

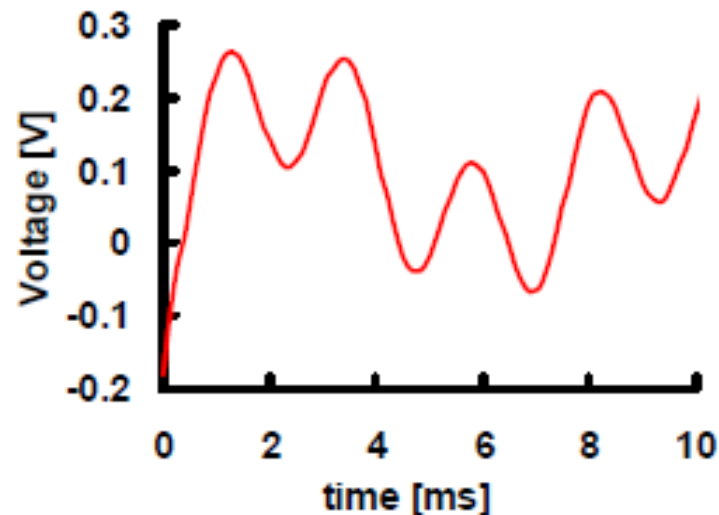
# IP телефония (VoIP)

Протоколи за разговори в реално  
време

# Теоретични основи на преноса на данни. Аналогови и цифрови сигнали.

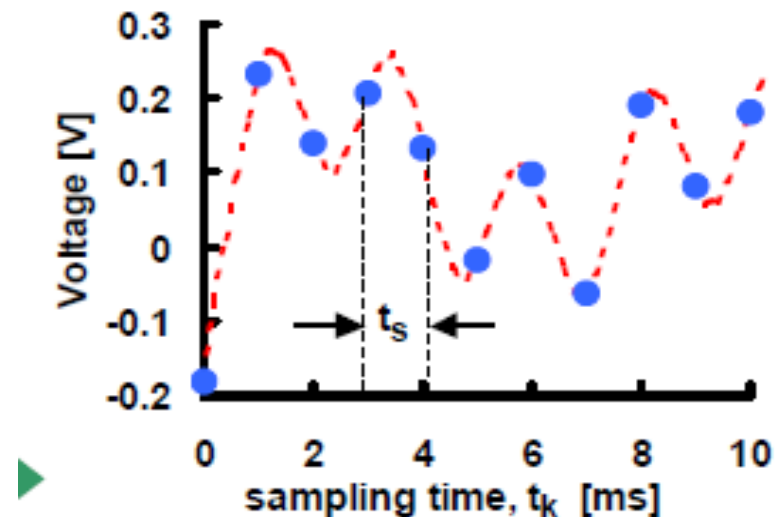
## Аналогов сигнал

Непрекъснатата функция  $V$  с непрекъснатата променлива  $t$  (напр. време) :  $V(t)$ .



## Цифров сигнал

Дискретна функция  $V_k$  от дискретна отчетна (sampling) променлива  $t_k$ ,  $k = \text{integer}$ :  $V_k = V(t_k)$ .



# Теоретични основи на преноса на аналоговите данни като цифрови

С други думи: **дискретните** данни могат да заемат само определени стойности (0-те и 1-те от **компютъра**).

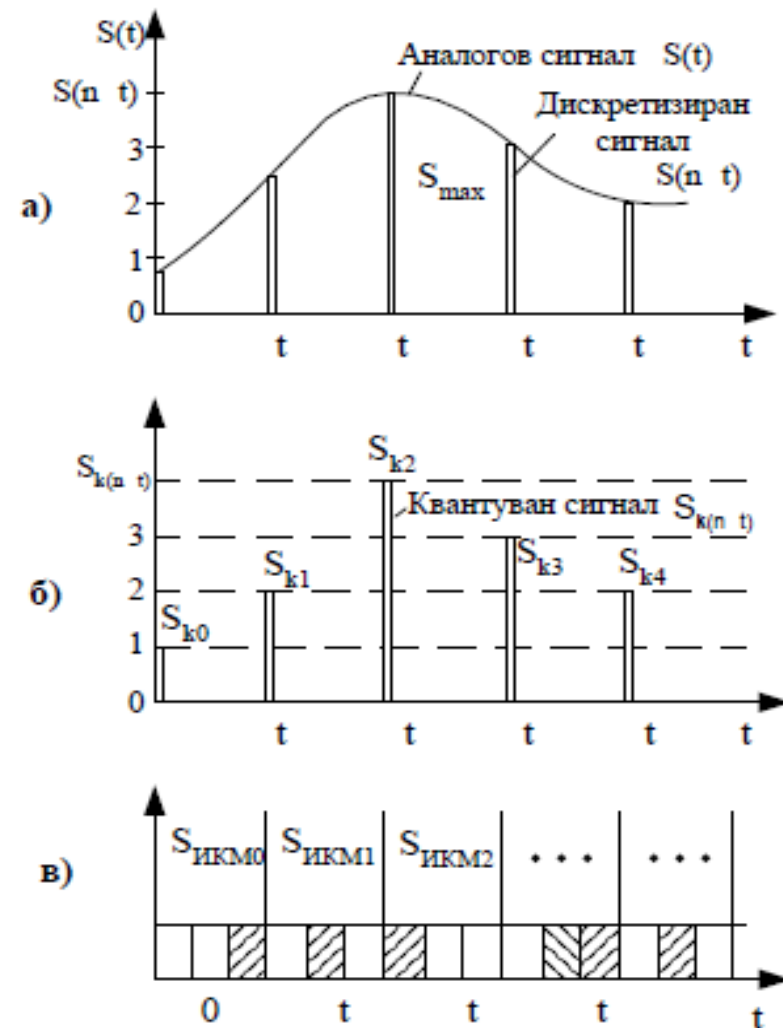
**Непрекъснатите** могат да заемат произволни стойности (в определени граници). Такива са **гласовите и изображенията** – подвижни и неподвижни.

Как ги представяме по компютърната мрежа. Това става с импулсно-кодова модулация (ИКМ).

# Импульсно-кодовая модуляция (ИКМ)

ИКМ (PCM - Pulse Code Modulation) е начин за преобразуване на аналогов сигнал в цифров.

1. Дискретизация на аналоговия сигнал чрез равномерни отчети (*sampling*).
2. Квантуване.
3. Кодиране на импулсите – получава се поредица от СИМВОЛИ.



# PCM. 64 kbit/s канал.

Аналоговите сигнали се цифровизират с помощта на кодек (**coder-decoder**).

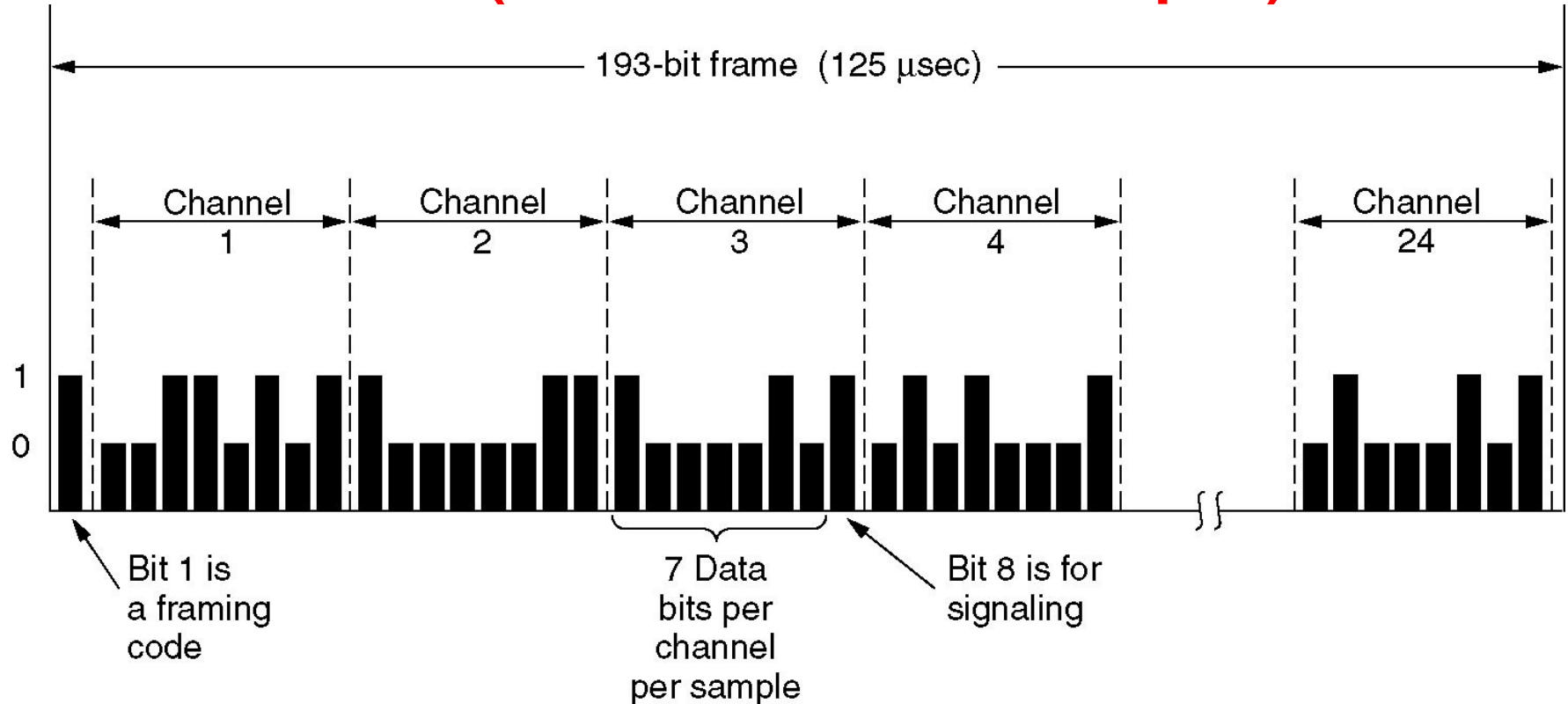
Получава се поредица от **8-битови числа**.

Кодекът произвежда **8000** отчета (samples) в секунда  
(**125  $\mu$ s/sample**).

Т.е: **64 kbit/s**

Според **Теоремата на Nyquist** това е достатъчно да се прихване цялата информация от един 4-kHz **телефонен канал**.

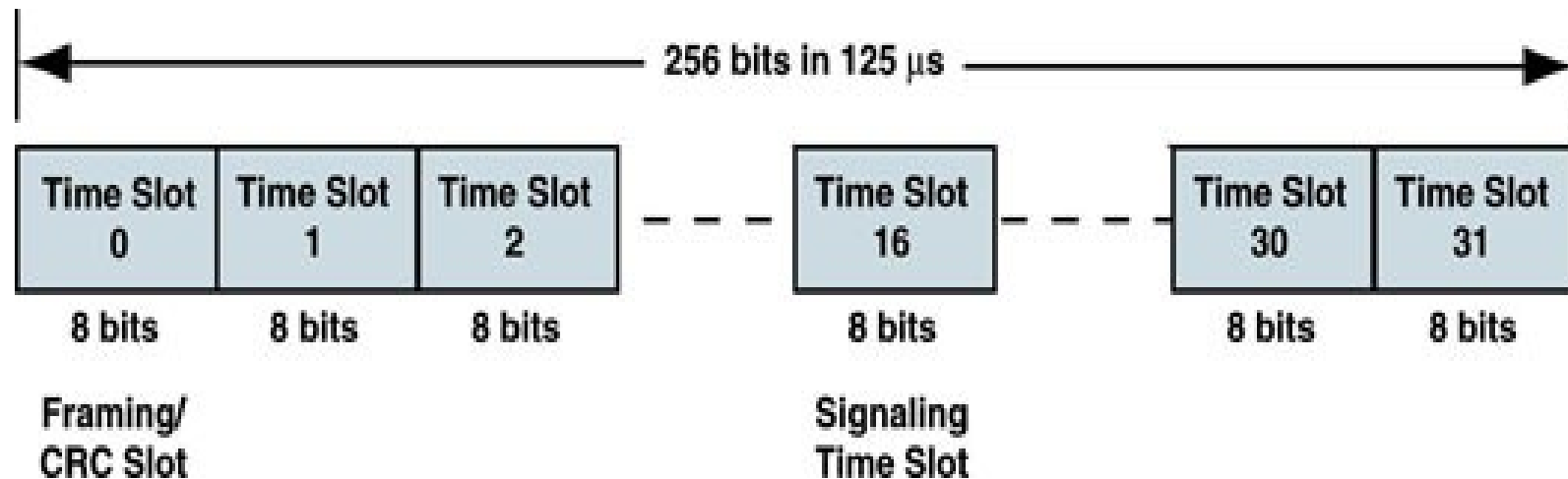
# TDM. (T1 - 1.544 Mbps)



Прилага се в Северна Америка, Япония, Корея.  
(7 x 8000 = 56,000 bps данни) + (1 x 8000 = 8000 bps) сигнална информация

(TDM. Припомнете си лекцията за Физ. Слой!)

# TDM в Европа. ( E1 - 2.048 Mbps)



E0	64 kbit/s
E1	2.048 Mbit/s
E2	8.448 Mbit/s
E3	34.368 Mbit/s
E4	139.264 Mbit/s
E5	564.000 Mbit/s

# 125 $\mu\text{s/sample}$ . Приложенията в реално време - VoIP.

Един характерен пример е IP телефонията или **Voice over IP** (VoIP).

Данните се генерират чрез събиране на отчети от микрофона, цифровизират се от конвертор на аналоговия сигнал в цифров (analog-to-digital - A/D).

Цифровите отчети се поставят в (гласови) пакети, които се предават по мрежата.

Т.е в мрежа с **комутация на пакети**, каквато е Интернет за разлика от традиционната телефонна мрежа, която е с **комутация на канали**.

**Припомнете си 1-та лекция.**



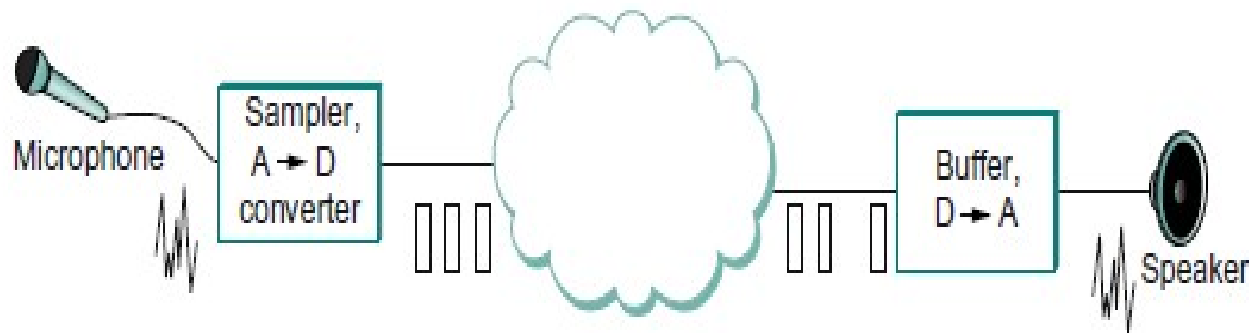
# 125 $\mu\text{s}/\text{sample}$ . Приложенията в реално време - VoIP.

При получателя възпроизвеждането трябва да се извърши със същата скорост - по един на **125  $\mu\text{s}$** .

Ако някой пакет пристигне по-късно поради закъснения по мрежата, той ще е излишен.

Не е възможно в среда като Интернет да се гарантира точно определено закъснение на всеки гласов пакет.

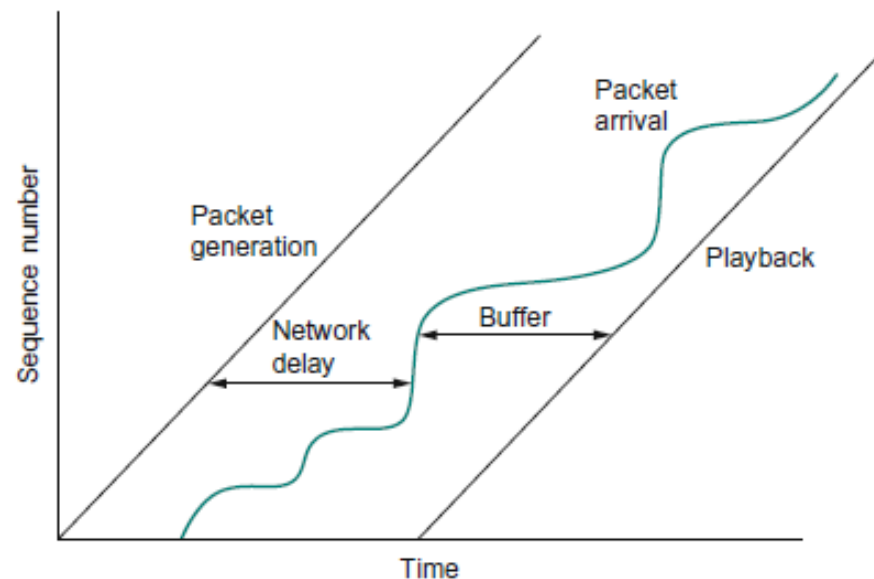
Начин за справяне с този проблем е **буфериране** на пакетите при получателя.



# 125 $\mu\text{s}/\text{sample}$ . Приложенията в реално време - VoIP.

Въпросът е колко време да се отложи възпроизвеждането при получателя.

За да се води нормален телефонен разговор, периодът от време между **изговаряне и чуване** отсреща трябва да е **150 ms**, но не повече от **300-400 ms**.



# 125 $\mu\text{s}/\text{sample}$ . Приложенията в реално време - VoIP.

Т.е трябва да се гарантира, че всички гласови пакети от даден разговор ще пристигнат в рамките на 300 ms.

Ако даден пакет пристигне **по-рано**, ще го **буферираме**, докато изтече периода от 125  $\mu\text{s}$  за възпроизвеждане.

Ако даден пакет пристигне **по-късно**, той е безполезен и се **изхвърля**.

# Фактори, влияещи на качеството при VoIP комуникациите.

## Големина на пакета.

Основно това са големина на пакетите, загуба на пакети, закъснение от мрежата, закъснение от чакане (в опашки), закъснение от джитер, тип на кодека и ехо.

Загубите на пакетите са сравнително ниски, когато техния обем е малък. Препоръчва се пакетизиране на **20 ms** речеви проби, което прави **20 байта** на пакет + *40 байта хедър* (*RTP=12 bytes + UDP=8 bytes + IP=20 bytes*).

# (Поносими) загуби на гласови пакети

Добрият кодек се справя със загубата на пакет по незабележим за потребителя начин.

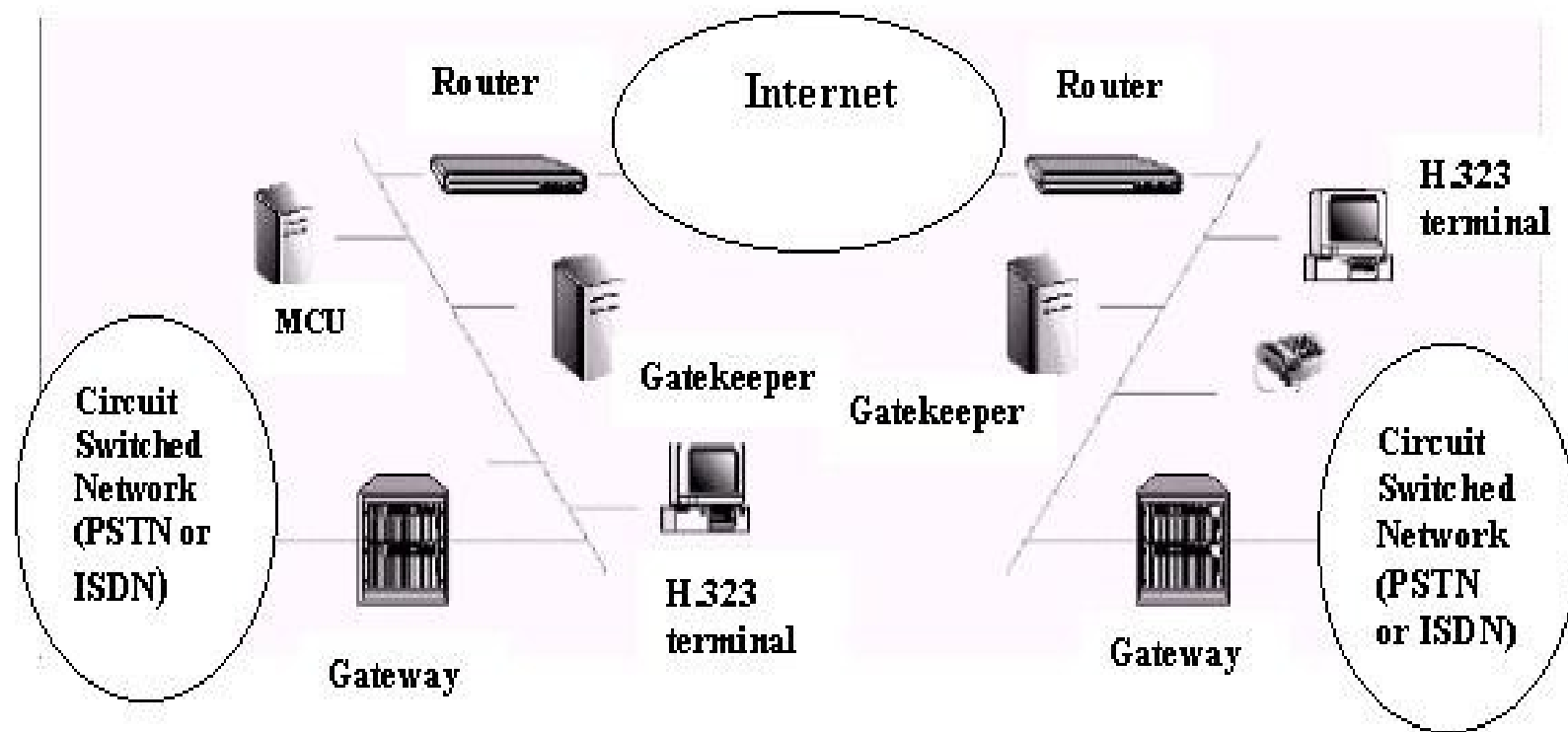
Загубата на пакети става сериозен проблем, когато е  $> 5 - 10 \%$ .

Препоръчва се  $< 5\%$  загуби за високо качество на звука.

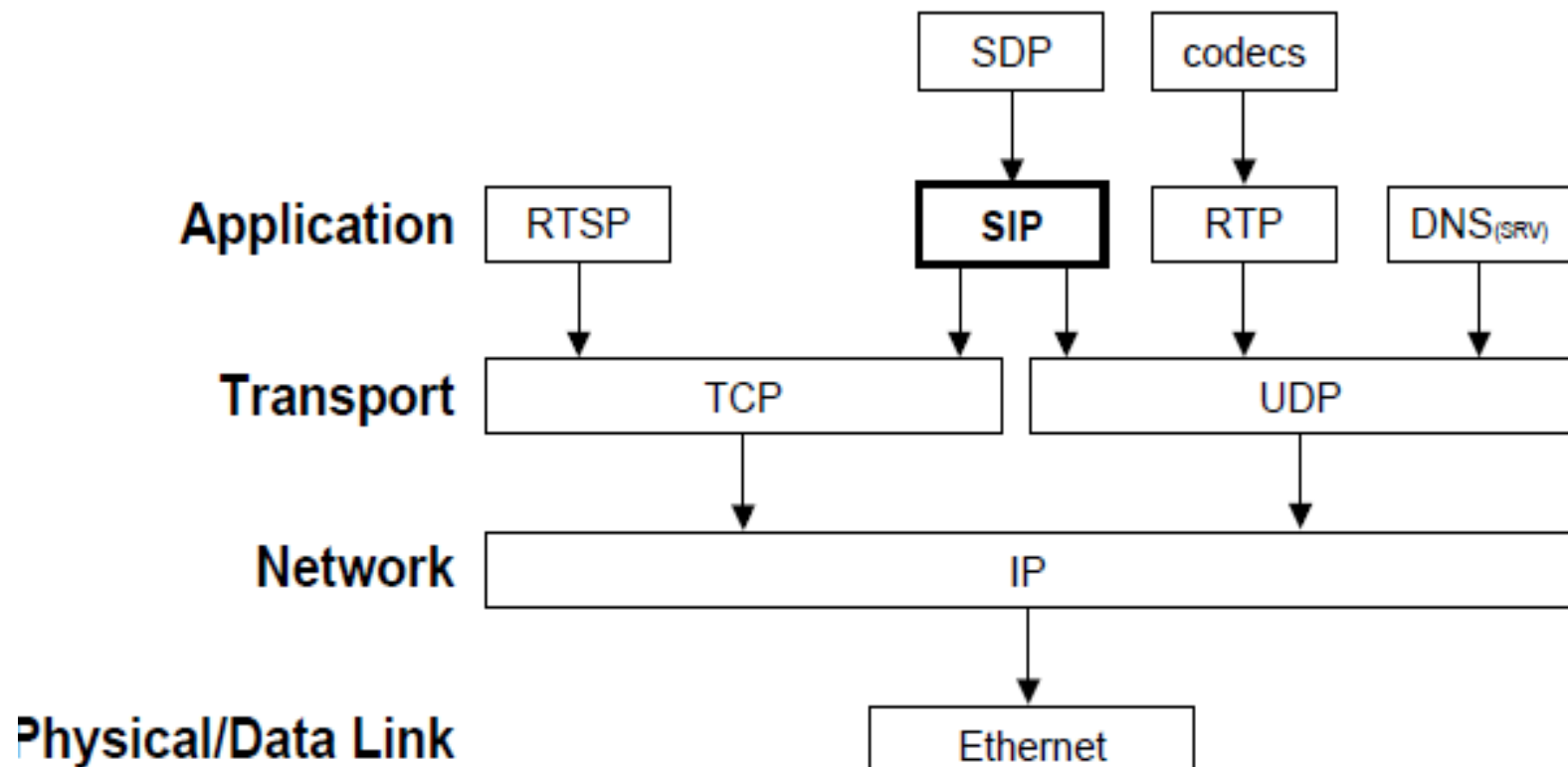
Внезапните (големи обеми - **burst**) влияят много по-зле на качеството отколкото случайните загуби на пакети.

# Протоколи за VoIP. H.323.

Стандарт на ITU, затова не се харесва на Интернет обществото.



# SIP и др. Чисто IETF (Интернет) решение.



# RTSP - Real-Time Streaming Protocol

RTSP е протокол клиент-сървър, който осигурява контрол при пренасянето на медийните потоци в реално време.



# Session Initiation Protocol (SIP)

SIP ([RFC 3261](#)) е “сигнален” протокол на приложния слой:

- създава, модифицира и терминира сесии с един или повече участници.

Използва се при гласови и видео комуникации, незабавни съобщения (instant messaging), игри и т.н.

Подобен е на HTTP:

- текстови съобщения;
- URLs, напр.: sip:user2@server2.com

# Компоненти на SIP

Потребителски агенти (User Agents – **UA**) – Клиенти, които правят запитвания и сървъри, които отговарят.

Сървъри:

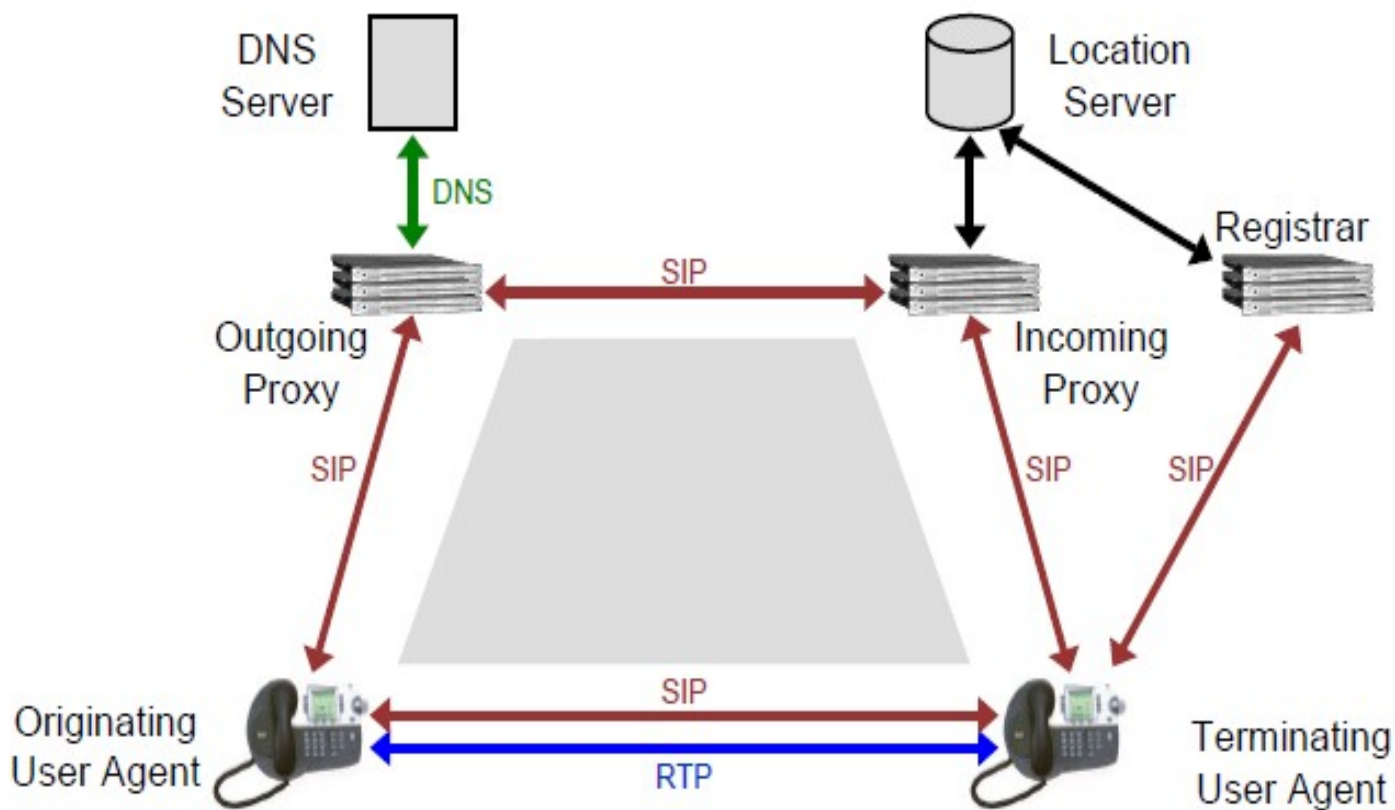
- Пренасочващ (**Redirect**) - UA генерира **3xx**<sup>1</sup> отговори на запитвания, насочвайки клиента към алтернативни URIs.
- Прокси
- Регистратор (**Registrar**) – UA сървър, който приема **REGISTER** запитвания и поставя информацията, която е получена в тези запитвания, в локационния сървър за домейна, който обслужва.

# Компоненти на SIP (прод.)

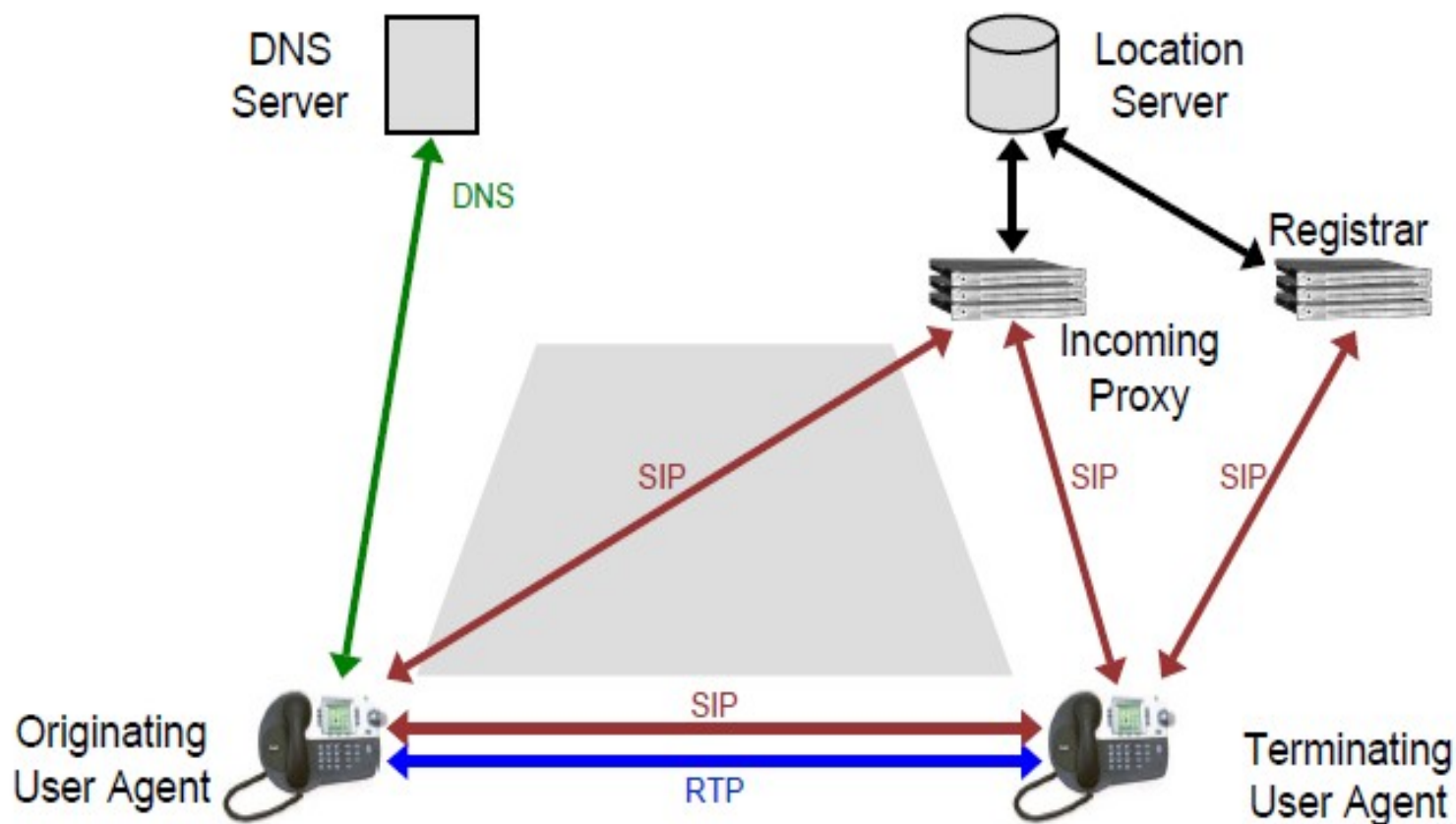
- За определяне на локацията (**Location**) – Услугата (сървър) за локация се използва от **SIP redirect** или **прокси** сървър за определяне на локацията на извикващия.
- **Gateway** (шлюз) – UA, осъществяващ връзка към друга мрежа – напр. the PSTN

<sup>1</sup>вж. Следващите слайдове

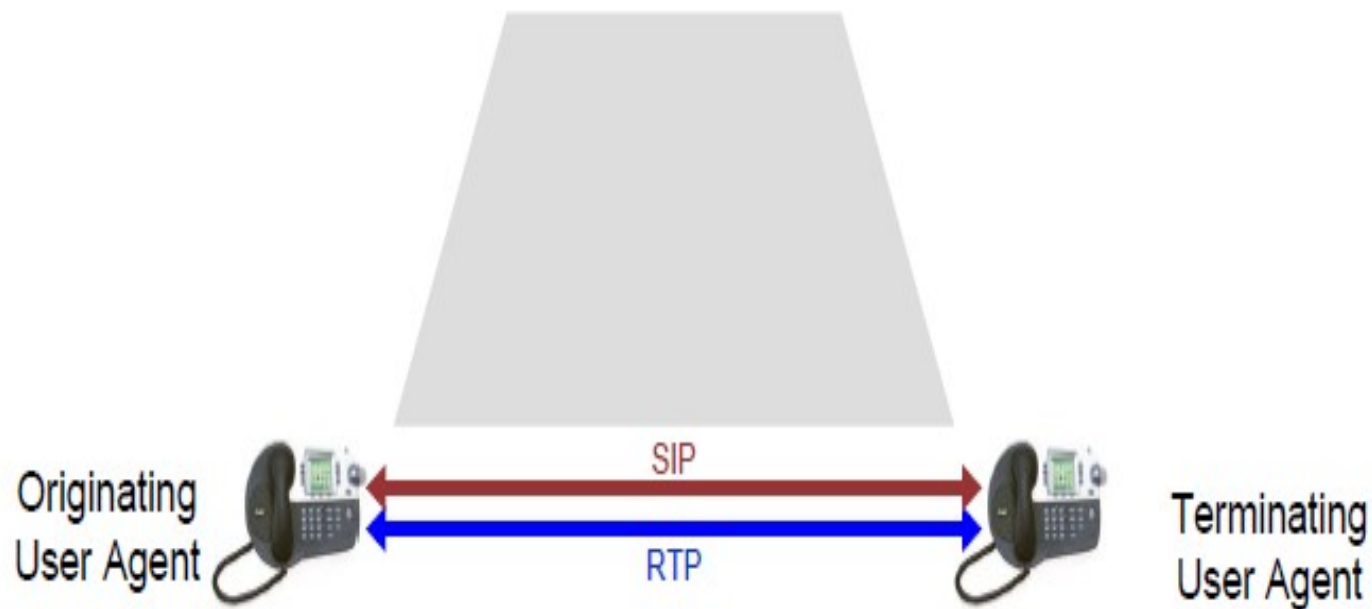
# SIP сценарий (трапец)



# SIP сценарий (триъгълник)



# SIP сценарий (Peer to Peer)



# SIP методи

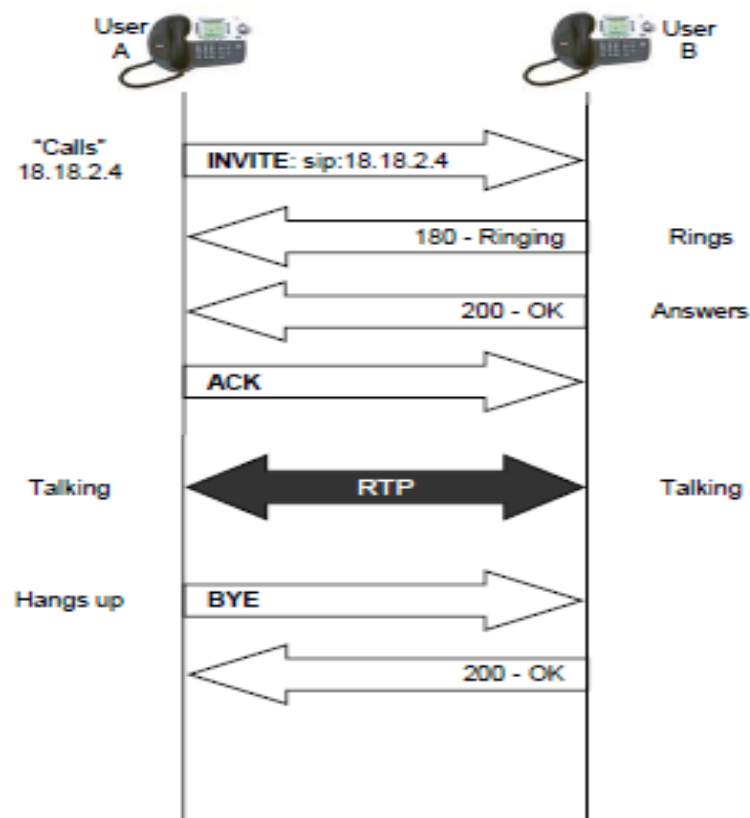
<b>INVITE</b>	Заявява сесия
<b>ACK</b>	Окончателен отговор на INVITE
<b>OPTIONS</b>	Пита за възможности на сървър
<b>CANCEL</b>	Отменя чакаща заявка
<b>BYE</b>	Терминира сесия
<b>REGISTER</b>	Изпраща адрес на потребител към сървър

# SIP отговори

1XX	Изпълнение на операцията	100 Trying
2XX	Успех	200 OK
3XX	Redirection	302 Преместено временно
4XX	Клиентска грешка	404 Not Found
5XX	Сървърска грешка	504 Server Time-out
6XX	Глобален отказ	603 Decline



# SIP потоци



# SIP INVITE

INVITE sip:user2@server2.com SIP/2.0

Via: SIP/2.0/UDP pc33.server1.com;branch=z9hG4bK776asdhds

Max-Forwards: 70

To: user2 <sip:user2@server2.com>

From: user1 <sip:user1@server1.com>;tag=1928301774

Call-ID: a84b4c76e66710@pc33.server1.com

CSeq: 314159 INVITE

Contact: <sip:user1@pc33.server1.com>

Content-Type: application/sdp

Content-Length: 142

# SIP response

SIP/2.0 200 OK

Via: SIP/2.0/UDP site4.server2.com;branch=z9hG4bKnashds8;received=192.0.2.3

Via: SIP/2.0/UDP

site3.server1.com;branch=z9hG4bK77ef4c2312983.1;received=192.0.2.2

Via: SIP/2.0/UDP pc33.server1.com;branch=z9hG4bK776asdhds;received=192.0.2.1

To: user2 <sip:user2@server2.com>;tag=a6c85cf

From: user1 <sip:user1@server1.com>;tag=1928301774

Call-ID: a84b4c76e66710@pc33.server1.com

CSeq: 314159 INVITE

Contact: <sip:user2@192.0.2.4>

Content-Type: application/sdp

Content-Length: 131

# Session Description Protocol (SDP)

SDP (RFC 4566) описва мултимедийните сесии: оповестяване, покана и др.

SDP включва:

- Типа медия (видео, аудио и др.)
- Транспортният протокол (RTP/UDP/IP, H.320 и др.)
- форматът на медията (H.261 видео, MPEG видео и др.)
- Информация за получателя (адреси, портове, формати и т.н.)

# SDP. Описание на сесията.

Protocol Version ("v=")

Origin ("o=") .....11

Session Name ("s=") .....12

Session Information ("i=") .....12

URI ("u=") .....13

Email Address and Phone Number ("e=" and "p=") .....13

Connection Data ("c=") .....14

Bandwidth ("b=") .....16

Timing ("t=") .....17

Repeat Times ("r=") .....18

Time Zones ("z=") .....19

Encryption Keys ("k=")

Attributes ("a=")

Media Descriptions ("m=")

# SDP. Пример.

v=0

o=jdoe 2890844526 2890842807 IN IP4 10.47.16.5

s=SDP Seminar

i=A Seminar on the session description protocol

u=<http://www.example.com/seminars/sdp.pdf>

e=j.doe@example.com (Jane Doe)

c=IN IP4 224.2.17.12/127

t=2873397496 2873404696

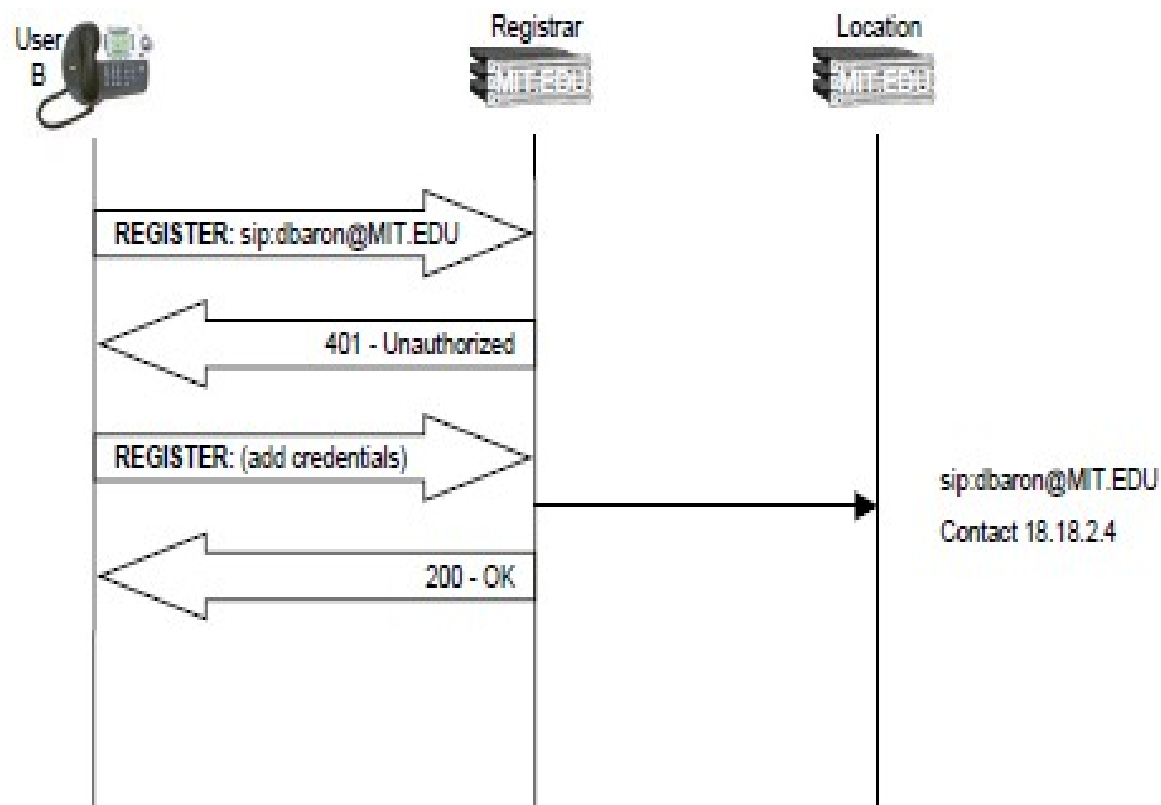
a=recvonly

m=audio 49170 RTP/AVP 0

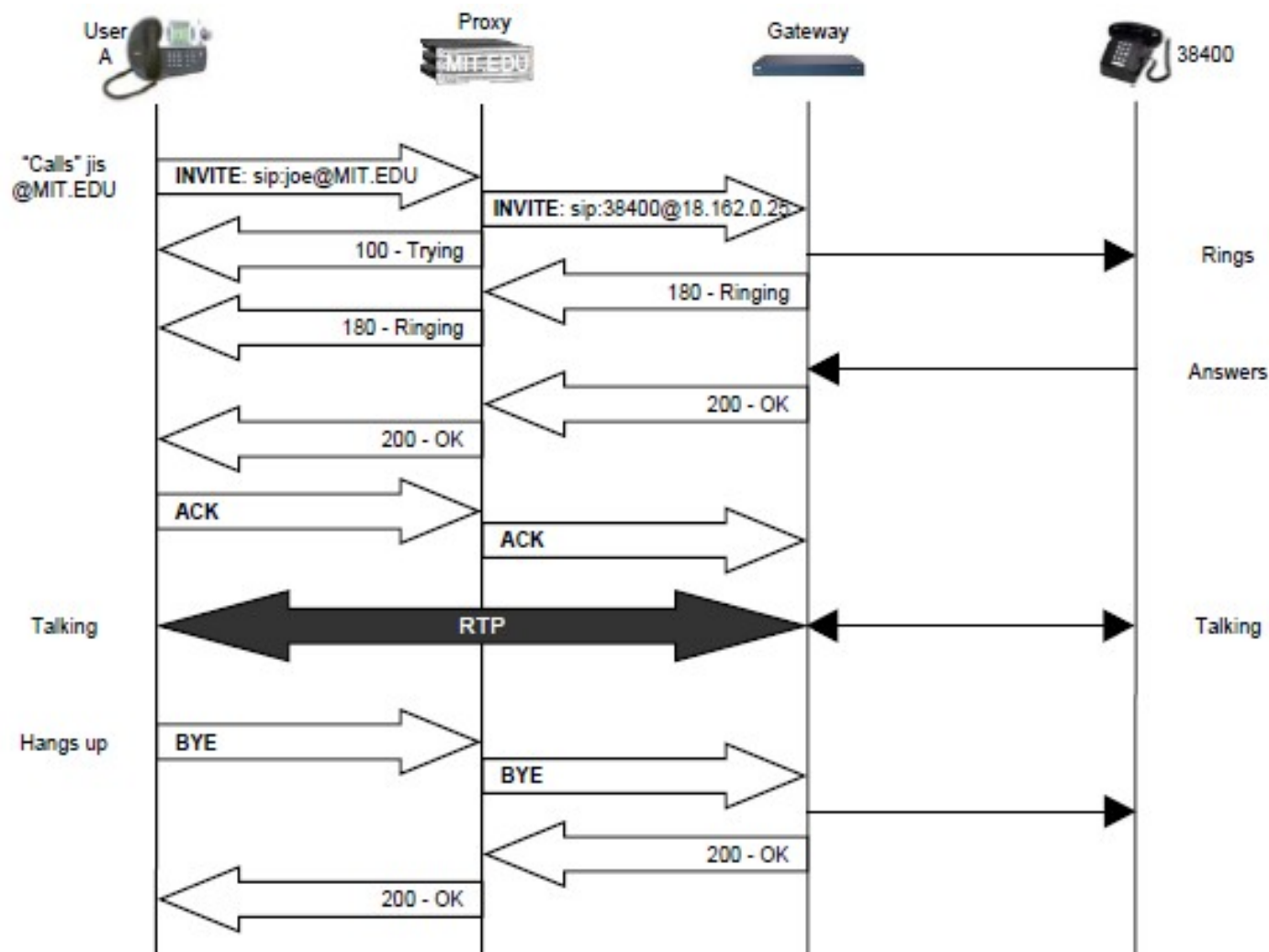
m=video 51372 RTP/AVP 99

a=rtpmap:99 h263-1998/90000

# SIP потоци. Регистрация.



# SIP потоци през Gateway





# Real-time Transfer Protocol (RTP)

RTP (RFC1889, обновен с RFC 3550, RFC 3551, RFC 7007 и RFC 5761) осигурява пренос “от край до край” на аудио и видео данни в реално време.

RTP сам по себе си не осигурява механизми за гарантиране на навременно доставяне на съобщенията, както и за качество на услугата (QoS).

RTP разчита на протоколите от по-ниско ниво (UDP, TCP и др.).

RTP осигурява подходяща функционалност за пренос на съдържание в реално време, напр., времеви печат (timestamp) и контролни механизми за синхронизиране на различни потоци, зависими от времето.

# Компоненти на RTP

Real Time Protocol (**RTP**): пренася данните в реално време.

Real Time Control Protocol (**RTCP**): следи за качеството на услугата и доставя информация за участниците.

# Приложения на RTP

## Мултикаст аудио конференция

Водещият получава мултикаст адрес и чифт портове един за аудио и един за контролни (RTCP) пакети, които се разпространяват сред участниците. Всеки участник изпраща по 20ms данни (пакети).

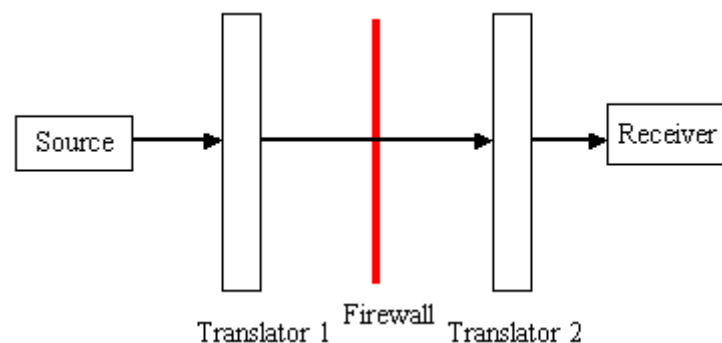
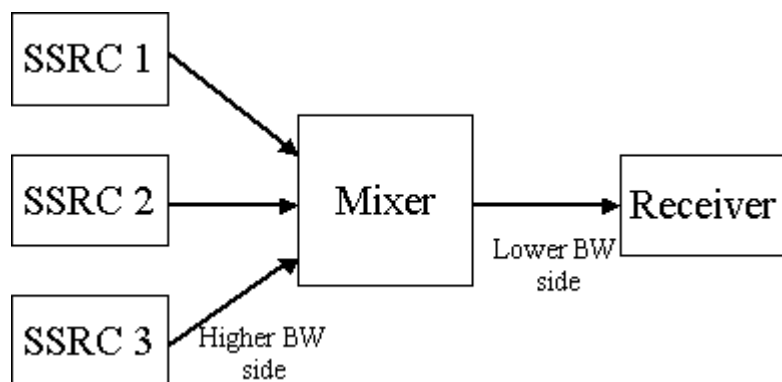
## Аудио и видео конференция

Създават се отделни RTP сесии и RTCP пакети за аудио и видеото, като се използват два различни сокета - UDP порт/мултикаст адрес.

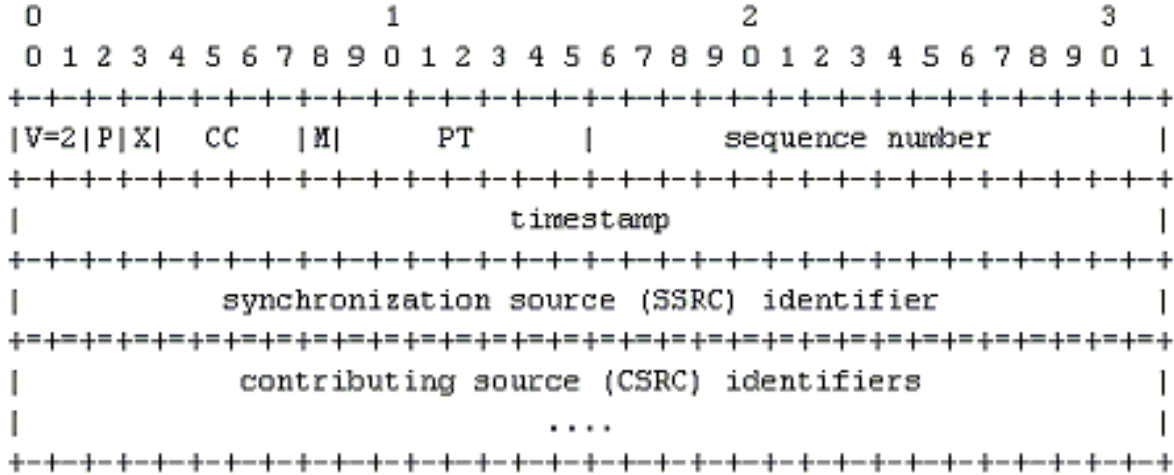
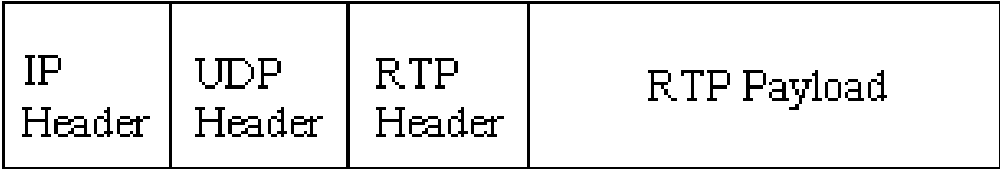
## Mixers and Translators (Миксери и транслатори)

Ако различните участници са свързани към мрежи с различна скорост или са зад защитна стена.

# Миксер и транслятор



# Структура на RTP пакета. Хедър.



# Полета на RTP хедъра

extension (X): 1 bit

Ако extension=1, фиксираният хедър е следван от едно и само едно удължение на хедъра.

CSRC count (CC): 4 bits

CSRC count съдържа броя на CSRC идентификаторите, които следват фиксирания хедър.

payload type (PT): 7 bits

Това поле дефинира формата (напр. encoding) на RTP полето за данни (payload) и определя интерпретацията му от приложението. Това поле не мултиплексира отделни медии.

sequence number: 16 bits

sequence number се инкрементира с 1 на всеки изпратен RTP пакет с данни и може да се използва от приемника за разпознаване на загуби на пакети и да възстанови последователността от пакети. Началната стойност на sequence number е произволна (random).

timestamp: 32 bits

timestamp отразява семплирания пример на първия октет в RTP пакета с данни. Този пример се извежда от часовник, който се инкрементира монотонно и линейно спрямо времето, за да има синхронизация и jitter изчисления.

SSRC: 32 bits

Полето SSRC идентифицира източника на синхронизация. Този идентификатор се избира произволно, защото не може да има два източника на синхронизация в рамките на една и съща RTP сесия да имат един и същ SSRC идентификатор.

CSRC list: 0 до 15 елемента, всеки по 32 bits

CSRC list идентифицира източниците на съдържанието за данни (payload) в пакета. Броят на идентификаторите е даден от полето CC. Ако има повече от 15 такива източници, идентифицират се само 15. CSRC идентификаторите се вмъкват от миксерите.

# Application Level Framing

За напасване към конкретно приложение. Настройват се полета в хедъра.

**extension** бит =1 означава, че след фиксираното заглавие имаме разширено. Допълнителни полета носят допълнителна информация за конкретното приложение.

Битът **marker** се дефинира от профила. Маркира определени събития като граници на фрейма в потока от пакети. В профила може да се дефинират допълнителни маркерни битове, съответно полета за данни.

# RTP control protocol (RTCP)

RTCP периодически предава контролни пакети до всички участници в сесията. Изпълнява следните функции:

Обратна връзка за качеството на данните.

Носи идентификатор за RTP източника - CNAME. SSRC може да се променя, но CNAME остава същия – идентифицира участника в сесията. RTCP може да съдържа и допълнителна информация като email адрес на участник.

Броят на участниците се използва за изчисляване на скоростта, с която се изпращат пакетите. При повече участници отделен източник може да изпраща пакети по-рядко.



# Типове RTCP пакети

**SR:** Sender report, предаване и приемане на статистика от участниците - предаватели

**RR:** Receiver report, приемане на статистика от участниците, които не са активни предаватели

**SDES:** Source description, вкл. CNAME

**BYE:** Край на участието

**APP:** Функция, специфична за приложението

# SIP и IPv6

<https://edvina.net/sipv6/protocol/>

IPv6 в SIP URI:

sip:6000@[2620:0:2ef0:7070:250:60ff:fe03:32b7]:5060;  
transport=tcp

IPv6 във Via хедъри (прокси):

Via: SIP/2.0/TCP

[2620:0:2ef0:7070:250:60ff:fe03:32b7]:5060;branch=z9  
hG4bK4882ebf2298267bc4ba97d222289760c.1;rport

# SIP и IPv4/IPv6 DNS

...

\_sip.\_tcp SRV 20 0 5060 sip1.example.com

SRV 0 0 5060 sip2.example.com

\_sip.\_udp SRV 20 0 5060 sip1.example.com

SRV 0 0 5060 sip2.example.com

sip1 IN A 192.0.2.1

sip1 IN AAAA 2001:db8::1

sip2 IN A 192.0.2.2

sip2 IN AAAA 2001:db8::2

...

# Записът SRV

Service ([SRV](#)) записът в DNS зоновия файл е дефиниран в [RFC 2782](#).

Необходим е на протоколи като SIP и Extensible Messaging and Presence Protocol (XMPP), напр. за jabber.

Има следния вид:

`_service._proto.name. TTL class SRV priority weight port target.`

`_sip._tcp.example.com. 86400 IN SRV 0 5 5060 sipserver.example.com.`

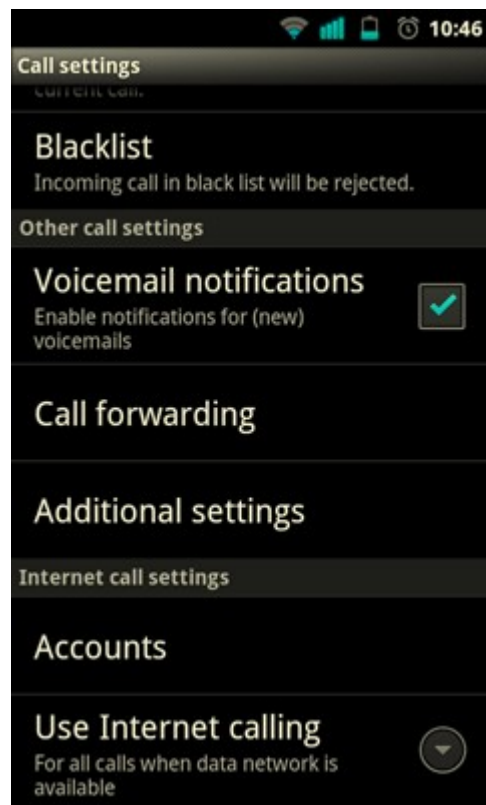
# SIP клиент за Android

<http://www.voipvoip.com/android/sip.html>

Първо трябва да инсталирате SIP клиент. След това:

Можете да правите гласови повиквания по internet до потребители, които също имат SIP акаунти (SIP адрес).

# Създаване на SIP акаунт



# Създаване на SIP акаунт

