

声网的WebRTC服务架构与实践

陈功

声网Agora WebRTC 架构师





陈 功

声网Agora WebRTC 架构师

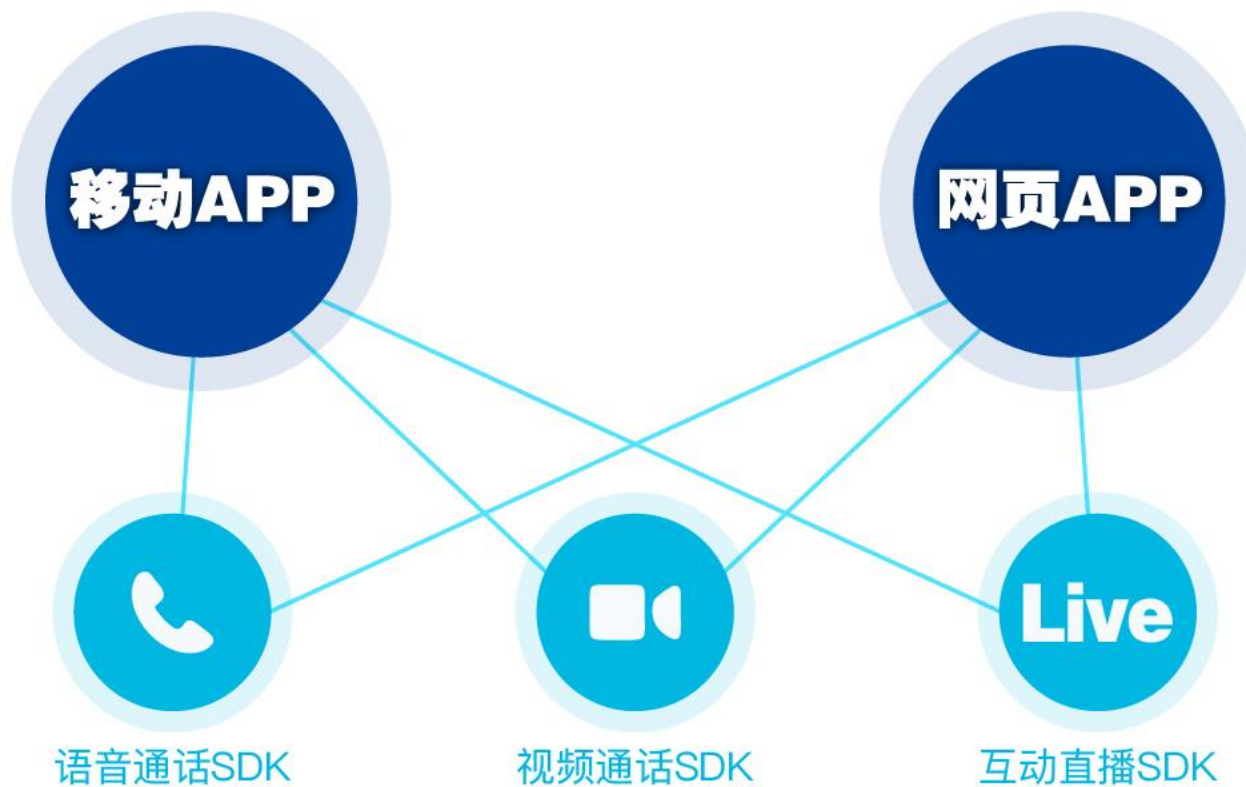
毕业于中国科学技术大学，Ph.D

原Intel 服务器事业部多媒体架构师，主导WebRTC视频会议解决方案搭建。

曾任职Marvell视频部门，研究多媒体系统框架，参与Google TV等项目。

超过10年的音视频相关领域经验。





实时云服务 Real-time Communication as a Service





10万开发者



10亿终端用户



亿级分钟通话/天

MOMO

阿里巴巴
Alibaba.com

meet me

花椒直播

meitu 美图

小米

一直播

PANDA

迅雷
XUNLEI

hike

gravy.

饭局狼人杀
相爱·相杀

4399

闲徕互娱
XIANLAIHUYU

小米互娱
MI ENTERTAINMENT

西山居

球球大作战

乐逗游戏

人人乐

凡普金科
FINUP

UCLLOUD

网宿科技

appMagics

faceunity

amazon
web services

又拍云

百度云

金山云
WWW.KSYUN.COM

融云
RONGCLOUD

KK

陪我

触手

blued

FOTOABLE

Holla

FOXUC 网狐

荔枝 FM

LIVESTAR

Va

SENDING.IO

V-CUBE

TAL 好未来

咸鱼家

TCL

鱼泡泡

Flurry

CCtalk
汇聚天下网师

贪吃蛇大作战

ABC360

脑穿越

Lenovo

Stager Live

Faceu
脸萌 搞怪

中国移动 | 中移在线
China Mobile

途牛
tuniu.com

FREE WhatsCall
FREE Global Calls

百合

ARHT
MEDIA

去哪儿? Gunar.Com
聪明 你的 旅行

掌门101

视

讯飞开放平台



Web Client App

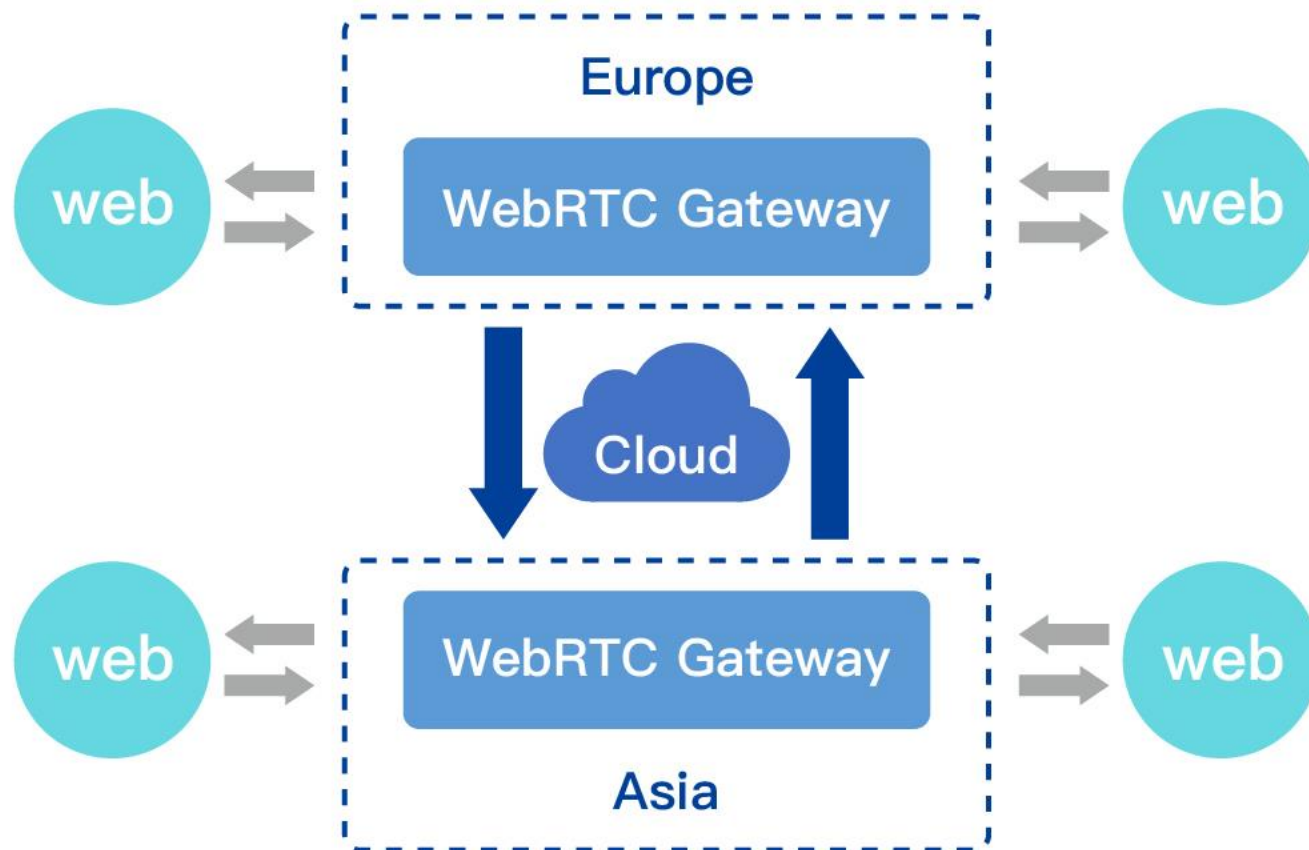


Agora Web SDK



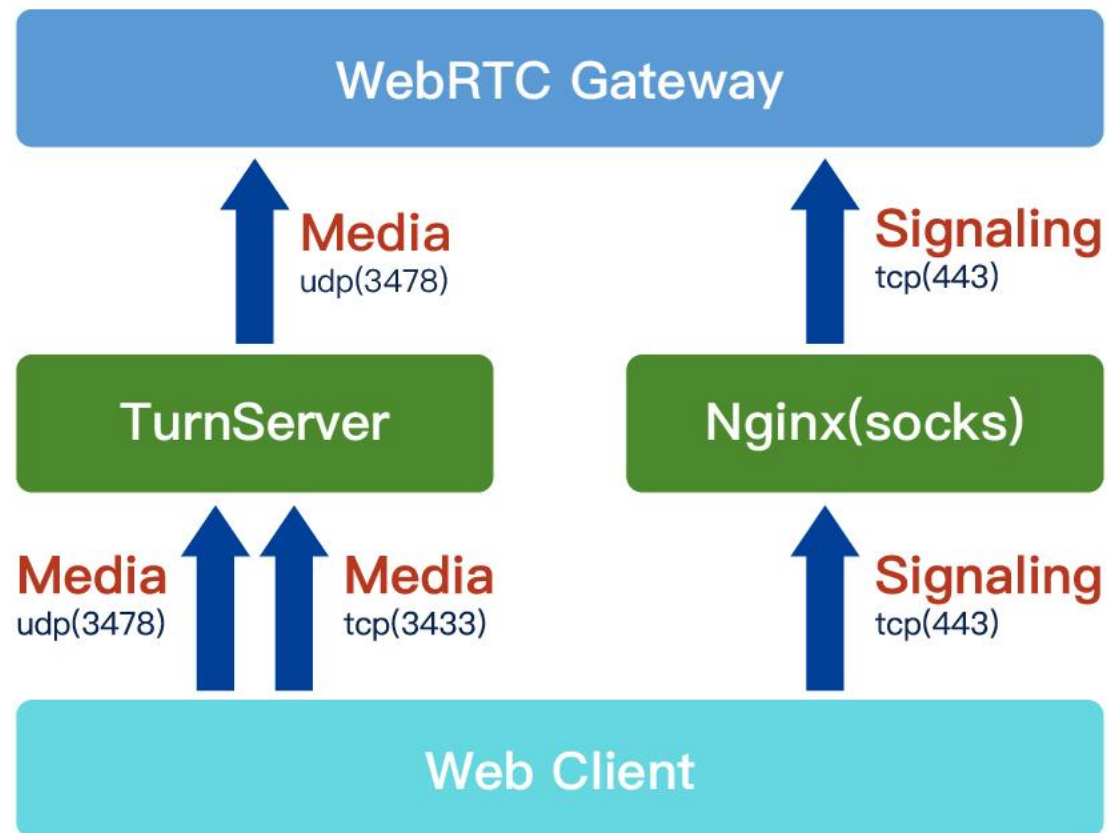
WebRTC 服务架构

- 媒体网关负责多路流分发处理
- 利用Agora传输网络优势
- 节省用户上行带宽
- 保障低延时通信

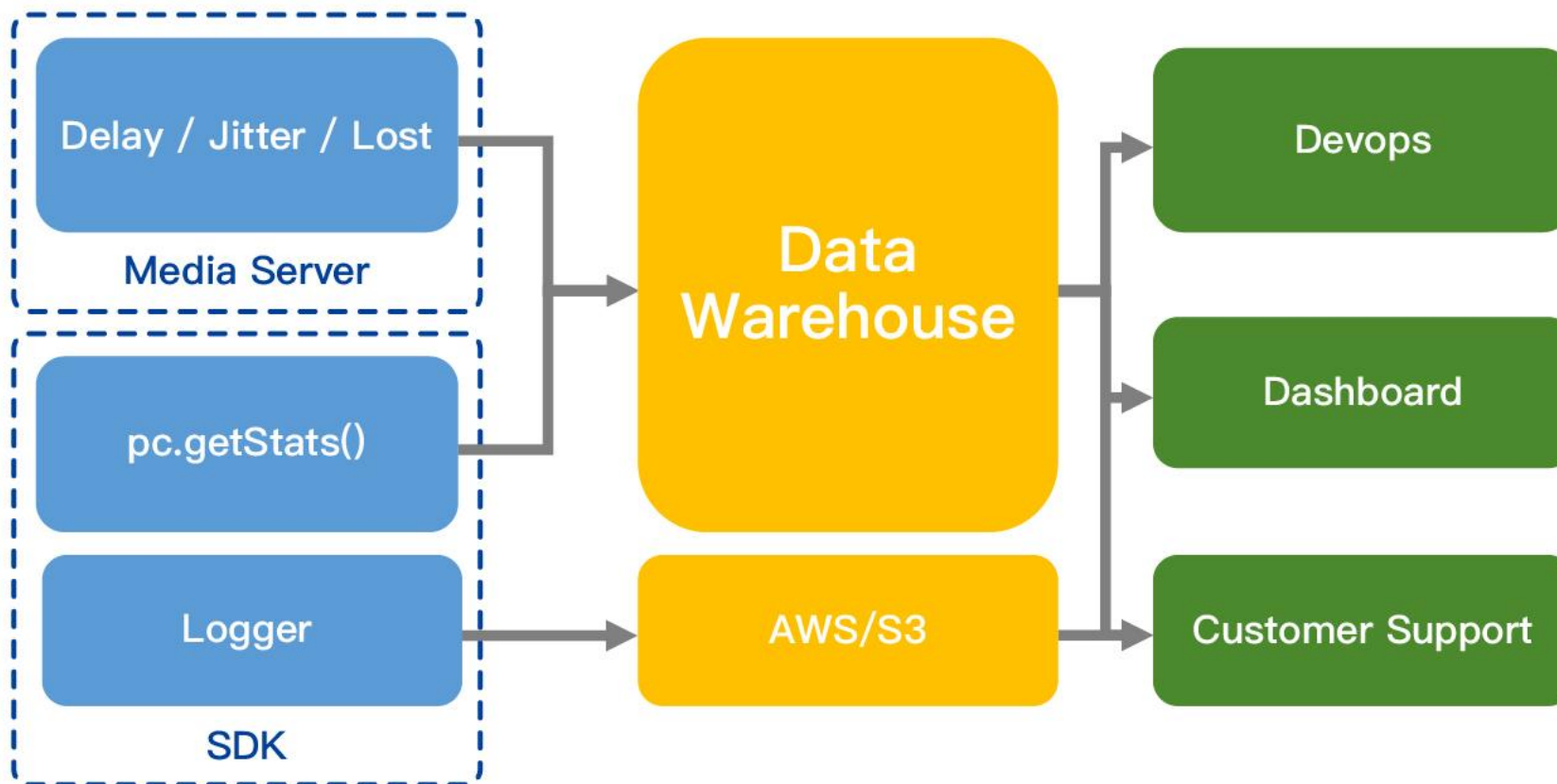


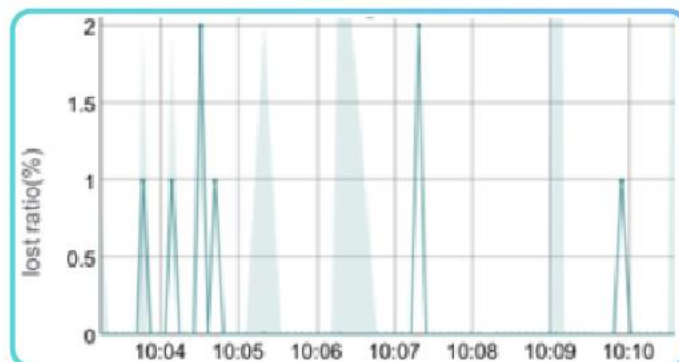
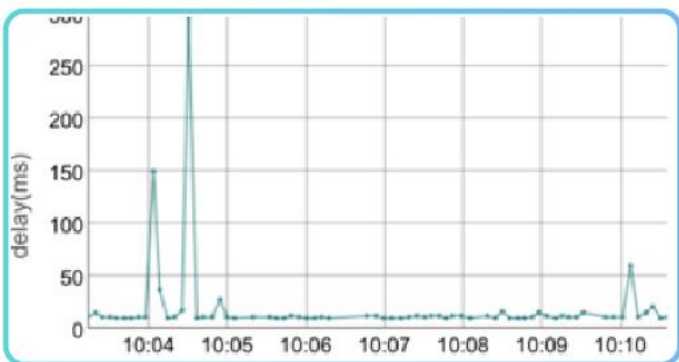
企业网服务架构

- 应对严格的防火墙限制
- 支持私有化部署中转服务器



数据服务





数据服务

SD-RTN

实时虚拟通信网

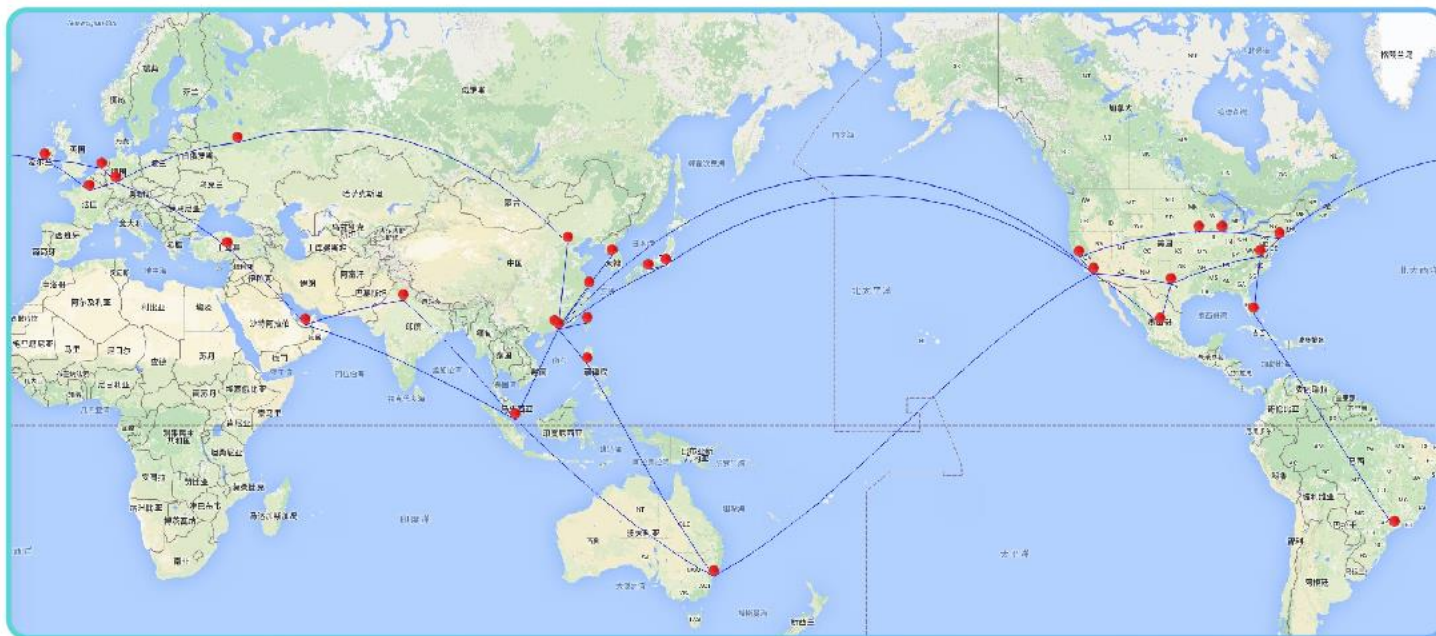
基于UDP协议，延时可控

全球近200个数据中心，分布式架构

实时监控传输线路动态，选择最通畅线路进行传输

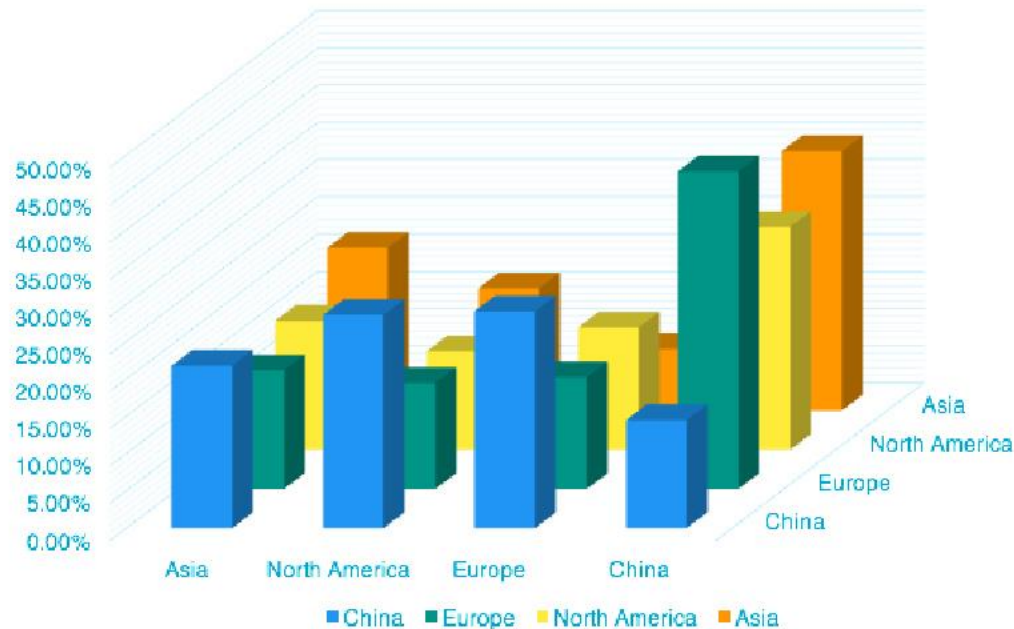
通过重传机制恢复丢失数据包，有效降低丢包率

全球端到端延迟76毫秒

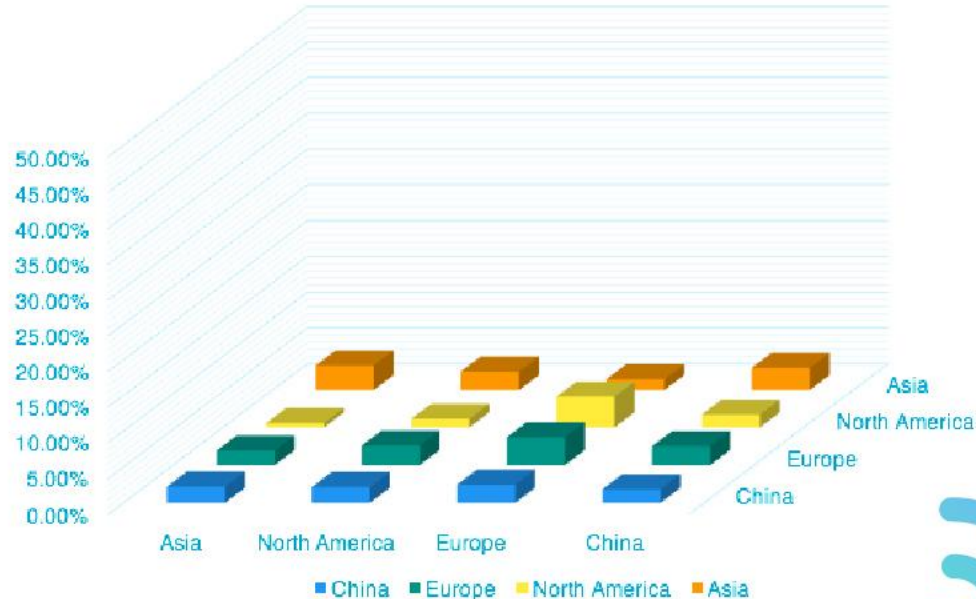


SD-RTN

原生P2P - 地区之间传输 不达标率



SD-RTN - 地区之间传输 不达标率

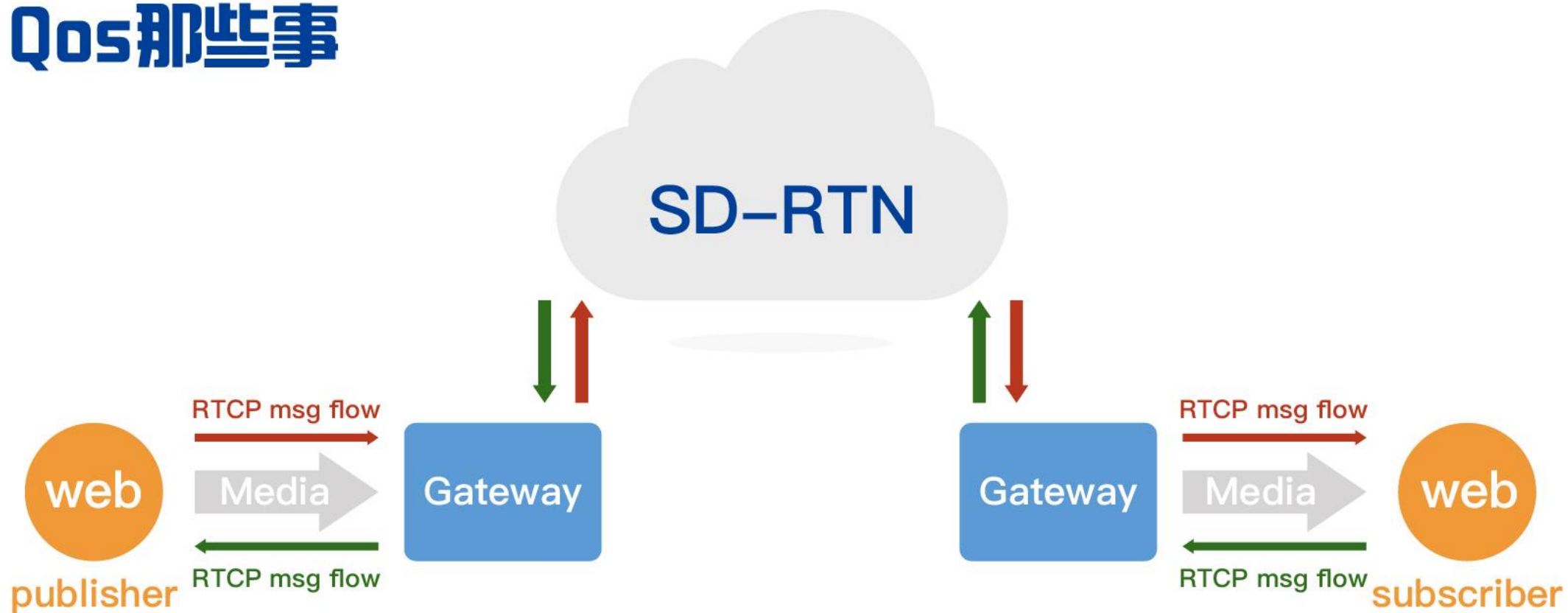


SD-RTN

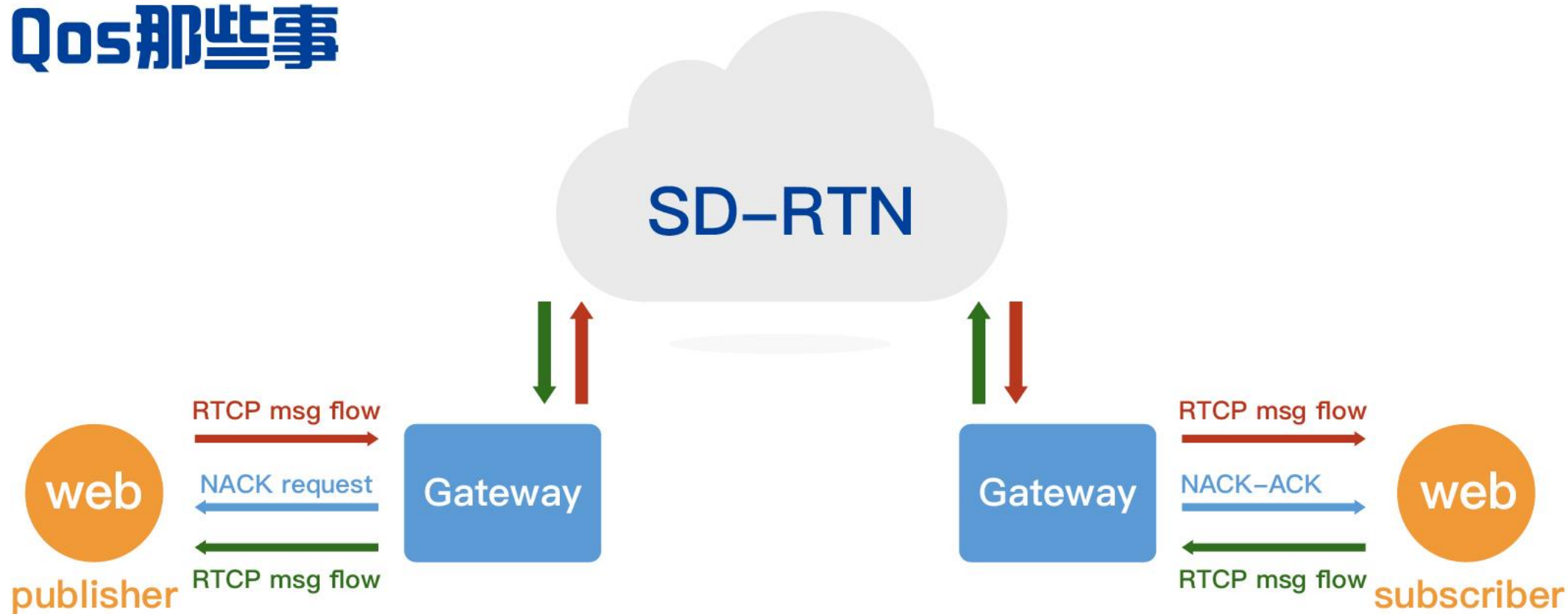
	其他网络	SD-RTN 实时网
延迟	延迟通常为5-20秒	延迟通常为200毫秒-2秒
互动	单向互动 通常为1个主播 N个观众	多向互动 最多7个主播 N个观众
连麦	不支持语音或视频实时沟通	可语音连麦或视频连麦
丢包对抗	无丢包对抗 网络丢包至30%时 视频将完全卡住	超强丢包对抗 网络丢包30%时 画面流畅



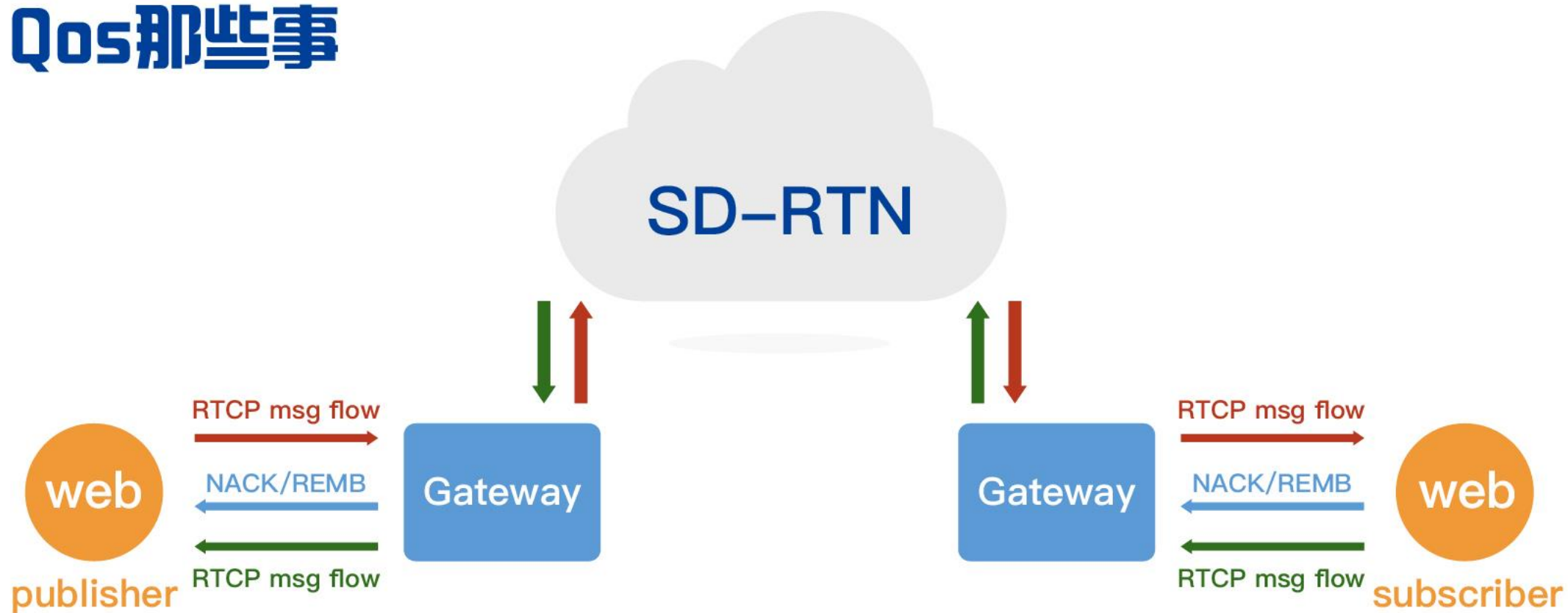
Qos那些事



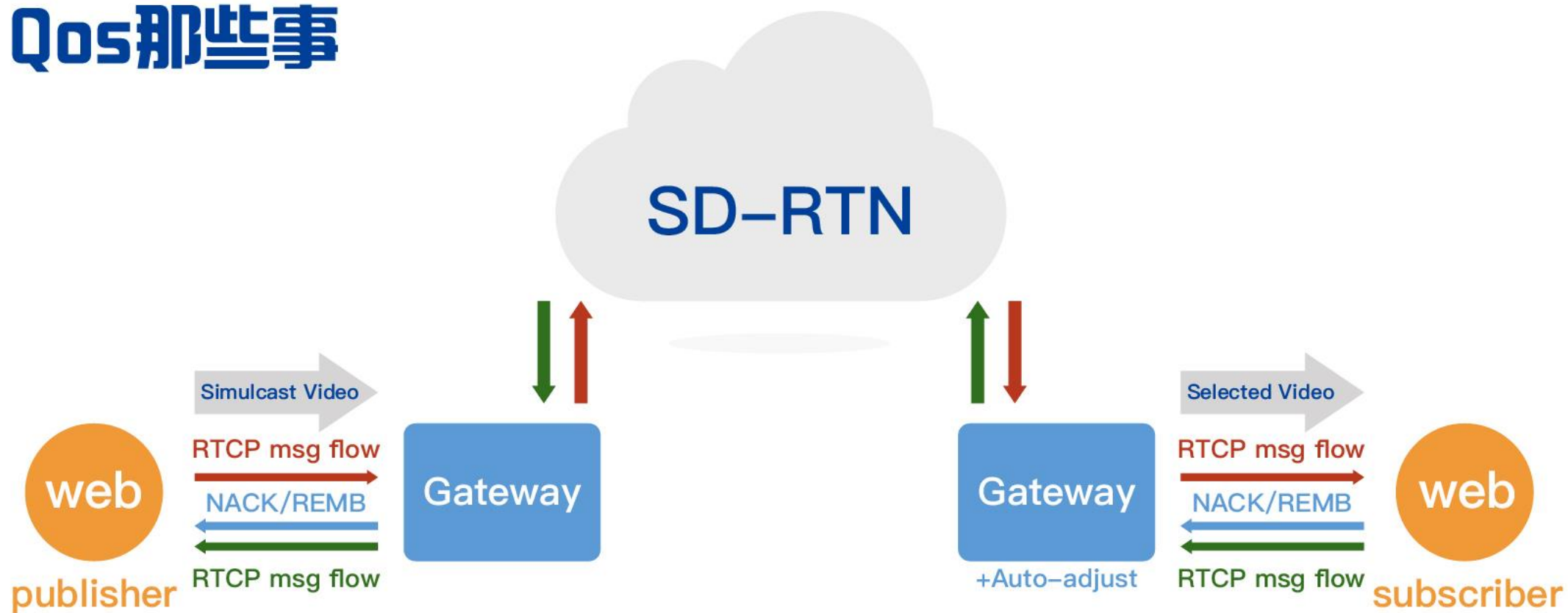
Qos那些事



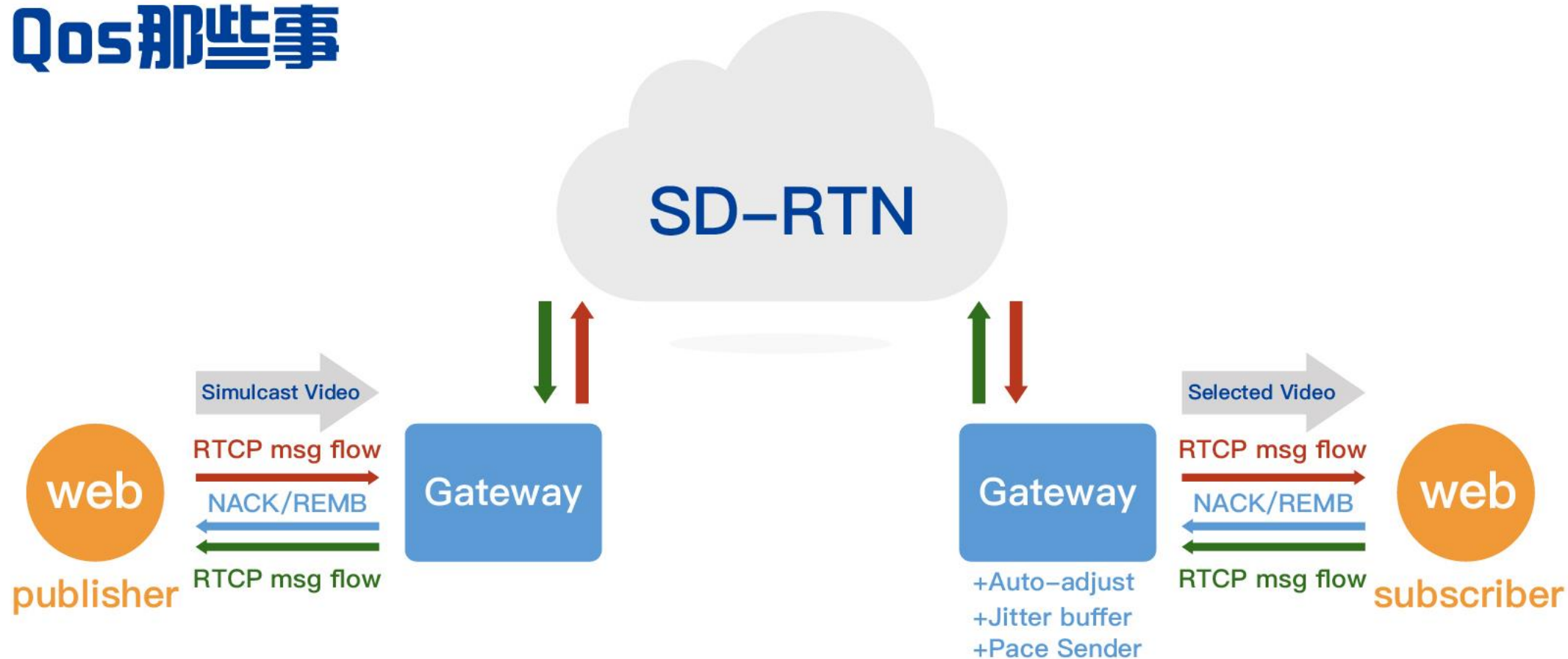
Qos那些事



Qos那些事

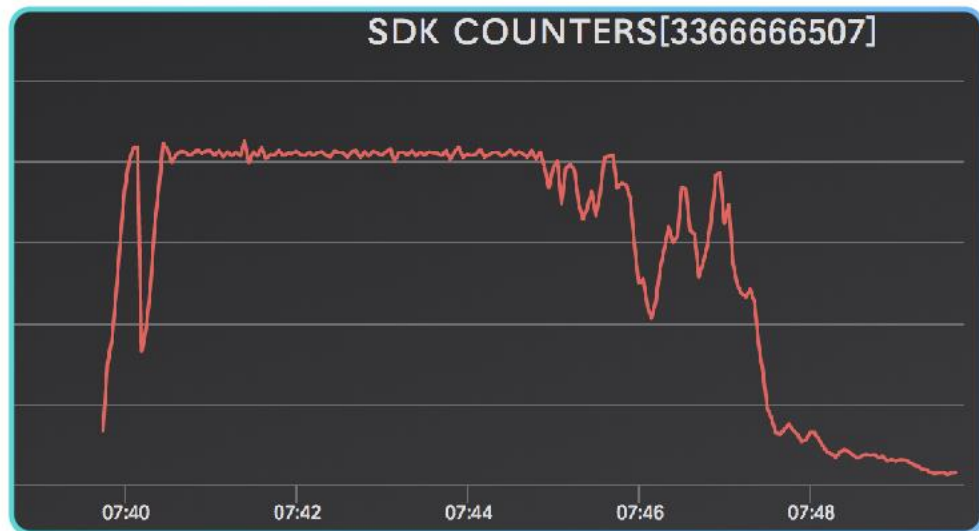


Qos那些事

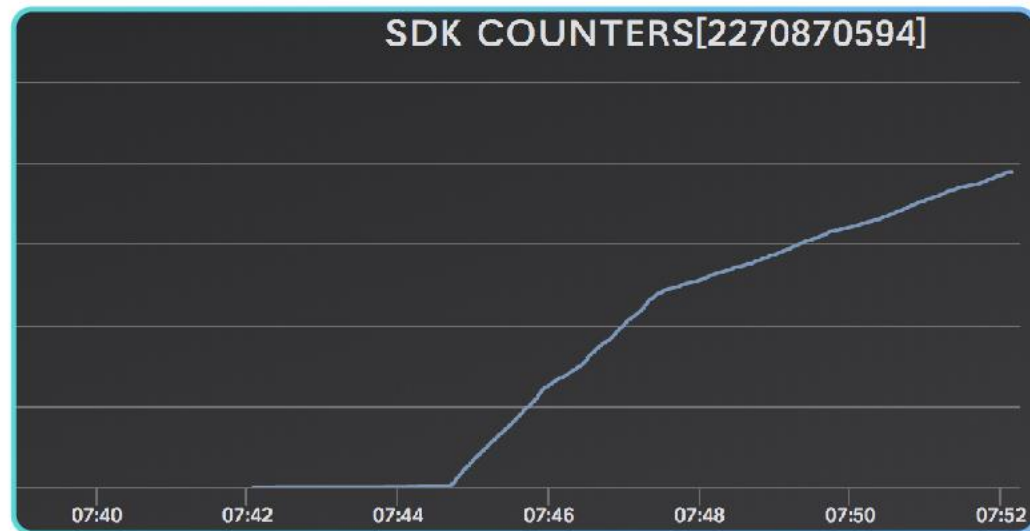


基于机器学习的网络模型预测

- 利用循环神经网络(RNN)学习媒体后台数据
- BWE/Bitrate/Jitter/Loss/Delay
- 使用训练模型预测当前网络，以决定网络对抗丢包策略



Sender bitrate



Receiver packet loss

基于机器学习的 网络模型预测



TensorFlow

```
num_features = 7 # num of rtc network features
timesteps = 15 # timesteps, 15 secs / sampling
num_hidden = 128 # hidden layer num of features
num_classes = 2 # lossy

X = tf.placeholder("float", [None, timesteps, num_features])
Y = tf.placeholder("float", [None, num_classes])

weights = {
    'out': tf.Variable(tf.random_normal([num_hidden, num_classes]))
}
biases = {
    'out': tf.Variable(tf.random_normal([num_classes]))
}

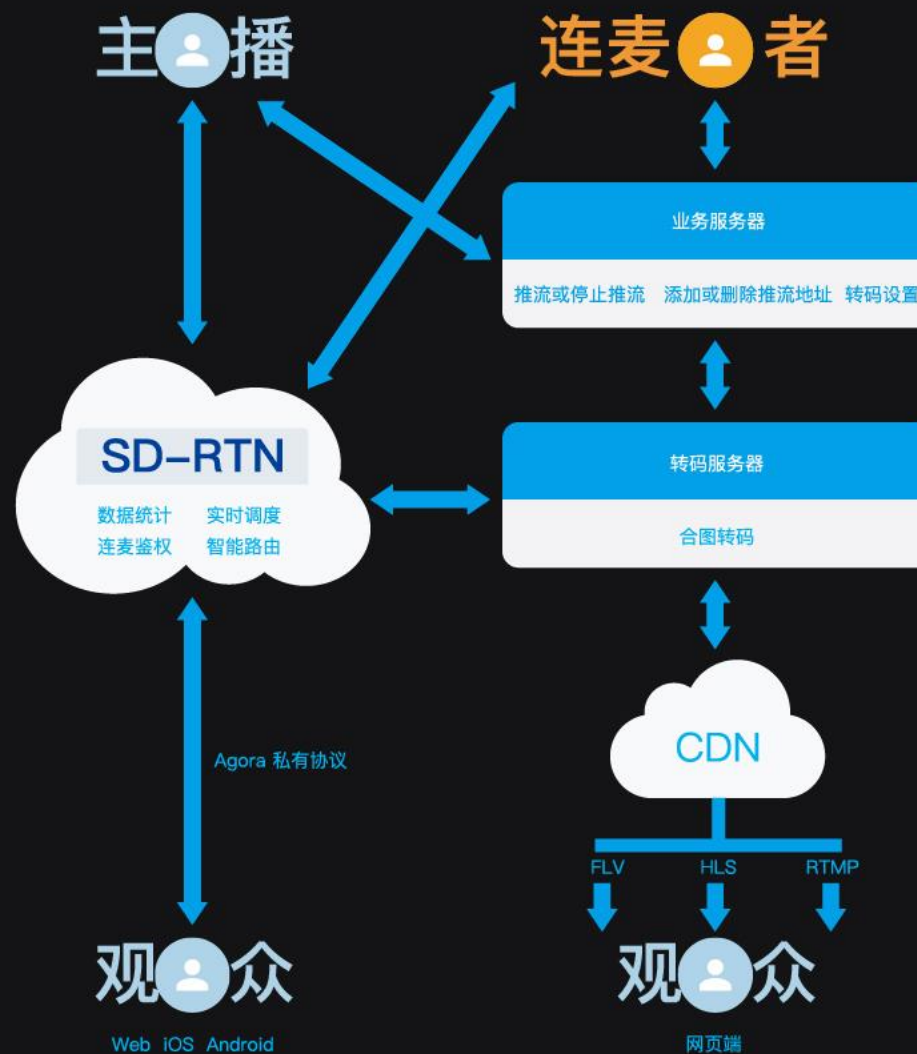
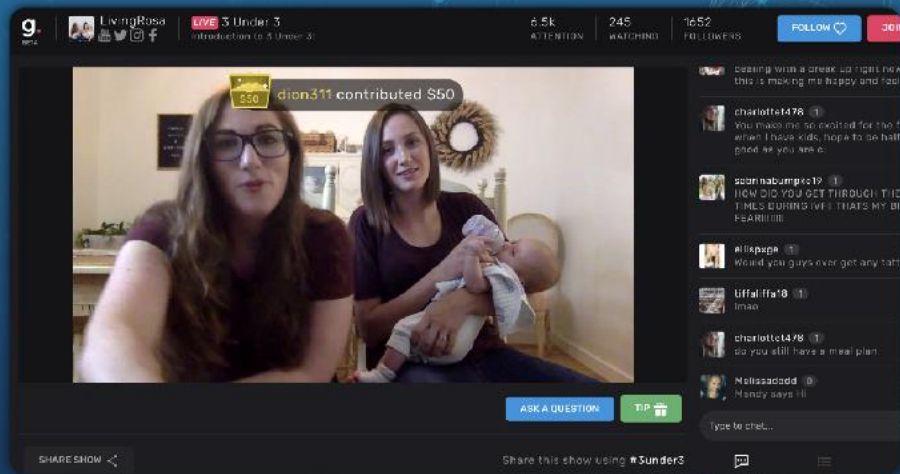
def RNN(x, weights, biases):
    x = tf.unstack(x, timesteps, 1)
    lstm_cell = rnn.BasicLSTMCell(num_hidden, forget_bias=1.0)
    outputs, states = rnn.static_rnn(lstm_cell, x, dtype=tf.float32)
    return tf.matmul(outputs[-1], weights['out']) + biases['out']

logits = RNN(X, weights, biases)
prediction = tf.nn.softmax(logits)

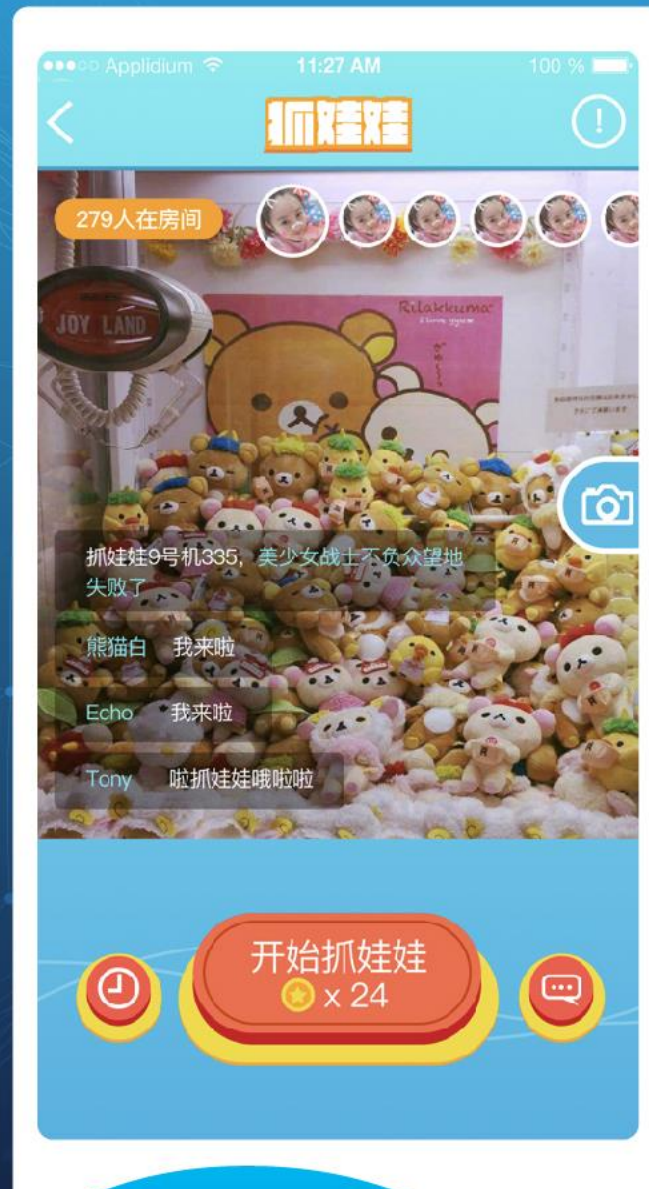
loss_op = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(
    logits=logits, labels=Y))
optimizer = tf.train.GradientDescentOptimizer(learning_rate=learning_rate)
train_op = optimizer.minimize(loss_op)

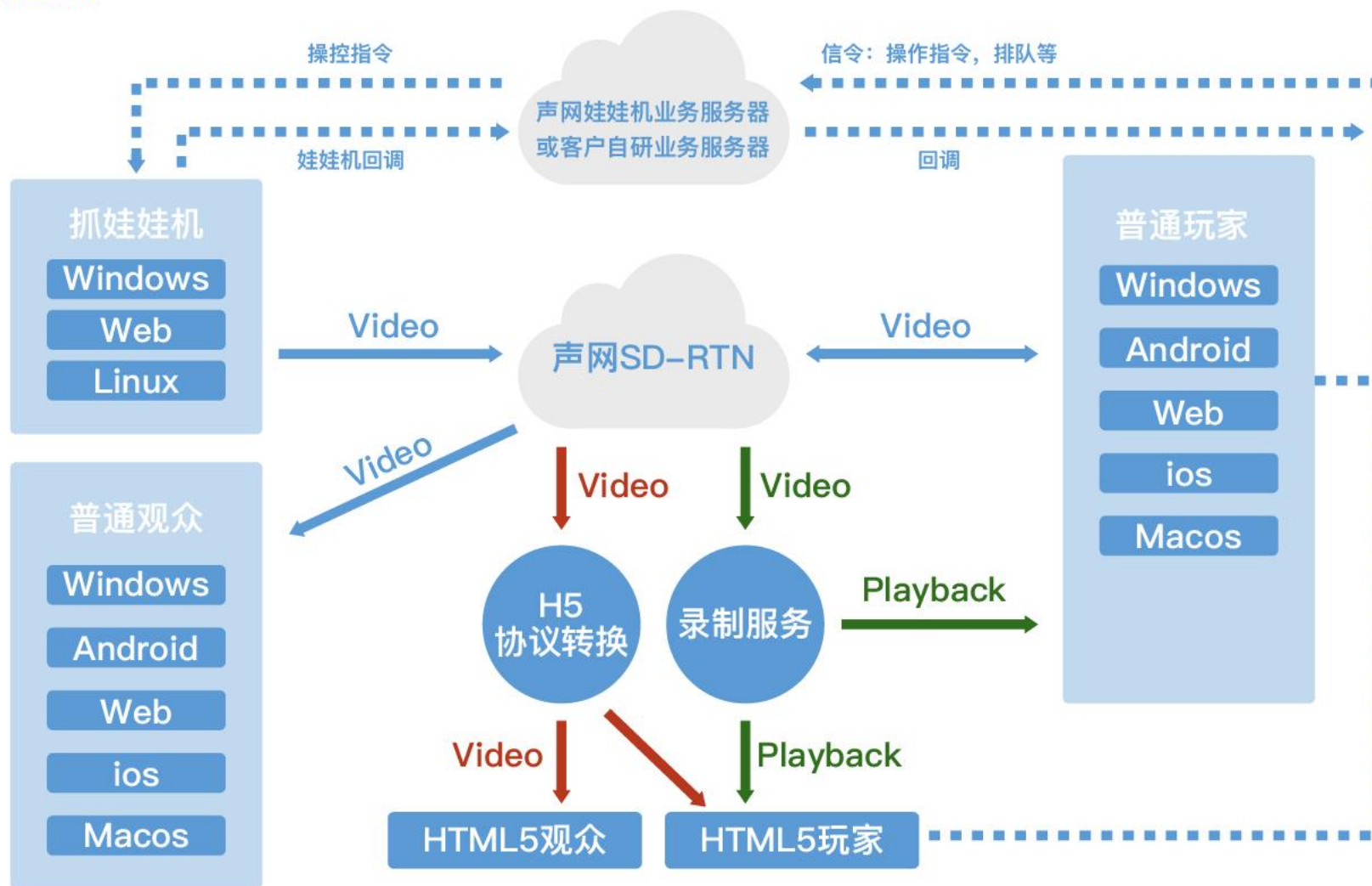
correct_pred = tf.equal(tf.argmax(prediction, 1), tf.argmax(Y, 1))
accuracy = tf.reduce_mean(tf.cast(correct_pred, tf.float32))
```

典型场景 — 互动直播



典型场景 娃娃机





感谢您的聆听

陈功

声网Agora WebRTC 架构师

