.2. Разработка структуры программы

Программа для построения сети на базе стандарта H.323 состоит из файла проекта H.323.FLA, который объединяет вместе все классы программы, находящиеся в каталоге Sub. Далее следует краткое описание классов, описанных в указанных файлах (полный текст программы смотрите в приложении):

Variant.as – содержит данные о вариантах, которые служат для проверки правильности выполненной работы.

DrawingCanvas.as – описывает структуру рабочего поля, порядок создания, удаления, взаимодействия и проверки устройств.

Device.as – описывает общий класс устройств, их общие функции и характеристики (сюда относятся и описывает транспортные технологии, и сетевые устройства).

Hardware.as – является расширением класса Device, описывает сетевые устройства, порядок их функционирования.

Cable.as – является расширением класса Device, описывает транспортные технологии, порядок их функционирования.

Port.as – является вспомогательным классом для класса Hardware, служащий для взаимодействия Hardware и Cable и указывающий количество и тип портов устройства.

3.3. Разработка блок-схемы программы

Данная программа, как и большинство нынешних программ, является ориентированной на события, т.е. запуск программы или отдельных ее модулей происходит при определенном событии, например при щелчке мышью или при нажатии клавиши. Поэтому целиком рисовать блок-схему для программы не совсем правильно. Правильнее будет рисовать блок-схему для реакции на основные события.

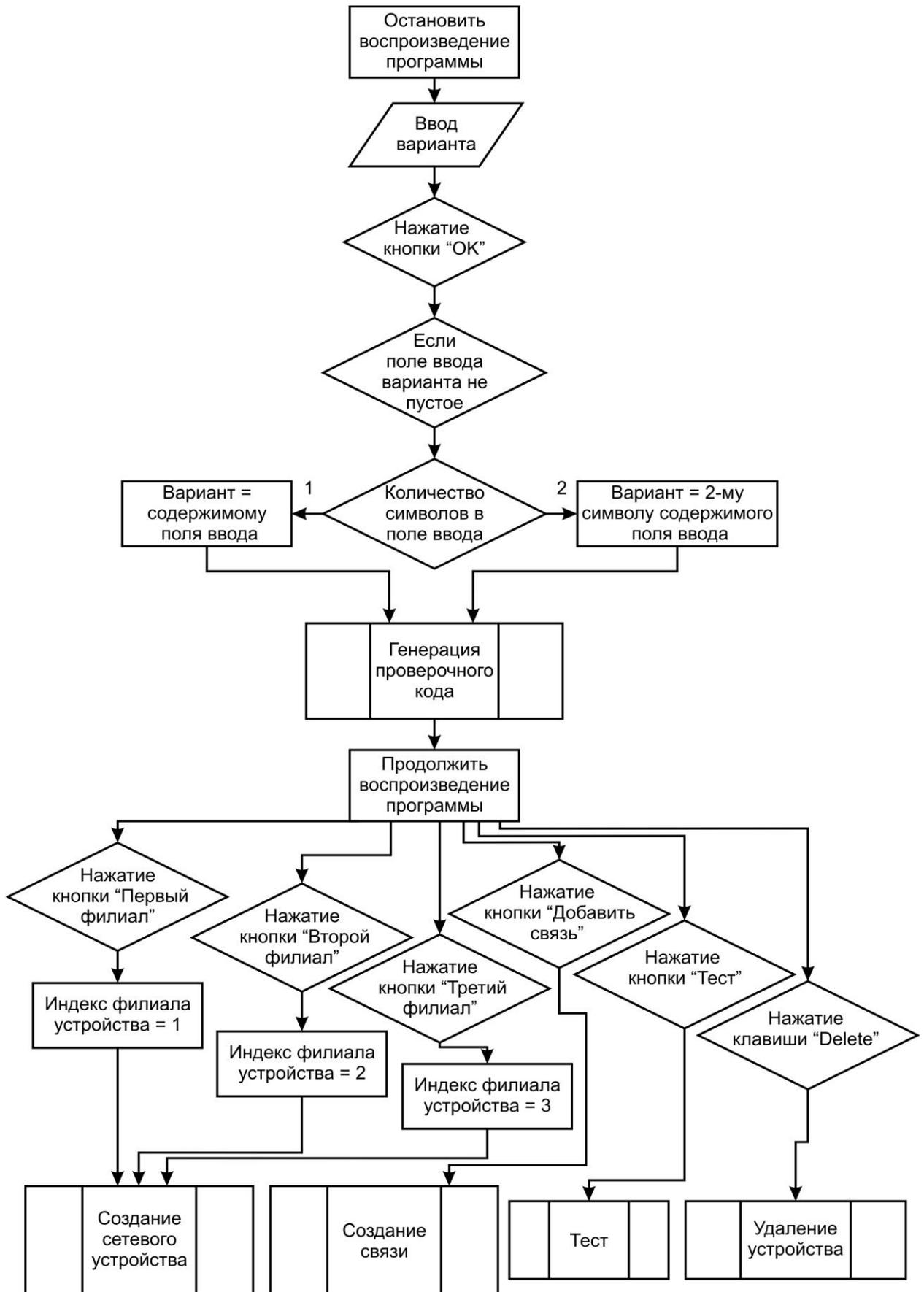


Рис. 3.2. Блок-схема для первого кадра проекта



Рис. 3.3. Блок-схема создания сетевого устройства

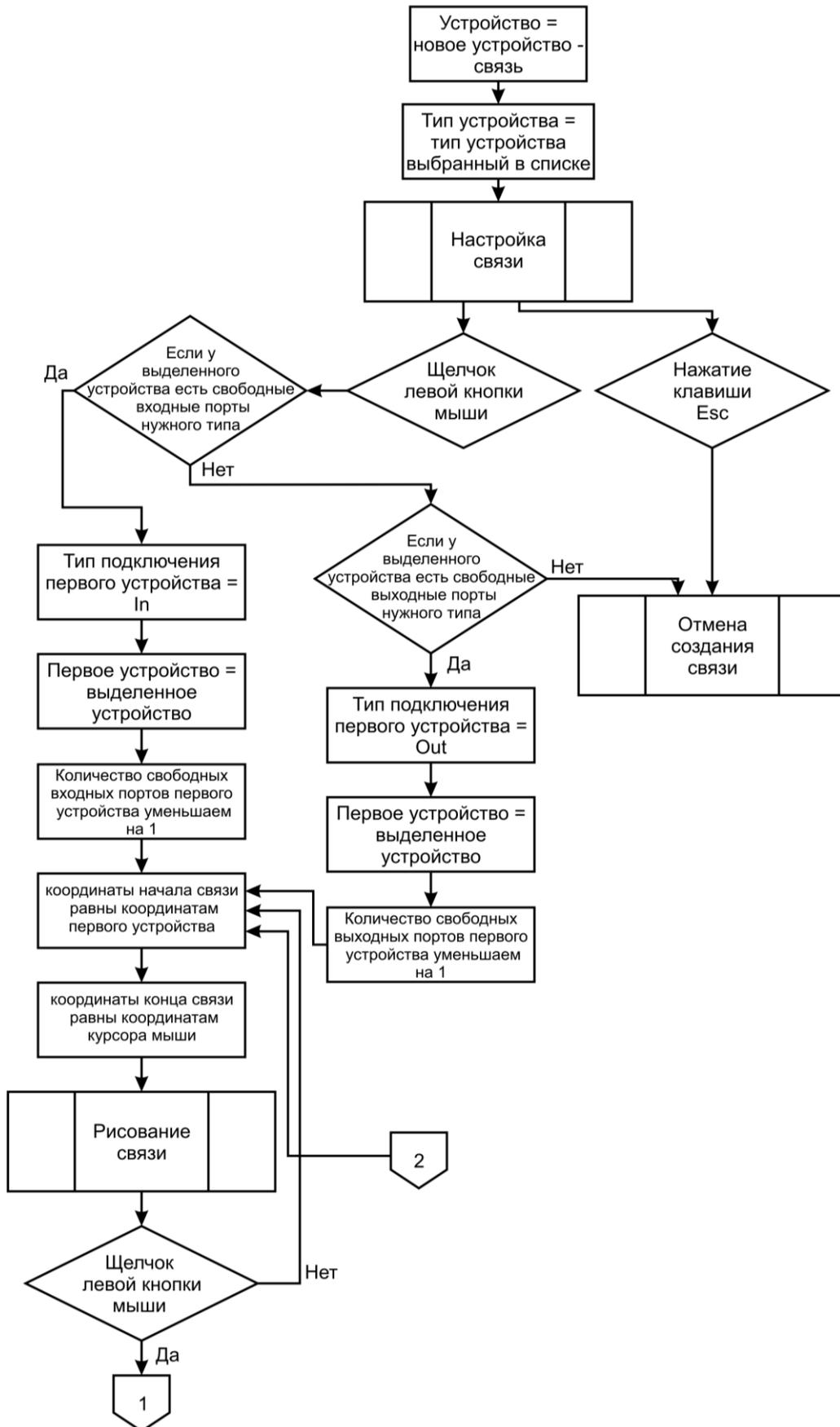


Рис. 3.4. Блок-схема создания связи между устройствами

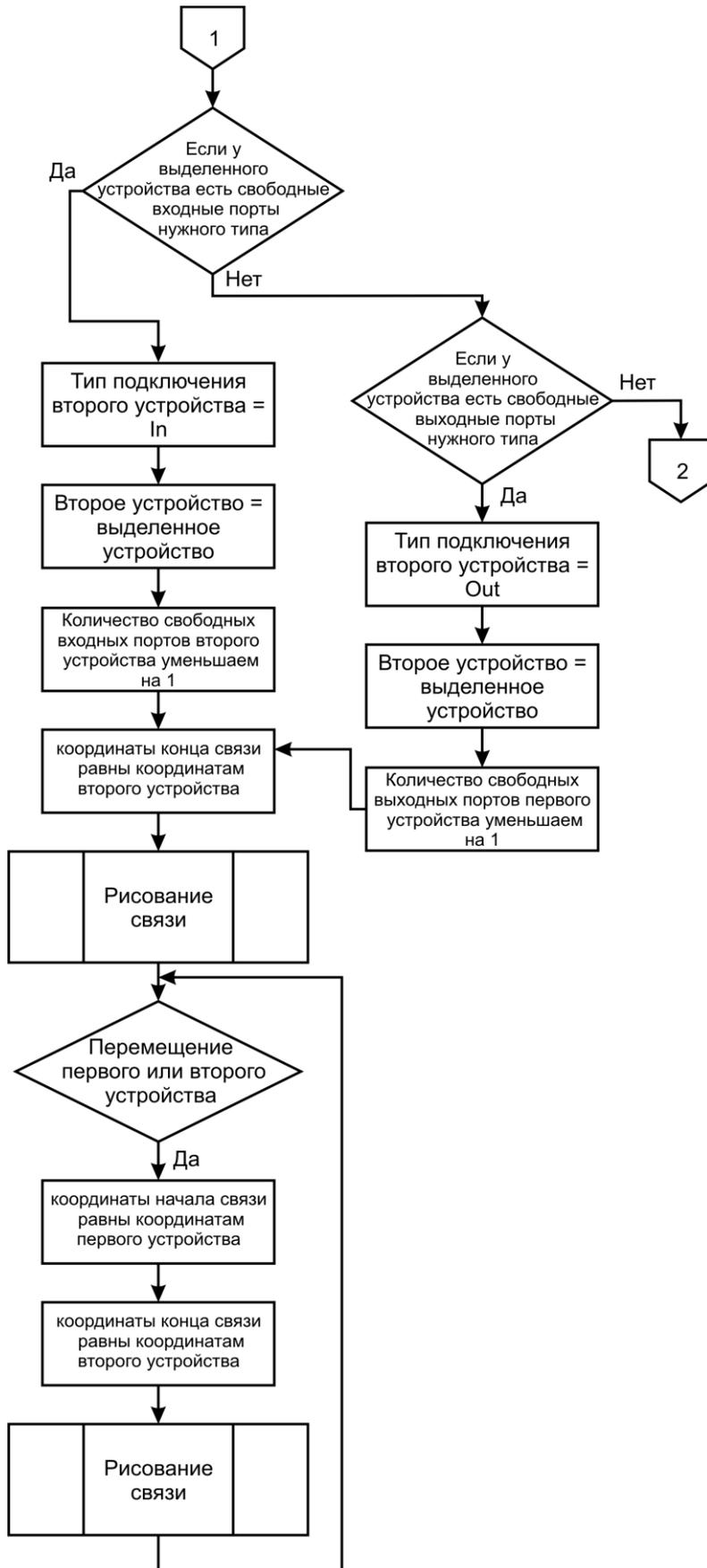


Рис. 3.5. Блок-схема создания связи между устройствами (продолжение)

3.4. Разработка графического интерфейса

Графический интерфейс программы разработан так же в программе Adobe Flash, и базируется на принципах векторной графики. Существует 2 типа представления графики в цифровом виде: растровый и векторный. Растровые изображения обычно используются для фотографий, и они описываются как прямоугольная матрица из пикселей различных цветов. Обычно они имеют большой размер. В векторной графике изображение представлено в виде набора кривых Безье, базирующихся на контрольных точках и касательных. Данный вид графики не пригоден для фотографий, но хорошо подходит для чертежей, схем, нефотореалистичных изображений и символов. При этом размер графических файлов мал, что делает данные изображения удобными для использования в сети Интернет.

Сам же интерфейс программы прост и удобен в использовании (рис 3.7). В верхней части программы расположена главная панель. В нижней части – рабочая область, разделенная на 3 филиала.

На главной панели расположены следующие элементы управления:

1. Элемент ComboBox со списком сетевых элементов. Для добавления элемента в филиал необходимо выбрать нужный элемент, затем нажать на кнопки филиалов.
2. Кнопки филиалов – добавляют выбранные устройства в указанный филиал.
3. Элемент ComboBox со списком транспортных технологий. Для добавления транспортной технологии нужно нажать на кнопку «Добавить связь».
4. Кнопка «Добавить связь» - начинает создание указанной транспортной технологии между выбранными сетевыми устройствами. Чтобы создать связь, необходимо кликнуть левой кнопкой мыши на первом устройстве, затем на втором. Связь будет создана при наличии свободных портов указанного типа у данного устройства. В процессе создания устройства рядом с кнопкой будет гореть индикатор. Создание связи можно отменить, нажав клавишу Esc.
5. Кнопка «Тест» - проверяет сеть на корректность построения в соответствии с вариантом. Если вариант решен верно, то появляется надпись «Задание выполнено», если же нет, то происходит просто активация / деактивация устройств.

Рабочая область программы позволяет выделять, перетаскивать сетевые элементы в пределах их филиала. Выделенное устройство или связь подсвечивается синим цветом. Выделенные устройства можно удалять при помощи клавиши Delete. Если выделенное устройство – связь между сетевыми элементами, то удаляется только связь, а если – сетевой элемент, то удаляется само устройство и все связи с ним.

При активации устройства в процессе тестирования связи меняют цвет на более яркий, а сетевые элементы проявляют внешние признаки активности, схожие с реальными прототипами (загорается экран, мигают индикационные лампочки и т. д.).

3.5. Пример решения задачи

После запуска программы предлагается ввести вариант (рис 3.8). В нашем случае это будет вариант № 1. Нажимаем ОК. Появляется рабочее поле. Итак, приступим к созданию сети.

Для варианта 1 начальные условия следующие:

Количество телефонных аппаратов в первом филиале – 8;

Количество персональных компьютеров в первом филиале – 8;

Количество IP-видеотелефонов в первом филиале – 4;

Расположение первого и второго филиала относительно друг друга - В пределах 1 АТС;

Количество телефонных аппаратов во втором филиале – 4;

Количество персональных компьютеров во втором филиале – 5;

Количество IP-видеотелефонов во втором филиале – 2;

Расположение третьего филиала - город Самарканд;

Количество телефонных аппаратов во втором филиале – 3;

Количество персональных компьютеров во втором филиале – 5;

Количество IP-видеотелефонов во втором филиале – 0;

После ввода варианта приступим к созданию самой сети. Для начала создадим устройства, указанные в условии (рис. 3.9). Для этого выбираем из списка необходимое устройство, затем добавляем его в нужный филиал соответствующей кнопкой.

После создания терминальных устройств необходимо определиться с конфигурацией сети. Внутри филиала сеть строиться по топологии «звезда» вокруг хаба. К нему подключаются все IP терминалы, а так же шлюз IP/ТфОП, привратник, сервер видеоконференций и маршрутизаторы.

В данной сети привратник ставится только в один филиал – в первый. Использование видеотелефонов предполагает ведение видеоконференций, поэтому необходимо установить сервер видеоконференций. Сервер так же устанавливается централизованно.

Шлюз IP/ТфОП устанавливается в каждом филиале и служит для подключения обычных телефонных аппаратов к сети H.323.

В качестве транспортных технологий между первым и вторым филиалом будет использоваться Ethernet, поскольку они расположены достаточно близко друг от друга. А между первым и третьим филиалами целесообразно использовать оптоволоконные линии связи с соответствующим маршрутизатором. В результате конечная сеть будет выглядеть так, как показано на рис. 3.10

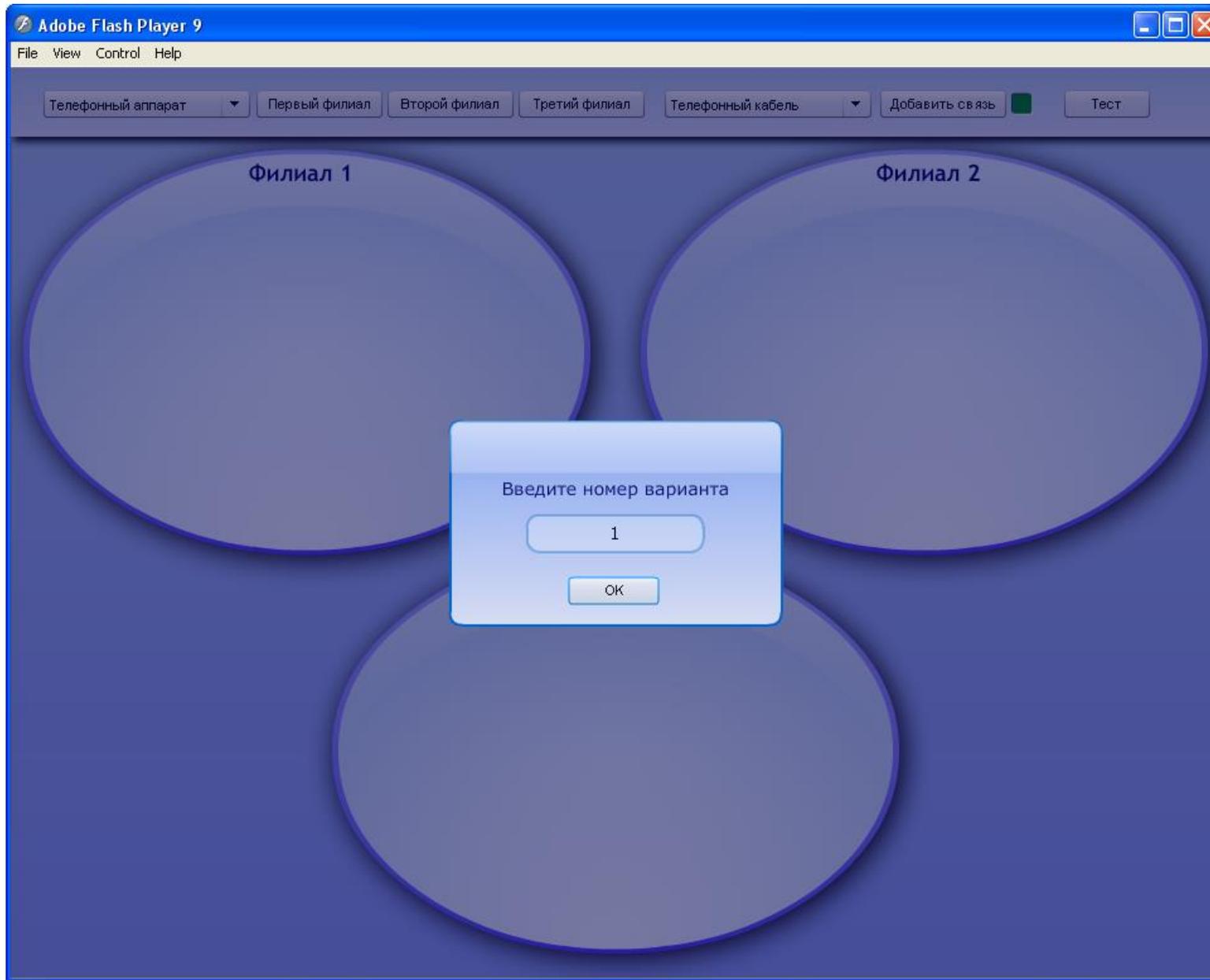


Рис. 3.8. Ввод варианта

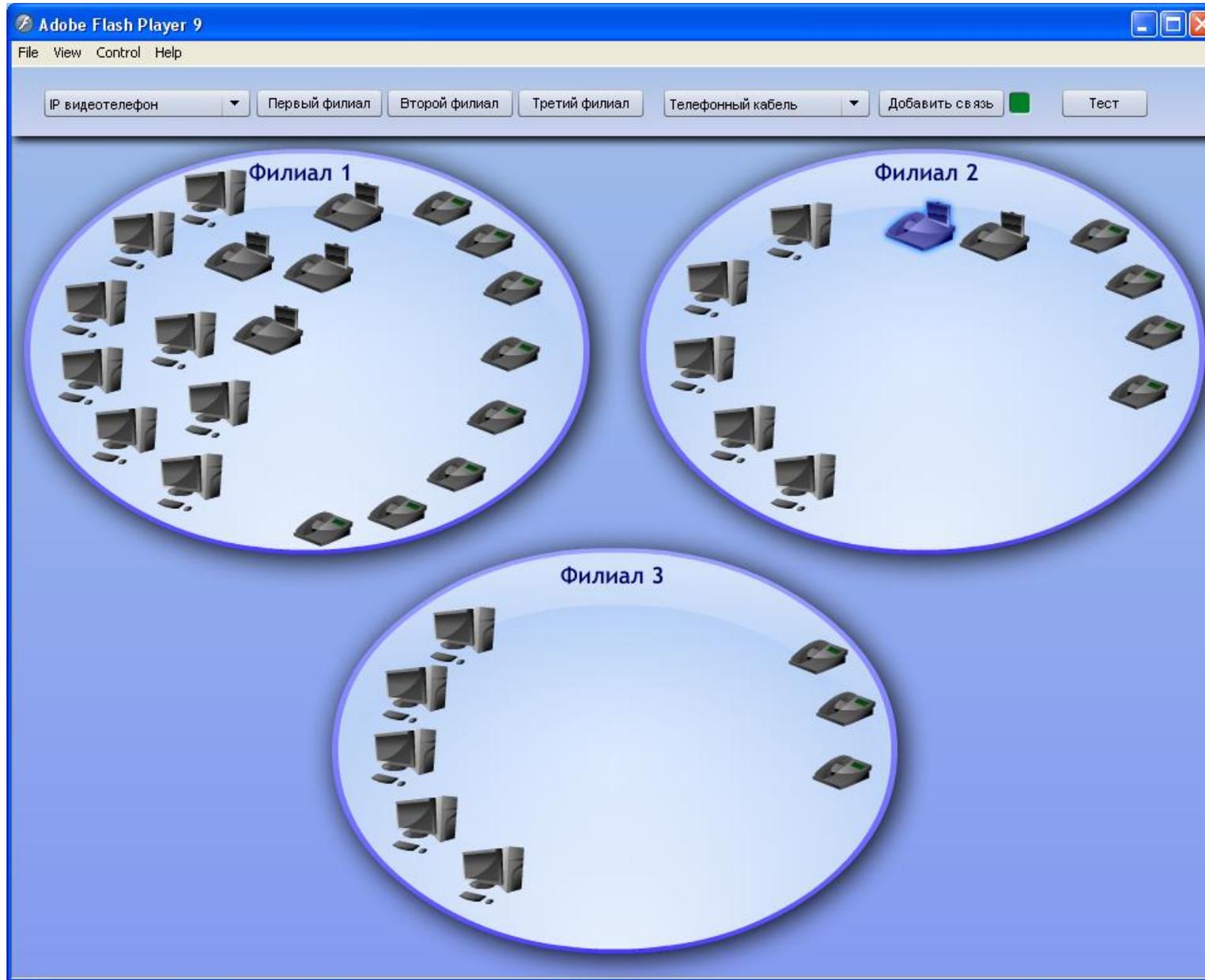


Рис. 3.9. Добавление терминальных устройств

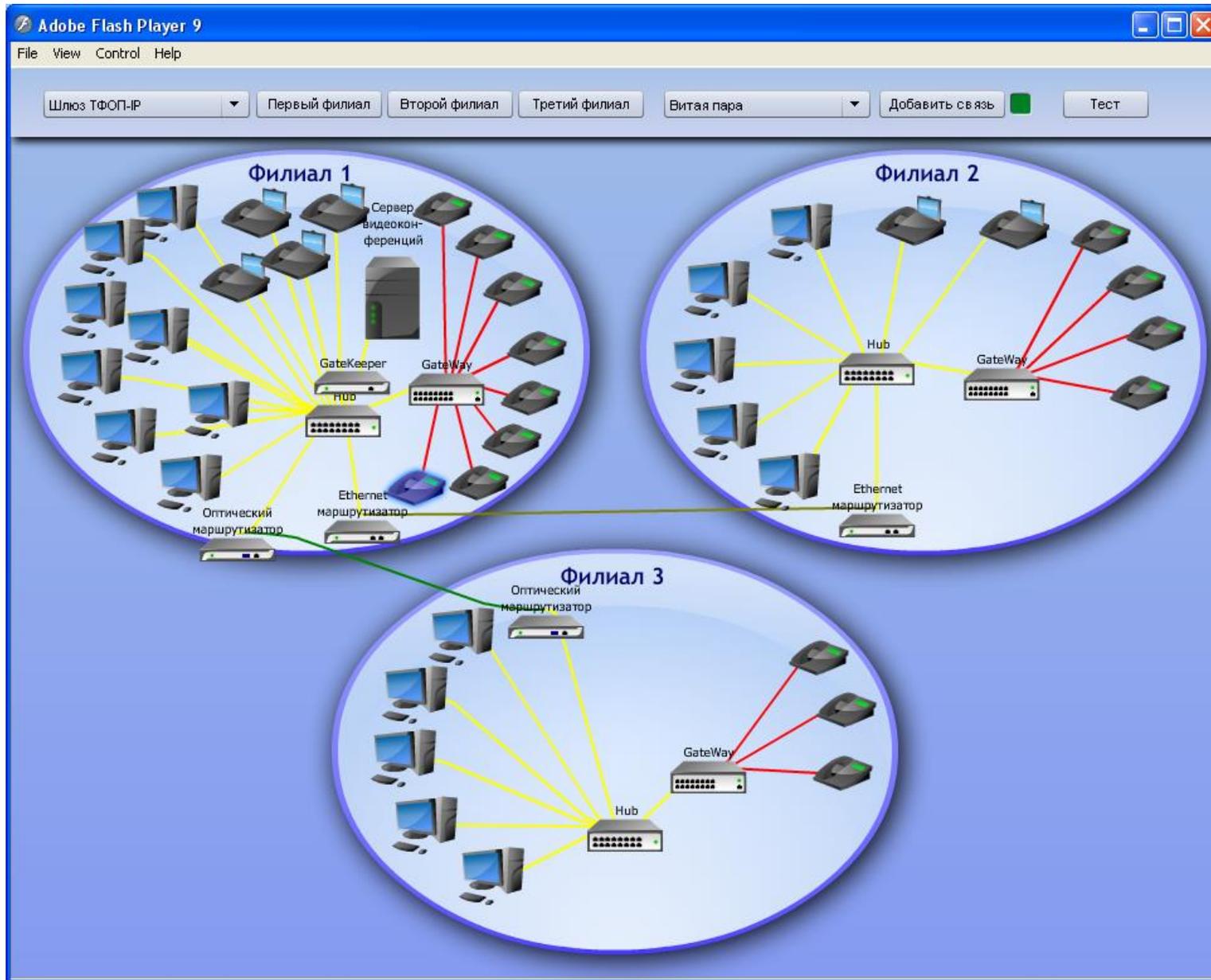


Рис. 3.10. Готовая сеть

Выводы

Данная программа позволяет строить сеть на базе стандарта H.323 и проверять правильность построения в зависимости от начальных условий. Это в свою очередь позволяет экономить время и силы преподавателя, затраченное на проверку практических заданий. Так же в отличие от бумажного варианта практического занятия, программа позволяет наглядно видеть функционирование сети.

Использование при разработке данной программы среды Adobe Flash делает ее более мультимедийной и яркой в плане дизайна без увеличения размера самой программы, чем ее аналоги, созданные в других средах разработки.

Программная часть является основанной на событиях и состоящей из нескольких модулей и классов. Код программы является достаточно сложным, поскольку необходимо иметь дело с множеством динамических объектов, взаимодействующих между собой. Это отличает данную программу от других аналогичных виртуальных практических и лабораторных занятий, которые мне приходилось использовать во время учебы. В данных программах всегда набор объектов был статическим, что упрощало построение программы.

В общей сложности код программы содержит около 1200 строк. На рисунках 3.2 – 3.6 показаны блок-схемы основных событий. Показывать блок-схемы всех подпрограмм не имеет смысла, поскольку это заняло бы около 30 страниц.

В настоящее время данная работа не имеет аналогов в пределах ТУИТ, и имеет огромный потенциал для развития вплоть до создания полноценного конструктора сетей наподобие OPNET.

4. ОХРАНА ТРУДА И ТЕХНИКА БЕЗОПАСНОСТИ

4.1. Меры безопасности при работе за компьютером

Введение

Ни для кого не секрет, что компьютеризацию сегодня принято считать панацеей – только компьютер может повысить эффективность образования и промышленности, банковского дела и торговли, объединить через Интернет весь мир. И, очевидно, эта "пандемия" неостановима. Как всякий новый этап в развитии общества, компьютеризация несет с собой и новые проблемы. И одна из наиболее важных – экологическая.

У экологической проблемы компьютеризации две составляющие. Первая определяется физиологическими особенностями работы человека за компьютером. Вторая – техническими параметрами средств компьютеризации. Эти составляющие – "человеческая" и "техническая" – тесно переплетены и взаимозависимы. Исследования подобных проблем – предмет эргономики, науки о взаимодействии человека, основной целью которой является создание совершенной и безопасной техники, максимально ориентированной на человека, организация рабочего места, профилактика труда. Эргономика изучает трудовую деятельность в комплексе, в ней объединяются научные дисциплины, развивавшиеся прежде независимо друг от друга. Поэтому задачу этого реферата можно сформулировать так – анализ эргономической безопасности компьютера и методы ее обеспечения.

Почему работа за компьютером наносит вред здоровью человека

То, что работа за компьютером вовсе не безопасна, раньше других почувствовали в самой компьютеризированной стране мира - Соединенных Штатах Америки. Эпидемия "белых воротничков" охватила США, нанося вред здоровью многих людей. По данным Министерства труда, одни только так называемые "повторяющиеся травмирующие воздействия при работе с компьютером" обходятся корпоративной Америке в 100 млрд. долларов ежегодно. Компенсации, выплаченные служащим, достигают астрономических размеров, а некоторым пострадавшим от работы за компьютером приходится расплачиваться жестокими болями в течение всей жизни.

4.1.1. Зрительная работа за компьютером и ее последствия

Основная нагрузка при работе за компьютером приходится на глаза. Их утомляемость во многом зависит не только от качества изображения на экране, но и от общей освещенности помещения. Согласно гигиеническим нормам освещенность на поверхности стола и клавиатуре должна быть не менее 300 люкс, а вертикальная освещенность экрана - всего 100-250 люкс. Исследования физиологов и гигиенистов убедительно доказали, что и полутьма, и слишком высокая освещенность экрана приводят к быстрому зрительному утомлению.

Размещать компьютер рекомендуется так, чтобы свет (естественный или искусственный) падал сбоку, лучше слева, это избавит вас от мешающих теней и поможет снизить освещенность экрана.

В качестве источников освещения рекомендуется применять люминесцентные лампы типа ЛБ со светильниками серии ЛПОЗ6 с зеркализированными решетками. Лампы накаливания лучше использовать для местного освещения зоны рабочего документа (клавиатуры, книги, тетради). Совет домашним пользователям: постарайтесь, чтобы люстра в вашей рабочей комнате имела закрытые снизу светильники, так чтобы на экран монитора падал рассеянно-отраженный свет. Это избавит вас от бликов и облегчит зрительную работу. А вот настольная лампа, наоборот, должна иметь плотный, непросвечивающий абажур, направляющий свет прямо в зону рабочего документа.

Уже в первые годы компьютеризации было отмечено специфическое зрительное утомление у пользователей дисплеев, получившее общее название "компьютерный зрительный синдром" (CVS-Computer Vision Syndrome). Причин его возникновения несколько. И прежде всего – сформировавшаяся за миллионы лет эволюции зрительная система человека, которая приспособлена для восприятия объектов в отраженном свете (картин природы, рисунков, печатных текстов и т. п.), а не для работы с дисплеем. Изображение на дисплее принципиально отличается от привычных глазу объектов наблюдения – оно светится; состоит из дискретных точек; оно мерцает, т. е. эти точки с определенной частотой зажигаются и гаснут; цветное компьютерное изображение не соответствует естественным цветам (спектры излучения люминофоров отличаются от спектров поглощения зрительных пигментов в колбочках сетчатки глаза, которые ответственны за наше цветовое зрение). Но не только особенности изображения на экране вызывают зрительное утомление. При работе на компьютере часами у глаз не бывает необходимых фаз расслабления, глаза напрягаются, их работоспособность снижается. Большую нагрузку орган зрения испытывает при вводе информации, так как пользователь вынужден часто переводить взгляд с экрана на текст и клавиатуру, находящиеся на разном расстоянии и по-разному освещенные.

В чем же выражается зрительное утомление? Сегодня уже миллионы пользователей жалуются на затуманивание зрения, трудности при переносе взгляда с ближних на дальние и с дальних на ближние предметы, кажущееся изменение окраски предметов, их двоение, неприятные ощущения в области глаз – чувство жжения, "песка", покраснение век, боли при движении глаз.

4.1.2. Чистота

Пыль, оседающая на экране, снижает его освещенность, ухудшает видимость изображения и способствует накоплению статического электричества. Лучше приучить себя всякий раз, начиная работу, еще до включения компьютера протирать экран мягкой неворсистой тканью.

Согласно СанПиНу в помещениях с ПЭВМ должна ежедневно проводиться влажная уборка, поэтому мы не рекомендуем размещать компьютеры в комнатах с коврами, ковровым покрытием и мягкой мебелью, иначе приходится выдвигать малореальное в современных условиях требование ежедневной их очистки с помощью моющего пылесоса.

4.1.3. Микротравмы

Микротравма - это постепенный износ организма в результате ежедневных нагрузок. Большинство нарушений в организме происходит из-за накапливающихся микротравм. Такой тип

повреждений не возникает вдруг, как перелом руки или ноги (макротравма). Прежде, чем вы почувствуете боль, может пройти несколько месяцев сидения в неправильной позе или повторяющихся движений. Боль может ощущаться по-разному: в виде жжения, колющей или стреляющей боли, покалывания.

Заболевания, вызванные повторяющимися нагрузками и повторяющиеся травмирующие воздействия при работе с компьютером.

Повторяющиеся действия приводят к накоплению продуктов распада в мышцах. Эти продукты и вызывают болезненные ощущения. Очень трудно предотвратить повторяющиеся движения кистей и ладоней при работе на компьютере, однако регулярные перерывы и упражнения на растягивание мышц могут предотвратить заболевания, вызванные повторяющимися нагрузками и повторяющиеся травмирующие воздействия при работе с компьютером.

Профилактические методы уменьшения усталости при работе за компьютером

Основанное на принципах ErgAerobics, программа профилактики травм, значительно снижает вероятность стать жертвой повторяющиеся травмирующие воздействия при работе с компьютером. Эрг-аэробика, разработанная двумя специалистами в области лечебной физкультуры, или "медицинской эргономики", включает в наиболее новаторские принципы эргономики, профилактические и реабилитационные упражнения.

Назовем предлагаемые здесь упражнения "эрг-упражнениями" (ErgErcises). Эрг-упражнения повышают выносливость и продуктивность работы пользователя компьютера. Их можно выполнять в любое время и любом месте. Эрг-упражнения просты, не привлекают к себе внимания и занимают очень мало времени. Их можно выполнять в перерывах на кофе, разговаривая по телефону или ожидая на автобусной остановке. Они помогают человеку поддержать прекрасную физическую форму и незаметно становятся естественным способом выполнения работы - здоровым и безболезненным.

4.1.4. Технические методы увеличения безопасности работы за компьютером

Эргономичное аппаратное оборудование

Чтобы работа была комфортной и безопасной необходимо позаботиться об аппаратном оборудовании компьютера. Как правило, наибольший вред здоровью пользователя компьютера наносят устройства ввода-вывода: монитор, клавиатура, мышь.

В наше время, когда проблемы безопасности работы за компьютером стоят как нельзя остро, появляется множество различных стандартов на экологическую безопасность оборудования персонального компьютера. Современный монитор должен соответствовать по крайней мере трем общепринятым стандартам безопасности и эргономике:

FCC Class B - этот стандарт разработан канадской федеральной комиссией по коммуникациям для обеспечения приемлемой защиты окружающей среды от влияния радиопомех в замкнутом пространстве. Оборудование, соответствующее требованиям FCC Class B, не должно мешать работе теле- и радио аппаратуры.

MPR-II - этот стандарт был выпущен в Шведском национальным департаментом. MPR-II налагает ограничения на излучения от компьютерных мониторов и промышленной техники, используемой в офисе.

ТСО'95 (а также современный ТСО'99) - рекомендация, разработанная Шведской конференцией профсоюзов и Национальным советом индустриального и технического развития Швеции (NUTEK), регламентирует взаимодействие с окружающей средой. Она требует уменьшения электрического и магнитного полей до технически возможного уровня с целью защиты пользователя. Для того, чтобы получить сертификат ТСО'95 (ТСО'99), монитор должен отвечать стандартам низкого излучения (Low Radiation), т.е. иметь низкий уровень электромагнитного поля, обеспечивать автоматическое снижение энергопотребления при долгом не использовании, отвечать европейским стандартам пожарной и электрической безопасности

EPA Energy Star VESA DPMS - согласно этому стандарту монитор должен поддерживать три энергосберегающих режима - ожидание (stand-by), приостановку (suspend) и "сон" (off). Такой монитор при долгом простое компьютера переводится в соответствующий режим, с низким энергопотреблением.

Необходимо также чтобы монитор имел возможность регулировки параметров изображения (яркость, контраст и т.д.). Рекомендуется, чтобы при работе с компьютером частота вертикальной развертки монитора была не ниже 75Гц (при этом пользователь перестает замечать мерцание изображения, которое ведет к быстрому уставанию глаз).

В настоящее время многие фирмы производители мониторов начали массовый выпуск так называемых плоскпанельных мониторов (LCD), которые лишены многих экологических недостатков, присущих мониторам с электронно-лучевой трубкой, как то: электромагнитное излучение, магнитное поле, мерцание и т.д.

4.1.5. Устройства ввода информации

В отличие от мониторов для компьютерных устройств ввода (клавиатура и мышь) в настоящее время не имеется общепринятых и широко распространенных стандартов. В тоже время многие производители данного оборудования рекламируя свою продукцию, описывают различные конструктивные решения, повышающие эргономичность ее использования: клавиатура с возможностью регулирования расположение клавиш, мышь с формой, уменьшающей усталость кисти при длительной работе. Хотя некоторые из них стоит рассматривать только как броскую рекламу, многие модели действительно являются своеобразным технологическим скачком вперед с точки зрения безопасности работы за компьютером.

4.1.6. Эргономичная организация рабочего места

Даже самое эргономичное оборудование в мире не поможет вам избежать заболеваний, если использовать его неправильно. Следуя простым советам по эргономичной организации рабочего места, можно предотвратить дальнейшее развитие заболеваний.

4.1.7. Рабочее пространство

Научная организация рабочего пространства базируется на данных о средней зоне охвата рук человека - 35-40 см. Ближней зоне соответствует область, охватываемая рукой с прижатым к туловищу локтем, дальней зоне - область вытянутой руки.

4.1.8. Работа с клавиатурой

Неправильное положение рук при печати на клавиатуре приводит к хроническим растяжениям кисти. Важно не столько отодвинуть клавиатуру от края стола и опереть кисти о специальную площадку, сколько держать локти параллельно поверхности стола и под прямым углом к плечу. Поэтому клавиатура должна располагаться в 10-15 см (в зависимости от длины локтя) от края стола. В этом случае нагрузка приходится не на кисть, в которой вены и сухожилия находятся близко к поверхности кожи, а на более "мясистую" часть локтя. Современные, эргономичные модели имеют оптимальную площадь для клавиатуры за счет расположения монитора в самой широкой части стола. Глубина стола должна позволять полностью положить локти на стол, отодвинув клавиатуру к монитору.

4.1.9. Расположение монитора

Монитор, как правило, располагается чрезмерно близко. Существует несколько научных теорий, по разному определяющих значимые факторы и оптимальные расстояния от глаза до монитора. Например, рекомендуется держать монитор на расстоянии вытянутой руки. Но при этом что человек должен иметь возможность сам решать, насколько далеко будет стоять монитор.

Именно поэтому конструкция современных столов позволяет менять глубину положения монитора в широком диапазоне. Верхняя граница на уровне глаз или не ниже 15 см ниже уровня глаз.

4.1.10. Внутренний объем

Значимым фактором является под пространство столешницей. Высота наших столов соответствует общепринятым стандартам, и составляет 74 см. Также необходимо учесть, что пространства под креслом и столом должно быть достаточно, чтобы было удобно сгибать и разгибать колени.

4.1.11. Кресло

Казалось бы, требования к нему сформулировать предельно просто, - оно должно быть удобным. Но это еще не все. Кресло должно обеспечивать физиологически рациональную рабочую позу, при которой не нарушается циркуляция крови и не происходит других вредных воздействий. Кресло **обязательно** должно быть с подлокотниками и иметь возможность поворота, изменения высоты и угла наклона сиденья и спинки. Желательно иметь возможность регулировки высоты и расстояния между подлокотниками, расстояния от спинки до переднего края сиденья. Важно,

чтобы все регулировки были независимыми, легко осуществимыми и имели надежную фиксацию. Кресло должно быть регулируемым, с возможностью вращения, чтобы дотянуться до далеко расположенных предметов.

4.1.12. Положение за компьютером

Регулируемое оборудование должно быть таким, чтобы можно было принять следующее положение:

Поставьте ступни плоско на пол или на подножку.

Поясница слегка выгнута, опирается на спинку кресла.

Руки должны удобно располагаться по сторонам.

Линия плеч должна располагаться прямо над линией бедер.

Предплечья можно положить на мягкие подлокотники на такой высоте, чтобы запястья располагались чуть ниже, чем локти.

Локти согнуты и находятся примерно в 3 см от корпуса.

Запястья должны принять нейтральное положение (ни подняты, ни опущены).

Выводы

Любая работа с электроприборами должна производиться в соответствии с нормами техники безопасности. Даже если приборы являются максимально безопасными, привычными в каждодневном использовании и максимально изолированными. Необходимо иметь в пределах досягаемости средства пожаротушения, а так же знать расположение электрического щитка. Так же перед началом необходимо проверять все токоведущие провода и удлинители на предмет повреждений. Только в этом случае может быть гарантировано отсутствие несчастных случаев, связанных с поражением током или пожаром, вызванным неисправными электроприборами.

Компьютерная техника развивается сегодня особенно стремительно, с необычайной быстротой появляются, и также быстро устаревают и отмирают различные технические решения и стандарты. По прогнозам различных экономико-социологических организаций компьютерная техника и телекоммуникации будут оставаться одной из наиболее развивающихся отраслей мировой индустрии еще по крайней мере в течение 10 - 15 лет. Так что уменьшения числа людей, работающих за компьютерами ждать не приходится. Наоборот, повальная компьютеризация, уже давно охватившая бизнес-сектор, сегодня все больше захватывает массового потребителя. В подобной гонке, где нет ничего постоянного, сложно давать рекомендации, принимать какие-либо долговечные решения, а тем паче устанавливать стандарты. А потому, пока компьютерный бум не пойдет на убыль, перед эргономикой и эргономистами будут вставать все новые задачи, касающиеся организации безопасных и комфортных условий для людей работающих с компьютерами .

ЗАКЛЮЧЕНИЕ

На современном рынке телекоммуникационных услуг сети IP-телефонии являются прогрессивной, активно развивающейся и конкурентоспособной технологией. С учетом инфраструктуры телекоммуникационных сетей республики, данная технология является одной из оптимальных для решения проблем междугородних и международных аудио переговоров и видеоконференций, поскольку является более дешевой и менее требовательной к линиям связи технологией, чем альтернативные технологии.

В связи с востребованностью данной технологии на нашем рынке, необходимо осуществить обучение специалистов по данному направлению, способных осуществлять построение и эксплуатацию сетей IP-телефонии.

Данная программа призвана обучить студентов основам построения сетей IP-телефонии. Имея дружественный интерфейс, программа обладает достаточно сложной функциональной начинкой, и имеет огромный потенциал для усовершенствования. В настоящее время основной ее функцией является проверка правильности построения сети на основе анализа ее структуры и составных элементов. В дальнейшем возможно введение дополнительных модулей анализа, дополнительных устройств и начальных данных. Например, можно составить программу для анализа загруженности каналов, учитывая нагрузку, создаваемую каждым терминалом. Возможно, даже составить полноценную программу – анализатор наподобие OPNET, но для этого требуется много времени, труд не одного человека и сотрудничество с компаниями – производителями оборудования IP-телефонии.

Программа предназначена для высших и средне – специальных учебных заведений по направлению телекоммуникации, а так же для специализированных учебных центров. В случае усовершенствования данной программы, ее можно использовать для инженерных расчетов и проектировании реальных сетей на производстве.

Использование при создании данного приложения среды разработки Adobe Flash так же дает программе преимущество перед аналогичными программами. Первый плюс – это размер программы. Если на компьютере установлен Adobe Flash Player, то размер программы не превышает 70 КБ. Так же к плюсам можно отнести возможность взаимодействия с веб-приложениями, и публикации программы непосредственно на веб-странице. При этом даже обладатели подключения к Интернету по технологии Dial – Up на скорости 56 Кбит/с потратят на загрузку приложения не более 15 - 20 с. Еще одним преимуществом является возможность быстро внести изменения и в программную и в графическую часть программы, не используя дополнительных приложений.

В век информационных и телекоммуникационных технологий, необходимо идти в ногу со временем. Когда компьютер есть почти в каждом доме, и почти не осталось людей, у которых не было бы мобильного телефона во всех отраслях необходимо использовать последние достижения научно-технического прогресса. Не является исключением и педагогика. Различные мультимедийные обучающие приложения призваны наглядно продемонстрировать те или иные аспекты профессиональной деятельности, при этом не имея высоких требований к оборудованию. Данные программы могут быть эффективно использованы для обучения в любой области и являются экономически эффективным решением для системы образования в целом.

ЛИТЕРАТУРА

1. Б.С. Гольдштейн, А.В. Пинчук, А.Л. Суховицкий. IP-Телефония. Москва «Радио и связь». 2001
2. Д. Ф. Куроуз, К. В. Росс. Компьютерные сети. Многоуровневая структура Интернета. 2-е издание. «Питер». 2004
3. А.Б. Гольдштейн, Б.С. Гольдштейн. Softswitch. «БХВ – Санкт-Петербург». 2006
4. Cisco IP Telephony Network Design Guide
5. Toga J., Ott J. ITU-T standardization activities for interactive multimedia communications on packet-based networks: H.323 and related recommendations. Computer Networks. 1999.
6. Д. Лотт, Д. Шалл, К. Питере. ActionScript 3.0 Сборник рецептов. Санкт-Петербург – Москва. 2008
7. Adobe® Flash® CS3 Professional Руководство Пользователя. Adobe Inc. 2007
8. Программирование на Adobe® ActionScript® 3.0. Adobe Inc. 2007
9. R. Braunstein, M. H. Wright, J. J. Noble. ActionScript™ 3.0 Bible. Wiley Publishing, Inc. 2008
10. Э. А. Арустамов. Безопасность жизнедеятельности. Москва. 2006
11. www.Flasher.ru
12. www.Wikipedia.org
13. www.AskDoctor.Ru
14. www.mhts.ru

ПРИЛОЖЕНИЕ 1**Листинг программы**

Текст первого кадра основной сцены (основной код, описывающий взаимодействие между всеми элементами).

```
import Sub.Device;

import Sub.Hardware;

import Sub.DrawingCanvas;

import Sub.Variant;

FirstAddBtn.addEventListener(MouseEvent.CLICK, addDevice1);

SecondAddBtn.addEventListener(MouseEvent.CLICK, addDevice2);

ThirdAddBtn.addEventListener(MouseEvent.CLICK, addDevice3);

CableAddBtn.addEventListener(MouseEvent.CLICK, addCable);

TestBtn.addEventListener(MouseEvent.CLICK, test);

stage.addEventListener(KeyboardEvent.KEY_DOWN, onDelPress);

stage.addEventListener(Event.ENTER_FRAME, PlayClip);

stage.addEventListener(Event.ENTER_FRAME, Indicator);

stop();

var IndexH:int = 1;

var IndexD:int = 1;

function addDevice1(e:MouseEvent):void {

    Canvas1.addDev(1, DeviceCh.value, 1, IndexH);

    IndexH +=1;

}

function addDevice2(e:MouseEvent):void {

    Canvas1.addDev(1, DeviceCh.value, 2, IndexH);
```

```
        IndexH +=1;
    }
function addDevice3(e:MouseEvent):void {
    Canvas1.addDev(1, DeviceCh.value, 3, IndexH);
    IndexH +=1;
}
function addCable(e:MouseEvent):void {
    Canvas1.addDev(2, CableCh.value, 0, IndexD);
    IndexD +=1;
}
function onDelPress(event:KeyboardEvent):void {
    if (event.keyCode == 46) {
        this.Canvas1.DeleteDev(Device.selectedDevice);
    }
}
function test(e:MouseEvent):void {
    Canvas1.TestRun();
}
function PlayClip(event:Event):void {
    if (Variant1.flagStart) {
        Canvas1.CodeV = Variant1.Code;
        Canvas1.VP = Variant1.VP;
        stage.removeEventListener(Event.ENTER_FRAME, PlayClip);
        play();
    }
}
function Indicator(event:Event):void {
    if (Canvas1.CreateLink) {
```

```

        LinkIndikator.gotoAndStop(2);
    } else {
        LinkIndikator.gotoAndStop(1);
    }
}

```

Текст 15 кадра (отслеживание правильного решения)

```

this.stop();

stage.addEventListener(Event.ENTER_FRAME, End);

function End(event:Event):void {
    if (Canvas1.flagTest) {
        play();
        stage.removeEventListener(Event.ENTER_FRAME, End);
    }
}

```

Текст модуля рабочего поля DrawingCanvas.as.

```

package Sub{

    import Sub.Device;

    import flash.events.*;

    import flash.events.MouseEvent;

    import flash.display.MovieClip;

    import flash.display.SimpleButton;

    import flash.display.Sprite;

    import flash.display.DisplayObjectContainer;

    import flash.display.DisplayObject;

    public class DrawingCanvas extends Sprite {

        public var flagchange1:Boolean;

        public var flagTestA:Boolean=false;
    }
}

```

```

public var flagTestV:Boolean=false;

public var flagTestF:Boolean=false;

public var flagTest:Boolean=false;

public var CreateLink:Boolean=false;

public var selectedDevice:Device=Device.selectedDevice;

public var currentDevice:Device=Device.currentDevice;

public var CodeV:String;

public var Code:String;

public var
TAq,PCq,TH323q,GVH323q,GKH323q,SERVq,HUBq,ROPTq,RVPq,RE1q:Number;

public var VP:Boolean;

public function DrawingCanvas():void {
    this.addEventListener(MouseEvent.MOUSE_UP,onMouseUp);
}

public function TestRun():void {
    flagTest=false;

    TestActive();

    TestVar();

    TestActive();

    trace("CodeV: ",CodeV);

    trace("Code: ",Code);

    trace("FA: ",flagTestA);

    trace("FV: ",flagTestV);

    trace("FF: ",flagTestF);

    trace("F: ",flagTest);

    if (flagTestA && flagTestV && flagTestF) {
        flagTest=true;
    }
}

```

```

public function TestActive():void {
    var Dev:Device;
    var activebefore:Boolean;
    for (var i1:Number=0; i1 <= this.numChildren - 1; i1++) {
        Dev=this.getChildAt(i1) as Device;
        Dev.Reset();
    }
    do {
        flagchange1=false;
        for (var i2:Number=0; i2 <= this.numChildren - 1; i2++) {
            Dev=this.getChildAt(i2) as Device;
            activebefore=Dev.Active;
            Dev.TestDevice();

            if (activebefore != Dev.Active) {
                flagchange1=true;
            }
        }
    } while (flagchange1);
    TestFil();
    flagTestA=true;
    for (var i3:Number=0; i3 <= this.numChildren - 1; i3++) {
        Dev=this.getChildAt(i3) as Device;
        if (Dev.Active == false) {
            flagTestA=false;
        }
    }
}

```

```

public function TestFil():void {
    var Dev:Device;
    var flagTestF2:Boolean=false;
    var flagTestF3:Boolean=false;
    for (var i3:Number=0; i3 <= this.numChildren - 1; i3++) {
        Dev=this.getChildAt(i3) as Device;
        if (Dev.DType == "Cable") {
            Dev.TestDeviceBranch2(VP);
            Dev.TestDeviceBranch3();
            if (Dev.flagF2) {
                flagTestF2=true;
            }
            if (Dev.flagF3) {
                flagTestF3=true;
            }
        }
    }
    if (flagTestF2 && flagTestF3) {
        flagTestF=true;
    } else {
        flagTestF=false;
    }
}

public function TestVar():void {
    flagTestV=false;

    TAq=0,PCq=0,TH323q=0,GVH323q=0,GKH323q=0,SERVq=0,HUBq=0,ROPTq=0,RVPq=0,RE1q=0;

    var Dev:Device;
    for (var i1:Number=0; i1 <= this.numChildren - 1; i1++) {

```

```
Dev=this.getChildAt(i1) as Device;
if (Dev.Branch == 1) {
    switch (Dev.Type) {
        case "TA" :
            TAq++;
            trace(TAq);
            break;
        case "PC" :
            PCq++;
            break;
        case "TH323" :
            TH323q++;
            break;
        case "GVH323" :
            GVH323q++;
            break;
        case "GKH323" :
            GKH323q++;
            break;
        case "SERV" :
            SERVq++;
            break;
        case "HUB" :
            HUBq++;
            break;
        case "ROPT" :
            ROPTq++;
            break;
```

```

        case "RVP" :
            RVPq++;
            break;
        case "RE1" :
            RE1q++;
            break;
    }
}
}
Code="";
Code+= TAq.toString() + PCq.toString() + TH323q.toString() + GVH323q.toString()
+ GKH323q.toString() + SERVq.toString() + HUBq.toString() + ROPTq.toString() + RVPq.toString() +
RE1q.toString() + "_";

```

```
TAq=0,PCq=0,TH323q=0,GVH323q=0,GKH323q=0,SERVq=0,HUBq=0,ROPTq=0,RVPq=0,RE1q=0;
```

```
for (var i2:Number=0; i2 <= this.numChildren - 1; i2++) {
```

```
    Dev=this.getChildAt(i2) as Device;
```

```
    if (Dev.Branch == 2) {
```

```
        switch (Dev.Type) {
```

```
            case "TA" :
```

```
                TAq++;
```

```
                break;
```

```
            case "PC" :
```

```
                PCq++;
```

```
                break;
```

```
            case "TH323" :
```

```
                TH323q++;
```

```
                break;
```

```
            case "GVH323" :
```

```
                GVH323q++;
```

```

        break;
    case "GKH323" :
        GKH323q++;
        break;
    case "SERV" :
        SERVq++;
        break;
    case "HUB" :
        HUBq++;
        break;
    case "ROPT" :
        ROPTq++;
        break;
    case "RVP" :
        RVPq++;
        break;
    case "RE1" :
        RE1q++;
        break;
    }
}
}

Code+= TAq.toString() + PCq.toString() + TH323q.toString() + GVH323q.toString()
+ GKH323q.toString() + SERVq.toString() + HUBq.toString() + ROPTq.toString() + RVPq.toString() +
RE1q.toString() + "_";

```

```
TAq=0,PCq=0,TH323q=0,GVH323q=0,GKH323q=0,SERVq=0,HUBq=0,ROPTq=0,RVPq=0,RE1q=0;
```

```
for (var i3:Number=0; i3 <= this.numChildren - 1; i3++) {
```

```
    Dev=this.getChildAt(i3) as Device;
```

```
    if (Dev.Branch == 3) {
```

```
switch (Dev.Type) {  
    case "TA" :  
        TAq++;  
        break;  
    case "PC" :  
        PCq++;  
        break;  
    case "TH323" :  
        TH323q++;  
        break;  
    case "GVH323" :  
        GVH323q++;  
        break;  
    case "GKH323" :  
        GKH323q++;  
        break;  
    case "SERV" :  
        SERVq++;  
        break;  
    case "HUB" :  
        HUBq++;  
        break;  
    case "ROPT" :  
        ROPTq++;  
        break;  
    case "RVP" :  
        RVPq++;  
        break;  
}
```

```

        case "RE1" :
            RE1q++;
            break;
        }
    }
}

Code+= TAq.toString() + PCq.toString() + TH323q.toString() + GVH323q.toString()
+ GKH323q.toString() + SERVq.toString() + HUBq.toString() + ROPTq.toString() + RVPq.toString() +
RE1q.toString();

if (Code == CodeV) {
    flagTestV=true;
} else {
    flagTestV=false;
}
}

public function DeleteDev(Dev:Device):void {
    var ContactCable:Device;
    if (Dev.DType == "Hard") {
        for (var i:Number=this.numChildren - 1; i >= 0; i--) {
            ContactCable=this.getChildAt(i) as Device;
            if (ContactCable.DType == "Cable") {
                if (ContactCable.CChild1 == Dev || ContactCable.CChild2
== Dev) {
                    switch (ContactCable.CChild2P) {
                        case "In" :
                            ContactCable.CChild2.portsIn.amountFree++;
                            break;
                        case "Out" :

```

```

ContactCable.CChild2.portsOut.amountFree++;

                                                break;
    }
    switch (ContactCable.CChild1P) {
        case "In" :

ContactCable.CChild1.portsIn.amountFree++;

                                                break;
        case "Out" :

ContactCable.CChild1.portsOut.amountFree++;

                                                break;
    }
    this.removeChild(ContactCable);
}
}
}
this.removeChild(Dev);
}
if (Dev.DType == "Cable") {
    switch (Dev.CChild1P) {
        case "In" :
            Dev.CChild1.portsIn.amountFree++;
            break;
        case "Out" :
            Dev.CChild1.portsOut.amountFree++;
            break;
    }
}

```

```

switch (Dev.CChild2P) {
    case "In" :
        Dev.CChild2.portsIn.amountFree++;
        break;

    case "Out" :
        Dev.CChild2.portsOut.amountFree++;
        break;
}
this.removeChild(Dev);
}

}

public function
addDev(DeviceTypeIndex:int,DeviceSubTypeIndex:String,Branch:int,Index:int):void {

    var newDevice:Device;

    switch (DeviceTypeIndex) {
        case 1 :
            newDevice=new Hardware ;
            newDevice.Branch=Branch;
            newDevice.Index=Index;
            newDevice.Type=DeviceSubTypeIndex;
            newDevice.SetupDevice();
            newDevice.SetupBounds();
            newDevice.x=newDevice.rectBounds.x +
newDevice.rectBounds.width / 2;
            newDevice.y=newDevice.rectBounds.y +
newDevice.rectBounds.height / 2;
            this.addChild(newDevice);

```

```

        break;

    case 2 :

        newDevice=new Cable ;

        newDevice.Type=DeviceSubTypeIndex;

        CreateLink = true;

        this.addEventListener(MouseEvent.CLICK,CreateBegin);

        this.addChild(newDevice);

        Device.currentDevice=newDevice;

stage.addEventListener(KeyboardEvent.KEY_DOWN,CancelCreate);

        this.setChildIndex(newDevice,0);

        break;

    }

}

public function CreateBegin(event:MouseEvent):void {

    if (Device.selectedDevice.portsIn.Type == Device.currentDevice.Type &&
Device.selectedDevice.portsIn.amountFree > 0) {

        CreateLink = true;

        Device.currentDevice.SetupDevice();

        Device.currentDevice.StartCreateIn(Device.selectedDevice);

        this.removeEventListener(MouseEvent.CLICK,CreateBegin);

        this.addEventListener(MouseEvent.CLICK,CreateEnd);

    } else {

        if (Device.selectedDevice.portsOut.Type == Device.currentDevice.Type
&& Device.selectedDevice.portsOut.amountFree > 0) {

            CreateLink = true;

            Device.currentDevice.SetupDevice();

            Device.currentDevice.StartCreateOut(Device.selectedDevice);

```

```

this.removeListener(MouseEvent.CLICK,CreateBegin);

        this.addListener(MouseEvent.CLICK,CreateEnd);

    } else {

        this.removeListener(MouseEvent.CLICK,CreateBegin);

        CreateLink = false;

        this.removeListener(MouseEvent.CLICK,CreateEnd);

        this.removeChild(Device.currentDevice);

    }

}

}

public function CancelCreate(event:KeyboardEvent):void {

    if (event.keyCode == 27) {

        CreateLink = false;

        if (Device.currentDevice.CChild1 != null) {

            if (Device.currentDevice.CChild1P == "In") {

                Device.currentDevice.CChild1.portsIn.amountFree=Device.currentDevice.CChild1.portsIn.amoun
tFree + 1;

            }

            if (Device.currentDevice.CChild1P == "Out") {

                Device.currentDevice.CChild1.portsOut.amountFree=Device.currentDevice.CChild1.portsOut.am
ountFree + 1;

            }

        }

        if (Device.currentDevice.CChild2 != null) {

            if (Device.currentDevice.CChild2P == "In") {

                Device.currentDevice.CChild2.portsIn.amountFree=Device.currentDevice.CChild2.portsIn.amoun
tFree + 1;

```

```

    }

    if (Device.currentDevice.CChild2P == "Out") {

        Device.currentDevice.CChild2.portsOut.amountFree=Device.currentDevice.CChild2.portsOut.am
        ountFree + 1;

    }

}

stage.removeEventListener(KeyboardEvent.KEY_DOWN,CancelCreate);
this.removeEventListener(MouseEvent.CLICK,CreateBegin);
this.removeEventListener(MouseEvent.CLICK,CreateEnd);
this.removeChild(Device.currentDevice);

}

}

public function CreateEnd(event:MouseEvent):void {

    var Device1:Device=Device.selectedDevice;

    CreateLink = false;

    if (Device.selectedDevice.portsIn.Type == Device.currentDevice.Type &&
    Device.selectedDevice.portsIn.amountFree > 0) {

        Device.currentDevice.EndCreateIn(Device1);

        stage.removeEventListener(KeyboardEvent.KEY_DOWN,CancelCreate);

        this.removeEventListener(MouseEvent.CLICK,CreateEnd);

    } else {

        if (Device.selectedDevice.portsOut.Type == Device.currentDevice.Type
        && Device.selectedDevice.portsOut.amountFree > 0) {

            Device.currentDevice.EndCreateOut(Device1);

            stage.removeEventListener(KeyboardEvent.KEY_DOWN,CancelCreate);

            this.removeEventListener(MouseEvent.CLICK,CreateEnd);

        }

    }
}

```

```

        }
    }
    public function onMouseUp(evt:MouseEvent):void {
        if (selectedDevice != null && selectedDevice.Selected) {
            selectedDevice.onMouseUp(evt);
        }
    }
}
}
}

```

Текст модуля Device.as

```

package Sub{
    import Sub.Port;
    import Sub.Hardware;
    import Sub.Cable;
    import flash.events.*;
    import flash.geom.*;
    import flash.text.*;
    import flash.display.MovieClip;
    import flash.display.SimpleButton;
    import flash.display.Sprite;
    import flash.display.DisplayObjectContainer;
    import flash.display.DisplayObject;
    import flash.filters.GlowFilter;
    import flash.filters.BitmapFilterQuality;
    import flash.utils.Timer;
    public class Device extends MovieClip {

```

```
public var Selected:Boolean = false;

public var Index:int = 1;

public var DType:String;

private var selectGlow:GlowFilter = new GlowFilter;

private var filterArray:Array = new Array(selectGlow);

public var Active:Boolean = false;

public var Type:String;

public var portsIn:Port = new Port();

public var portsOut:Port = new Port();

public var Branch:int = 1;

public var rectBounds:Rectangle = new Rectangle();

public var x1:Number,y1:Number,x2:Number,y2:Number;

public var FieldContainer:Sprite = new Sprite();

public var color:uint = 0xFF0000;

public var colorP:uint = 0xFF0000;

public var colorA:uint = 0xFF0000;

public var CChild1:Device;

public var CChild1P:String;

public var CChild2:Device;

public var CChild2P:String;

public var NoLockSelect:Boolean = true;

public var Dragging:Boolean = false;

public var flagF2:Boolean = false;

public var flagF3:Boolean = false;

public var podskazkaText:String;

public static var selectedDevice:Device;

public static var currentDevice:Device;
```

```
public function DeleteDev(Device3:Device):void {
    super.DeleteDev(Device3);
}

public function Device():void {
    setupGlow();
    SetupDevice();
    this.addEventListener(MouseEvent.CLICK, onMouseDown);
    this.addEventListener(MouseEvent.CLICK, onMouseUp);
}

public function TestDevice():void {

}

public function TestDeviceBranch3():void {
}

public function TestDeviceBranch2(VP:Boolean):void {
}

public function StartCreateIn(Device1:Device):void {
}

public function StartCreateOut(Device1:Device):void {
}

public function EndCreateIn(Device1:Device):void {
}

public function EndCreateOut(Device1:Device):void {
}

public function SetupDevice():void {

}
```

```

private function onMouseDown(evt:MouseEvent):void {

    this.select();
    if (DType == "Hard") {
        trace("In:", this.portsIn.amountFree);
        trace("Out:", this.portsOut.amountFree);
        this.startDrag(false, rectBounds);
        Dragging = true;
    }
}

public function onMouseUp(evt:MouseEvent):void {
    this.stopDrag();
    Dragging = false;
}

public function select():void {
    if (NoLockSelect) {
        this.transform.colorTransform = new ColorTransform(1, 1, 1, 1, 0, 0, 80,
0);

        this.filters = filterArray;
        Selected = true;
        if (Device.selectedDevice != this) {
            if (Device.selectedDevice != null) {
                Device.selectedDevice.deselect();
            }
            Device.selectedDevice = this;
        }
    }
}

public function deselect():void {

```

```
this.filters = null;

this.transform.colorTransform = new ColorTransform(1, 1, 1, 1, 0, 0, 0, 0);

Selected = false;

}

public function Reset():void {

}

public function SetupBounds():void {

    switch (Branch) {

        case 1 :

            rectBounds.x = 20;

            rectBounds.y = 75;

            rectBounds.width = 440;

            rectBounds.height = 310;

            break;

        case 2 :

            rectBounds.x = 510;

            rectBounds.y = 75;

            rectBounds.width = 440;

            rectBounds.height = 310;

            break;

        case 3 :

            rectBounds.x = 265;

            rectBounds.y = 412;

            rectBounds.width = 440;

            rectBounds.height = 310;
```

```

                break;
            }
        }
        private function setupGlow():void {
            selectGlow.color = 0x3C9DFF;
            selectGlow.quality = BitmapFilterQuality.HIGH;
            selectGlow.blurX=5;
            selectGlow.blurY=5;
        }
    }
}

```

Текст модуля Hardware.as

```

package Sub{
    import Sub.Port;
    import Sub.Device;
    import flash.events.*;
    import flash.geom.*;
    import flash.display.MovieClip;
    import flash.display.SimpleButton;
    import flash.display.Sprite;
    import flash.display.DisplayObjectContainer;
    import flash.display.DisplayObject;
    import flash.filters.GlowFilter;
    import flash.filters.BitmapFilterQuality;
    public class Hardware extends Device {
        private var PicIndex:int = 1;
    }
}

```

```
public function Hardware():void {  
    SetupDevice();  
}  
  
public override function Reset():void {  
    this.Active = false;  
    deactivate();  
    portsIn.Active = false;  
    portsOut.Active = false;  
}  
  
public override function TestDevice():void {  
    switch (Type) {  
        case "TA" :  
            if (portsIn.Active) {  
                this.Active = true;  
                activate();  
            } else {  
                this.Active = false;  
                deactivate();  
            }  
            break;  
        case "PC" :  
            if (portsIn.Active) {  
                this.Active = true;  
                activate();  
            } else {  
                this.Active = false;  
                deactivate();  
            }  
        }  
    }
```

```
    }  
    break;  
case "TH323" :  
    if (portsIn.Active) {  
        this.Active = true;  
        activate();  
    } else {  
        this.Active = false;  
        deactivate();  
    }  
    break;  
case "GVH323" :  
    if (portsIn.Active) {  
        portsOut.Active = true;  
        this.Active = true;  
        activate();  
    } else {  
        portsOut.Active = false;  
        this.Active = false;  
        deactivate();  
    }  
    break;  
case "GKH323" :  
    if (portsIn.Active) {  
        this.Active = true;  
        activate();  
    } else {  
        this.Active = false;
```

```
        deactivate();
    }
    break;
case "SERV" :
    if (portsIn.Active) {
        this.Active = true;
        activate();
    } else {
        this.Active = false;
        deactivate();
    }
    break;
case "HUB" :
    portsOut.Active = true;
    this.Active = true;
    activate();
    break;
case "ROPT" :
    if (portsIn.Active) {
        portsOut.Active = true;
        this.Active = true;
        activate();
    } else {
        portsOut.Active = false;
        this.Active = false;
        deactivate();
    }
    break;
```

```
case "RVP" :  
    if (portsIn.Active) {  
        portsOut.Active = true;  
        this.Active = true;  
        activate();  
    } else {  
        portsOut.Active = false;  
        this.Active = false;  
        deactivate();  
    }  
    break;  
case "RE1" :  
    if (portsIn.Active) {  
        portsOut.Active = true;  
        this.Active = true;  
        activate();  
    } else {  
        portsOut.Active = false;  
        this.Active = false;  
        deactivate();  
    }  
}  
}  
  
public override function SetupDevice():void {  
    switch (Type) {  
        case "TA" :  
            podskazkaText = "Телефонный аппарат";  
            portsIn.amount = 1;
```

```
portsIn.amountFree = 1;
portsOut.amount = 0;
portsOut.amountFree = 0;
portsIn.Type = "RJ11";
portsOut.Type = "NA";
PicIndex = 1;
break;
case "PC" :
    podskazkaText = "Персональный компьютер";
    portsIn.amount = 1;
    portsIn.amountFree = 1;
    portsOut.amount = 0;
    portsOut.amountFree = 0;
    portsIn.Type = "RJ45";
    portsOut.Type = "NA";
    PicIndex = 2;
    break;
case "ТН323" :
    podskazkaText = "Видеотелефон";
    portsIn.amount = 1;
    portsIn.amountFree = 1;
    portsOut.amount = 0;
    portsOut.amountFree = 0;
    portsIn.Type = "RJ45";
    portsOut.Type = "NA";
    PicIndex = 3;
    break;
case "GVH323" :
```

```
podskazkaText = "Шлюз ТФОП/IP";  
portsIn.amount = 1;  
portsIn.amountFree = 1;  
portsOut.amount = 16;  
portsOut.amountFree = 16;  
portsIn.Type = "RJ45";  
portsOut.Type = "RJ11";  
PicIndex = 4;  
break;  
case "GKH323" :  
    podskazkaText = "Привратник";  
    portsIn.amount = 1;  
    portsIn.amountFree = 1;  
    portsOut.amount = 0;  
    portsOut.amountFree = 0;  
    portsIn.Type = "RJ45";  
    portsOut.Type = "NA";  
    PicIndex = 5;  
    break;  
case "SERV" :  
    portsIn.amount = 1;  
    portsIn.amountFree = 1;  
    portsOut.amount = 0;  
    portsOut.amountFree = 0;  
    portsIn.Type = "RJ45";  
    portsOut.Type = "NA";  
    PicIndex = 6;  
    break;
```

```
case "HUB" :  
  
    podskazkaText = "HUB";  
  
    portsIn.amount = 0;  
  
    portsIn.amountFree = 0;  
  
    portsOut.amount = 32;  
  
    portsOut.amountFree = 32;  
  
    portsIn.Type = "NA";  
  
    portsOut.Type = "RJ45";  
  
    PicIndex = 7;  
  
    break;  
  
case "РОПТ" :  
  
    podskazkaText = "Оптический маршрутизатор";  
  
    portsIn.amount = 1;  
  
    portsIn.amountFree = 1;  
  
    portsOut.amount = 1;  
  
    portsOut.amountFree = 1;  
  
    portsIn.Type = "RJ45";  
  
    portsOut.Type = "ОПТ";  
  
    PicIndex = 8;  
  
    break;  
  
case "RVP" :  
  
    podskazkaText = "Ethernet маршрутизатор";  
  
    portsIn.amount = 1;  
  
    portsIn.amountFree = 1;  
  
    portsOut.amount = 1;  
  
    portsOut.amountFree = 1;  
  
    portsIn.Type = "RJ45";  
  
    portsOut.Type = "RJ45";
```

```
        PicIndex = 9;

        break;

    case "RE1" :

        podskazkaText = "E1 маршрутизатор";

        portsIn.amount = 1;

        portsIn.amountFree = 1;

        portsOut.amount = 1;

        portsOut.amountFree = 1;

        portsIn.Type = "RJ45";

        portsOut.Type = "E1";

        PicIndex = 10;

        break;

    }

    gotoAndStop(PicIndex);

    SetupBounds();

    DType = "Hard";

}

private function activate():void {

    if ((PicIndex >= 1)&&(PicIndex < 11)) {

        PicIndex += 10;

    }

    gotoAndStop(PicIndex);

}

private function deactivate():void {

    if ((PicIndex >= 10)&&(PicIndex < 21)) {

        PicIndex -= 10;

    }

    gotoAndStop(PicIndex);

}
```

```

    }
}
}

```

Текст модуля Cable.as

```

package Sub{

    import Sub.Port;

    import Sub.Device;

    import flash.events.*;

    import flash.geom.*;

    import flash.display.MovieClip;

    import flash.display.SimpleButton;

    import flash.display.Sprite;

    import flash.display.DisplayObjectContainer;

    import flash.display.DisplayObject;

    import flash.filters.GlowFilter;

    import flash.filters.BitmapFilterQuality;

    public class Cable extends Device {

        private var flagDragging:Boolean=false;

        private var drawn:Boolean=false;

        public function Cable():void {

            SetupDevice();

        }

        public override function Reset():void {

            this.Active=false;

            color=colorP;

```

```

        clearLine();

        drawLine();
    }

    public override function TestDeviceBranch3():void {

        if (CChild1.Branch == 1 && CChild2.Branch == 3 || CChild2.Branch == 1 &&
CChild1.Branch == 3) {

            if (CChild1P == "Out" && CChild1.Type == "ROPT" && CChild1.Active &&
CChild2P == "Out" && CChild2.Type == "ROPT" && CChild2.Active) {

                this.Active=true;

                color=colorA;

                clearLine();

                drawLine();

                flagF3=true;

            } else {

                flagF3=false;

            }

        } else {

            flagF3=false;

        }

    }

    public override function TestDeviceBranch2(VP:Boolean):void {

        if (CChild1.Branch == 1 && CChild2.Branch == 2 || CChild2.Branch == 1 &&
CChild1.Branch == 2) {

            if (VP && CChild1.Type == "RVP" && CChild1.Active && CChild2.Type ==
"RVP" && CChild2.Active) {

                this.Active=true;

                color=colorA;

                clearLine();

                drawLine();

                flagF2=true;

```

```

    } else {
        if (VP == false && CChild1P == "Out" && CChild1.Type == "RE1"
&& CChild1.Active && CChild2P == "Out" && CChild2.Type == "RE1" && CChild2.Active) {
            this.Active=true;
            color=colorA;
            clearLine();
            drawLine();
            flagF2=true;
        } else {
            flagF2=false;
        }
    }
} else {
    flagF2=false;
}
}

public override function TestDevice():void {

    if (CChild1.Branch == CChild2.Branch) {
        if (CChild1P == "Out" && CChild1.portsOut.Active) {
            this.Active=true;
            color=colorA;
            clearLine();
            drawLine();
            if (CChild2P == "In") {
                CChild2.portsIn.Active=true;
            }
            if (CChild2P == "Out") {

```

```
        CChild2.portsOut.Active=true;
    }
}
if (CChild2P == "Out" && CChild2.portsOut.Active) {
    this.Active=true;
    color=colorA;
    clearLine();
    drawLine();
    if (CChild1P == "In") {
        CChild1.portsIn.Active=true;
    }
    if (CChild1P == "Out") {
        CChild1.portsOut.Active=true;
    }
}
}
}

public override function SetupDevice():void {
    switch (Type) {
        case "RJ11" :
            podskazkaText="Телефонный кабель";
            colorP=0x800000;
            colorA=0xFF0000;
            break;
        case "RJ45" :
            podskazkaText="Витая пара";
            colorP=0x808000;
            colorA=0xFFFF00;
```

```
        break;

    case "OPT" :

        podskazkaText="Оптоволокно";

        colorP=0x008000;

        colorA=0x00FF00;

        break;

    case "E1" :

        podskazkaText="Поток E1";

        colorP=0x008080;

        colorA=0x00FFFF;

        break;

    }

    color=colorP;

    addChild(FieldContainer);

    DType="Cable";

}

public override function StartCreateIn(Device1:Device):void {

    CChild1P="In";

    CChild1=Device1;

    CChild1.portsIn.amountFree-= 1;

    NoLockSelect=false;

    x1=Device1.x;

    y1=Device1.y;

    addEventListener(Event.ENTER_FRAME,BeginDraw);

}

public override function StartCreateOut(Device1:Device):void {

    CChild1P="Out";

    CChild1=Device1;
```

```
CChild1.portsOut.amountFree-= 1;

NoLockSelect=false;

x1=Device1.x;

y1=Device1.y;

addEventListener(Event.ENTER_FRAME,BeginDraw);
}

public override function EndCreateIn(Device1:Device):void {

    CChild2=Device1;

    CChild2P="In";

    x2=Device1.x;

    y2=Device1.y;

    removeEventListener(Event.ENTER_FRAME,BeginDraw);

    CChild2.portsIn.amountFree-= 1;

    NoLockSelect=true;

    addEventListener(Event.ENTER_FRAME,Watch);
}

public override function EndCreateOut(Device1:Device):void {

    CChild2=Device1;

    CChild2P="Out";

    x2=Device1.x;

    y2=Device1.y;

    removeEventListener(Event.ENTER_FRAME,BeginDraw);

    CChild2.portsOut.amountFree-= 1;

    NoLockSelect=true;

    addEventListener(Event.ENTER_FRAME,Watch);
}

public function BeginDraw(event:Event) {

    clearLine();
```

```

        x2=mouseX;

        y2=mouseY;

        drawLine();

    }

    public function EndDraw(event:Event) {

        clearLine();

        x2=CChild2.x;

        y2=CChild2.y;

        drawLine();

    }

    public function drawLine():void {

        var graph:Sprite=new Sprite ;

        var xsize:Number=x2 - x1;

        var ysize:Number=y2 - y1;

        var a:Number=xsize * 0.1;

        var alfa=Math.atan(ysize / xsize * 0.8);

        var b:Number=a * Math.cos(alfa);

        var c:Number=a * Math.sin(alfa);

        graph.graphics.lineStyle(2,color);

        graph.graphics.moveTo(x1,y1);

        graph.graphics.curveTo(x1 + a,y1,x1 + a + b,y1 + c);

        graph.graphics.lineTo(x2 - a - b,y2 - c);

        graph.graphics.curveTo(x2 - a,y2,x2,y2);

        FieldContainer.addChild(graph);

        drawn=true;

    }

    public function Watch(event:Event):void {

        if (CChild1.Dragging) {

```

```
        x1=CChild1.x;
        y1=CChild1.y;
        clearLine();
        drawLine();
    }
    if (CChild2.Dragging) {
        x2=CChild2.x;
        y2=CChild2.y;
        clearLine();
        drawLine();
    }
}

public function clearLine():void {
    if (drawed) {
        FieldContainer.removeChildAt(0);
        drawed=false;
    }
}
}
```

Текст модуля Variant.as

```
package Sub{

    import flash.display.MovieClip;

    import flash.display.SimpleButton;

    import flash.events.MouseEvent;

    import flash.display.Sprite;
```

```

import flash.text.*;

public class Variant extends Sprite {

    public var flagStart:Boolean = false;

    public var Var:Number;

    public var Code:String;

    public var VP:Boolean;

    public function Variant():void {

        SetupVar();

        OK.addEventListener(MouseEvent.CLICK, startProg);

    }

    public function SetupVar():void {

        VarInput.maxChars = 2;

        VarInput.restrict = "0-9";

    }

    public function startProg(event:MouseEvent):void {

        if (VarInput.text != "") {

            if (VarInput.length == 1) {

                Var = int(VarInput.text);

            } else {

                Var =int(VarInput.text.charAt(1));

            }

            flagStart = true;

            VarChouse();

        }

    }

    public function VarChouse():void{

        switch(Var){

            case 1:

```

```
Code = "8841111110_4521001010_3501001100";  
VP = true;  
break;  
case 2:  
Code = "6921111110_3601001010_4211001100";  
VP = true;  
break;  
case 3:  
Code = "8401101101_5301001001_2501001100";  
VP = false;  
break;  
case 4:  
Code = "5731111101_6411001001_3401001100";  
VP = false;  
break;  
case 5:  
Code = "4801101110_3301001010_5301001100";  
VP = true;  
break;  
case 6:  
Code = "9521111110_5401001010_4311001100";  
VP = true;  
break;  
case 7:  
Code = "6641111101_4621001001_3601001100";  
VP = false;  
break;  
case 8:
```

```
        Code = "8601101101_6201001001_5401001100";  
        VP = false;  
        break;  
    case 9:  
        Code = "5921111110_2501001010_3311001100";  
        VP = true;  
        break;  
    case 0:  
        Code = "6401101110_4301001010_4401001100";  
        VP = true;  
        break;  
    }  
    }  
    }  
}
```

Текст модуля Port.as

```
package Sub{  
    public class Port extends Object{  
        public var Active:Boolean = false;  
        public var amount:int = 1;  
        public var amountFree:int = amount;  
        public var Type:String = "RJ45";  
    }  
}
```

ПРИЛОЖЕНИЕ 2

Слайды презентации