1. **Постановка задачи.** Руководством предприятия Элтекс поставлена задача - "Разработать модуль интерактивного голосового меню (IVR) для транковых шлюзов SMG1016M, SMG2016". SMG - signaling media gateway, шлюз сигнализации и медии, до недавнего времени устройства позиционировались как транковые шлюзы операторского класса (ставятся на границе сетей IP/TDM). Встала необходимость расширения функционала для удовлетворения требованиям IP-ATC. Одним из пунктов требований является наличие интерактивного голосового меню, реализация которого была поручена мне.
2. **Требования.** \*Количество поддерживаемых протоколов сигнализации должно соответствовать протоколам, поддерживаемым на устройстве - поддерживаются протоколы IP (SIP, H.323)/TDM (SS7, PRI).

\*Модульная архитектура устойчивая к отказам

\*Производительность не ниже производительности основной системы обработки вызовов: CPS : ip - 14; e1 - 30.

\*Поддержка функциональных блоков: «Info», «Play», «IVR», «Dial», «Rec», «Caller Info». «Info» - для возможности проигрывания в предответном состоянии (когда плата не взимается). «Play» - для проигрывания в ответном состоянии. «IVR» - для взаимодействия с абонентом (сбор цифр, проигрывание, донабор). «Dial» - для возможности совершения исходящих. «Rec» - для записи разговоров. «Caller Info» - для подмены имени вызывающего абонента.

1. **IVR .** Интерактивное Голосовое Меню  (англ. *Interactive Voice Response*) — это система предварительно записанных голосовых сообщений, выполняющая функцию маршрутизации звонков, пользуясь информацией, вводимой клиентом с помощью тонального набора.
2. **Сценарии обработки вызовов.** Сценарии обработки создаются с помощью веб – интерфейса (показываешь на картинку кароч) администратором устройства и представляют собой отдельные файлы в формате JSON с описанием блоков их связей и параметров.
3. **Управление сценариями .** IVR процессор – модуль управления сценариями, в системе представлен в виде отдельного процесса. На основе блоков обрабатываемого сценария IVR процессор формирует управляющие команды с помощью которых модуль IVR осуществляет управления вызовами.

Система IVR состоит из двух основных модулей:

IVR модуль: модуль для управления и взаимодействия с абонентами

IVR процессор: модуль управления сценариями, задача которого сводится к передаче управляющих команд в IVR модуль на основе которых и происходит управление вызовами.

Моя задача - разработать IVR модуль.

1. **Общая система обработки вызова .** Немного о общей системе обработки вызова устройств SMG. Система обработки вызова представляет собой систему из 4 транспортных модулей протокольного уровня, каждый из которых осуществляет работу с определенным протоколом сигнализации, и ядра обработки вызовов (PBX) через которое взаимодействуют разные транспортные модули.
2. **Работа с вызовами .** Модуль IVR, условно, можно изобразить за ядром обработки вызова, т.к. это конечная стадия обработки. Для каждого поступившего вызова в ядре создается объект XPORT. XPORT – унифицированное представление вызова во внутренней схеме обработки вызова. Так же, для привязки вызова к конкретной системе обработки, создается объект SRVPORT (сервисный порт), через него происходит взаимодействие с ядром. Так же, эти объекты позволяют эмулировать абонентов (используется при исходящих вызовах).
3. **Структура модуля IVR .** IVR модуль, условно, можно разделить на 7 частей, которые осуществляют работу с абонентом в зависимости от его состояния, либо работу с внутренними ресурсами.

осуществляют работу со своим набором данных,

1. **Состояния вызовов .** Каждый из вызовов в IVR имеет 5 состояний, свободен, предответное, ответное, разговаривает и вызывается. В зависимости от состояния вызова становятся активными различные возможности обработки вызова.
2. **Управление плейлистами .** Переходя к функциональной части. Одной из основных задач голосового меню является проигрывание голосовых сообщений. Проигрывание осуществляется на модулях MSP - Media stream processor - специализированые процессоры для обработки RTP траффика, которые управляют голосовыми каналами. Непосредственно они генерирует RTP пакеты в сторону абонента. Проигрываемые элементы делятся на два типа – звуковой файлы формата «.wav» или тон определенной частоты. Настройка плейлиста производится администратором устройства вечер веб-интерфейс. IVR процессор, после разбора сценария, отправляет команду со списком проигрываемых элементов в IVR модуль. Далее происходит поиск абонента для которого эта команда предназначена и проверка его состояния, у абонента формируется плейлист и запускается процедура проигрывания. Процедура проигрывания для файлов и тонов имеет отличия, процедура проигрывания файлов выглядит следующим образом - из плейлиста берется первый элемент - имя файла, производится поиск файла на носителе, из файла формируется буффер на проигрывание, вырезаются wav заголовки, если таковые имеются, формируется команда на проигрывание буффера и отправляется в модули MSP. После проигрывания, MSP сигнализирует уведомление об окончании проигрывания, которое перехватывается модулем IVR и происходит переключение на следующий элемент в списке. Если же файл не был найден, сразу происходит переход на следующий элемент. Для проигрывания тонов, в команде на проигрывание от IVR процессора, присутствует параметр duration, параметр определяет длительность проигрывания тона, т.к. некоторые из тонов могут генерироваться бесконечно. На основе этого параметра взводится таймер, формируется команда на генерацию тона определенной частоты в MSP, по истечению таймера формируется команда в MSP на остановку проигрывания и происходит переход на следующий элемент. По окончанию проигрывания всего плейлиста модуль IVR отправляет уведомление в IVR процессор об окончании проигрывания и статусами проигрываемых файлов (успешно, неуспешно).
3. **Запись разговоров.** Запись разговоров начинается по команде от IVR процессора. На основе параметров команды подготавливается файл для записи, в который предварительно записываются wav заголовки. На схеме, для удобства, изображено два вызова, т.е. ситуация, когда абонент А звонит абоненту Б через IVR сценарий. Создается еще один, внутренний, сервисный порт и на основе двух абонентских, сервисных портов, с помощью модулей MSP, создается конференция из которой звуковой канал со смикшированный звуком закрепляется за внутренним сервисным портом и производится запись в файл. Работа с потоком происходит следующим образом. Процессор MSP аккумулирует несколько RTP пакетов (что-то вроде jitter-буффера) и отправляет команду ядро обработки вызова с аккумулированными голосовыми данными, которую перехватывает IVR процессор и производит запись в файл.
4. **Сбор DTMF.** Dual-Tone Multi-Frequency тональный сигнал. Сбор тональных сигналов необходим для перехода по ветвям меню IVR, а так же для донабора внутреннего номера. Начало сбора происходит по приходу команды от IVR процессора. Происходит поиск нужного вызова и активируется сбор. Так же, на основе параметров команды, активируются таймера сбора - время общего сбора и межцифровой интервал. Тональный сигнал может прийти двумя путями - по протоколу сигнализации SIP (info) и закодированным в голосовом потоке. В первом случае тональный сигнал "ловит" транспортный модуль обработки SIP, а во втором медиа процессор при декодировании RTP пакета. В обоих случаях тональный сигнал,через XPORT, перехватывается IVR модулем. После получения сигнала производится проверка, включен ли сбор у вызова, проверяются условия сбора (кол-во необходимых цифр). Окончание сбора происходит либо по истечению одного из таймеров, либо по выполнению условий сбора (т.е. собрали достаточное кол-во цифр).
5. **Управление таймерами.** В модуле имеется один главный таймер - нить pthread в теле которой производится проверка счетчиков (таймеров) проигрывания, сбора DTMF, ошибок сервера. Для каждого из вызовов, на начало проигрывания бесконечного тона либо сбора DTMF, запоминается время, когда началось проигрывание либо запись. Таймер совершает обход всех вызовов и вычисляет время, в течении которого проигрывание, запись выполняется, если превышен предел допустимого времени, происходит остановка проигрывания, либо остановка сбора DTMF. Контроль ошибок сервера происходит немного иначе, каждые 100 мс сервер опрашивается на предмет ошибок. Ошибки бывают двух типов - критичные и не критичные. Не критичные - ошибки отдельных вызовов не влияющие на общую работу системы, такие ошибки обрабатываются отдельно, но в случае, когда ошибку обработать невозможно запись о ней сохраняется в списке ошибок. Когда превышен допустимый лимит не критичных ошибок (в данный момент > 20), происходит перезапуск сервера. При критичных ошибках перезапуск сервера происходит незамедлительно. Каждый перезапуск происходит с последующим восстановлением всех вызовов.
6. **Работа с сервером.** Работа с сервером осуществляется через входящую и исходящую очередь. В обмене между сервером имеется 4 типа сообщений:

Seize – сообщение входящего/исходящего занятия.

Progress – сообщение для управления вызова с предответном и ответном состояниях.

Answer – сообщение для управления вызовами в ответном состоянии.

Release – сообщение для уведомления о завершении вызова.

1. **Управление подключениями.** Уровень сервера обособлен от вызовов, сервер управляет сессиями. Сессии – логические подключения, в рамках главного подключения, соответствующие вызовам. Главное подключение: подключение по сокету + "аутентификация" IVR процессора (обмен протокольными сообщениями - запрос на подключение). Логические подключения: подключения в рамках главного подключения - отдельные сообщеня с идентификатором получателя (сессии).
2. **SMARTTI – сервер.** Как уже говорилось, работа с сервером производится через входящую и исходящую очереди. Для связи с вызовами IVR реализовано отображение вызовов (связь "Вызов" - подключение"). Имеется список подключений, таймер подключений - который при длительной неактивности в сессии начинает отправлять сообщения keep-alive. Уровень обработки протокольной логики - обработка запросов на новое подключение, валидация сообщений, ответ на встречные сообщения keep-alive, обработка закрытия сессий. Основная часть сервера - контроллер потоков сообщений. Ввиду особенности архитектуры, а так же из–за ограничений накладываемой самой библиотекой ZMQ работа с одним сокетом в разных потоках не представляется возможной. В связи с этим два потока (чтения и записи) были объединены в один в один. Для уменьшения времени задержек при отправке/приеме сообщений был реализован контроллер сообщений. Контроллер выполняет функцию своеобразного переключателя, переключая сокет с приема на отправку и обратно. Контроллер имеет две настройки – время переключения и максимальное количество сообщений для обработки. Для кодирования/ декодирование данных используется библиотека ASN.1.
3. **Обработка сообщений от сервера.** После получения сообщения сервер кладет сообщение во внутреннюю очередь обработки модуля IVR и отправляет главную очередь обработки ядра уведомление о том, что в модуле IVR есть данные, которые нужно обработать. После того, как ядро передаст управление модулю начнется обработка внутренних сообщений. В модуле IVR, при обработке элементов из входящей очереди команд, контролируется количество обработанных элементов. Такой подход необходим для того, чтобы обработка сообщений одного модуля не занимала главную очередь обработки ядра.