

TZ5002

TЗ на разработку микросервиса Язык разработки go, желательно при необходимости использовании использовать такие пакеты как gogin,lumberjack,gorm. Код должен собирать и работать как минимум в windows и linux ОС.

Описание

Сервис предоставления потока с помощью RTSP overt TCP из архива. Конфигурационный файл Файл должен лежать по умолчанию в директории рядом с исполняемым файлом с именем config-5002.json и имеет следующую структуру. Если файла конфигурационного нет, то файл должен быть создан со значениями по умолчанию указанными ниже.

```
{
  "webservice": {
    "host": "localhost",
    "port": 5002
  },
  "log": {
    "path": "/var/log/5001/",
    "loglevel": 0,
    "rotate": 10
  },
  "rtspserver": {
    "host": "localhost",
    "port": 3297,
    "auth": "base64code=" //login and password
  }
}
```

WebAPI

Должны быть реализованы следующие схемы

/sessions GET

Результат выполнения должен быть вывод массива в формате json текущих всех сессий, у каждой сессии должен быть индикатор. Для примера

```
[
  {
    "ip": "1.2.3.4", //внешний адрес клиента
    "startsession": 11111111, #timestamp
    "idsession": 123124123, #уникальный номер сессии, храним в
памяти
    "cameraid": 123
  },
  ...
]
```

/stop POST

Передавая в json формате только номер сессии, данная сессия должна быть терминирована со стороны сервера, как внутренними механизмами RTSP так и механизмом TCP.

```
{
  "idsession": 123124123
}
```

/stopall POST

Не какие данные не передаются, по аналогии с методом API /stop должны все сессии быть терминированы. Лучше всего применить вариант, обращение к методу /sessions и в цикле передать каждый индикатор сессии в метод /stop

Формат URL

Формат **rtsp://IP:3297/cam00001515** После cam необходимо определить id камеры путём преобразования строки 00001515 в число 1515 находим в директории storage папку с переданным ID

Контент

На время тестирования используем контент из директории storage кодеком h264 и отдавать контент с указанием временных диапазонов. 2020-01-01T00:00:00-2020-01-01T00:05:00, 2020-01-02T00:00:00-2020-01-02T00:05:00, 2020-01-03T00:00:00-2020-01-03T00:05:00.

При воспроизведении данной ссылки через плеер vlc, контент должен отдаваться единым потоком с указанием rtsp range по каждому файлу. Так же для теста протестируйте другими проигрывателями (пару штук), которые поддерживают RTSP.

Доступ к архиву по времени

Мы не знаем как это работает, но знаем, что такой функционал есть. должна быть возможность указать время начало получения архива.

Допустим у нас есть видеоконтент с 2020-01-01T00:00:00 по сегодняшний день, по умолчанию ссылка должна отдавать видеопоток с самой ранней даты, но должна быть возможность какими-то внутренними средствами RTSP передать начальную дату, допустим 2020-08-01T00:00:00 и предоставить видеопоток уже с текущей даты.

Также данный архив будет периодически меняться, и необходимо сделать так, чтобы когда от клиента приходит запрос на получения архива, данные о файлах и их временных периодах обновлялись с файла json Например

```
[  
  "file1.mp4":{"start":timestamp,"end":timestamp},  
  "file10.mp4":{"start":timestamp,"end":timestamp},  
  "file1.mp4":{"start":timestamp,"end":timestamp}  
  ...  
]
```

Этот файл будем генерировать мы сами, вы можете сделать его самостоятельно.

Требования (выдержка из документа)

Протоколов различного уровня, а именно:

- Управляющего протокола SDP в соответствии с RFC 4566.
- Прикладных протоколов RTP/AVP в соответствии с RFC 3550, предпочтительно в режиме interleaved.
- Транспортных протоколов TCP/UDP (рекомендуемый - TCP). • RTSP packetization-mode = 0 или 1.
 1. Последовательность кадров (aop) в видеопотоке не должна состоять из одних i-кадров, т.е. между i-кадрами обязательно наличие p- кадров при scale= 1.
 2. Перед каждым I-кадром должны присутствовать sps/pps параметры.
 3. Рекомендуется при Scale>4 отдавать видеопоток только i-кадрами, а также:
 - При перемотке изменение таймстемпа не должно происходить в отрицательную сторону, а также не должно происходить существенное изменение timestamp - например более чем на 2000 миллисекунд (в измерительной базе 1KHz timebase).
 - При перемотке на Scale: N разница между кадрами по времени должна сократиться в N раз, скорость передачи видеопотока должна возрасти в N раз (для N не более 4).
 - При перемотке на Scale: N разница между i-кадрами по времени должна сократиться в N раз (для N более 4), скорость передачи видеопотока должна соответствовать временным меткам.
 4. Взаимодействие по протоколу RTSP осуществляется с поддержкой следующих определений:
 - Типы авторизации:
 - Basic authorization или Digest authorization;
 - Методы:
 - OPTIONS - не обязателен;
 - DESCRIBE - обязателен и должен возвращать SDP (время получения ответных данных - не более чем 20 сек) при наличии запрашиваемого контента и заголовков User-Agent с реальным значением конкретного видеорегистратора (видеосервера).
 - SETUP -обязателен; • PLA Y - обязателен, управляющие заголовки:
 - Range = "Range" ":" "clock" "=" "utc";
 - Scale = "Scale" ":" ["-"] 1 *DIGIT ["." *DIGIT] - направление и масштаб скорости воспроизведения, изменяемый на стороне видеорегистратора (видеосервера):
 - значение Scale в диапазоне [-32;32] - ускоренное воспроизведение (рекомендуемые значения 121,1411,811,161,[321]);
 - значение Scale не более -1 - реверс;
 - trick play с заголовком/методикой Speed - не должен использоваться и не поддерживается. • • • PAUSE - обязателен; TEARDOWN - обязателен; GET_PARAMETER - обязателен, параметры в теле запроса/ответа: position текущее абсолютное время воспроизведения, должно соответствовать метке потока, в том числе если она продублирована наложением в самом кадре. Ответ в заголовках в следующем виде:

Range: clock=20130823T020000Z-

1. Расширение к базовому взаимодействию SDP: • Управление треками согласно протоколу. • URL трека не может меняться в рамках сессии. • Обязательные атрибуты:

- все доступные типы медиа потоков с указанием протокола и типа данных, например t=video ORTP/A VP 97:
- в случае если поток динамический, то типы медиа потоков (в соответствии с RFC 6184 (типы 96, 97)) должны указываться через атрибут rtpmap, например a=rtpmap:96 H264/90000 и fmp (с указанием packetization mode [none-interleaved], profile-level-id, sprop-parameters-set), например a=fmp:96 packetization-mode= 1; profile-level-id=420029; sprop-parameter-sets=ZOIAKeKQ FgJNgScF AQXh4kRU ,aM48gA==. Принят следующий метод информирования о доступных границах воспроизведения в рамках атрибута/поля range протокола SDP: a=range:clock=20130823 T020000Z-20130823T201228Z. В случае если запись архива не является целостной (имеет разрывы) в виде ряда значений, например:

a=range:clock=20130823 T020000Z- 20130823T201228Z;clock=201 30904T1 347082-201 30904T1 35513z 1IL11 a=range:clock=20130823 T020000Z-

20130823T201228Z;clock=201 30904T1 358082-, где: • range - список доступных сегментов; • clock - доступный сегмент, в формате 'startUTC-endUTC', может быть открытым, если конечная точка еще не известна. 8. В качестве альтернативы к описанным требованиям допускается использование проприетарных протоколов различного уровня, атрибутов и определений, совместимых с компонентами подсистем и видеоядра ЕЦХД.

Логирование

Директория с логами указана в конфиге. Для каждой камеры должен быть свой файл лога по её id. Также лог самой программы. Возможно три вариант логирования `loglevel`

- 0 - логи отключены
- 1 - обычный логи
- 2 - DEBUG

Также логи должны ретироваться каждый день, и автоматически удаляться по количеству дней `rotate`.

Дополнительно

Если есть какие-то дополнительные параметры которые, могут влиять на стабильность соединения и их необходимо\можно менять в потоке RTSP или любые другие, необходимо их либо вынести в общий конфиг.