

**Asterisk + WebRTC =
любовь и обожание
наконец**

Обо мне:



- Почти 20 лет опыта с колл-центрами и телефонией
- Начинал с Nortel далее Avaya сейчас в основном Asterisk

Содержание

- Плюсы WebRTC
- ICE – краеугольный камень
- Инструменты отладки сигнализации и голоса
- Мониторинг
- Резервирование
- Безопасность

Типичная конфигурация

- Небольшой контакт-центр
 - Аутсорсинг
 - Продажи
- Агенты работают удаленно

Плюсы WebRTC

- Нулевые затраты на рабочее место
- Внедрение телефонии в CRM – простая задача для веб-разработчиков
- Нет коннектора к CRM. Не нужно мучаться с AMI или ARI
- Телефония в CRM не зависит от типа АТС
- Расширения для браузера с телефоном и простейшим API

Технические требования

- Домен и сертификат (letsencrypt работает)
- Актуальный Asterisk (у клиентов 20 и 21)

Залезаем на плечи вебтехнологий

- Чат-рулетка как двигатель прогресса
- ICE – краеугольный камень для соединения между собой пользователей
- Набор кустылей технологий для общения браузеров между собой по UDP
- Нужно ли быть Asterisk похожим на браузер ?
- В WebRTC соединение устанавливается только на основе ICE
 - Настройка ICE – важнейшая часть!

Реальный кейс

- Asterisk за NAT
- Несколько внутренних адресов (е.g. внутренний IP, managed IP keepalived)
- По умолчанию ICE пошлет эти адреса в сторону браузера
- Браузер МОЖЕТ слать на них медиапоток если так решит ICE

Скрываем внутренние адреса

rtp.conf

```
[general]
```

```
rtpstart=10000  
rtpend=20000
```

```
rtpchecksums=yes  
strict RTP=yes
```

```
icesupport=yes  
stunaddr=stun.l.google.com:19302
```

```
[ice_host_candidates]
```

```
10.128.0.6 => .232.230.146  
10.128.0.2 => .232.230.146  
10.128.0.3 => .232.230.146
```

Включение ICE

Ноды Asterisk

SDP от ATC за NAT

Только один кандидат

```
v=0
o=- 5132437062062541021 4 IN IP4 10.128.0.3
s=Asterisk
c=IN IP4 10.128.0.3
t=0 0
a=msid-semantic:WMS *
a=group:BUNDLE 0
m=audio 10250 UDP/TLS/RTP/SAVPF 8 0 9
a=connection:new
a=setup:active
a=fingerprint:SHA-256 8A:35:D0:23:36:3D:86:C7:33:12:28:25:AD:C9:2E:7C:2E:5A
a=ice-ufrag:2216d0161ef4553423d6d27f604886f6
a=ice-pwd:2c038a763ee8632b5045566553b59c87
a=candidate:H23e8e692 1 UDP 2130706431 .232.230.146 10250 typ host
a=rtpmap:8 PCMA/8000
```

ВАЖНО: ATC шлет только свой внешний адрес с типом host

Проверка конфигурации клиента

Должен быть внешний IP

- Внешний адрес - тип srflx

```
v=0
o=- 8006288905601817089 2 IN IP4 127.0.0.1
s=-
t=0 0
a=group:BUNDLE 0
a=extmap-allow-mixed
a=msid-semantic: WMS 739445eb-c247-4360-a025-7f6162f24734
m=audio 52178 UDP/TLS/RTP/SAVPF 111 63 103 104 9 0 8 106 105 13 110 111 113 126
c=IN IP4 109.167.132.72
a=rtcp:9 IN IP4 0.0.0.0
a=candidate:4062576099 1 udp 2122260223 169.254.211.249 52177 typ host generation 0 network-id 2
a=candidate:2294146922 1 udp 2122194687 10.111.111.86 52178 typ host generation 0 network-id 1 network-cost 10
a=candidate:2310903116 1 udp 1685987071 109.167.132.72 52178 typ srflx raddr 10.111.111.86 rport 52178 generation 0 network-id 1 network-cost 10
a=candidate:2364153723 1 tcp 1518280447 169.254.211.249 9 typ host tcptype active generation 0 network-id 2
a=candidate:4134667762 1 tcp 1518214911 10.111.111.86 9 typ host tcptype active generation 0 network-id 1 network-cost 10
a=ice-ufrag:2ERY
```

У сотен агентов ни разу не было проблем с доступностью STUN и приемом RTP потока через NAT

Как смотреть сигнализацию SIP ?

- `pjsip set logger` – так себе по уровню удобства
- `sngrep` напрямую не работает из-за обязательного шифрования
 - Настраиваем NEP модуль `asterisk` на трансляцию
 - `sngrep -E` работает!
 - HOMER и аналогичные продукты

Отладка передачи речи в WebRTC

- Обычный RTP поток , только payload зашифрован
- `rtp set debug` - работает
- Можно использовать `wireshark` для анализа характеристик RTP потока
- `pjsip show channelstats` - показывает RTCP статистику
- MixMonitor и ChanSpy – работают разумеется
- `chrome://webrtc-internals/` - настоящий кладёзь информации
 - Работу ICE полностью можно увидеть
 - Real-time графики качества передачи голоса

Контроль качества агентских подключений

- У операторов на рабочих местах интернет может быть любого качества
- Ручной эхо-тест перед началом смены
- Записываем RTCP статистику агентского канала в CDR

```
;Канал к рабочему месту
exten => set_handlerA,1,Set(CHANNEL(hangup_handler_push)=xxxx-dialout,outbound_handlerA,1)
exten => set_handlerA,n,Return()

exten => outbound_handlerA,1,NoOp()
same => n,Set(CDR(rtpqa)=${CHANNEL(rtcp,all)})
same => n,Return()
```

CRM видит, у оператора проблемы. Не назначает ему важных лидов как вариант.

Мониторинг без затрат

- С 18 версии Asterisk умеет отдавать по HTTP метрики для Prometheus.
- Состояния эндпоинтов, каналы, продолжительность вызовов и прочее
- Гигантский шаг вперед по сравнению с SNMP!
- Отправляем метрики в Google Cloud Monitoring и манипулируем PromQL
- Одно удовольствие делать дашборды

График количества операторов



Безопасность

- SSL соединение без имени хоста невозможно
- fail2ban по ошибкам SSL подключения
- CRM генерирует и передает токен в заголовке вызова
- АТС проверяет его перед вызовом в ТФОП

Горячее резервирование без затрат

- Конфигурация Asterisk в Realtime.
- БД с резервированием/бэкапами и тп
- Две ноды Asterisk на keepalived
 - keepalived переключает ноды Active - Standby
 - Ручное переключение нод

В итоге

- **SIP по WebRTC -зрелая технология:** Надежность соединения и качество голоса не хуже чем у софтбокса.
- **У технологии много преимуществ:** Только браузер. Легко внедрить.
- **Инструментов отладки:** Богатые возможности отладки сигнализации и голоса.
- **Безопасность** не хуже традиционных решений
- **Программистам нравится работать с JsSIP** больше чем с AMI
- **Реальные проблемы** случались у хостинга, операторов ТФОП, на рабочих местах, настройках БД, сертификатах, но НИКОГДА собственно в WebRTC.
- **Метрики Prometheus** большой шаг вперед

Спасибо за внимание!

Буду рад ответить на все ваши вопросы
сейчас или свяжитесь со мной в будущем:

Александр Серебряный

Теллур-Сетевые технологии

krooto@gmail.com

+7-921-747-64-51

@krooto

