

# **Streaming Media in Hostile Networks**

Dafydd Harries

Collabora Ltd.

questions

background: media streaming

audio/video

in the context of a VoIP call

generally uses RTP

on top of UDP

IP addresses exchanged via signalling  
protocol

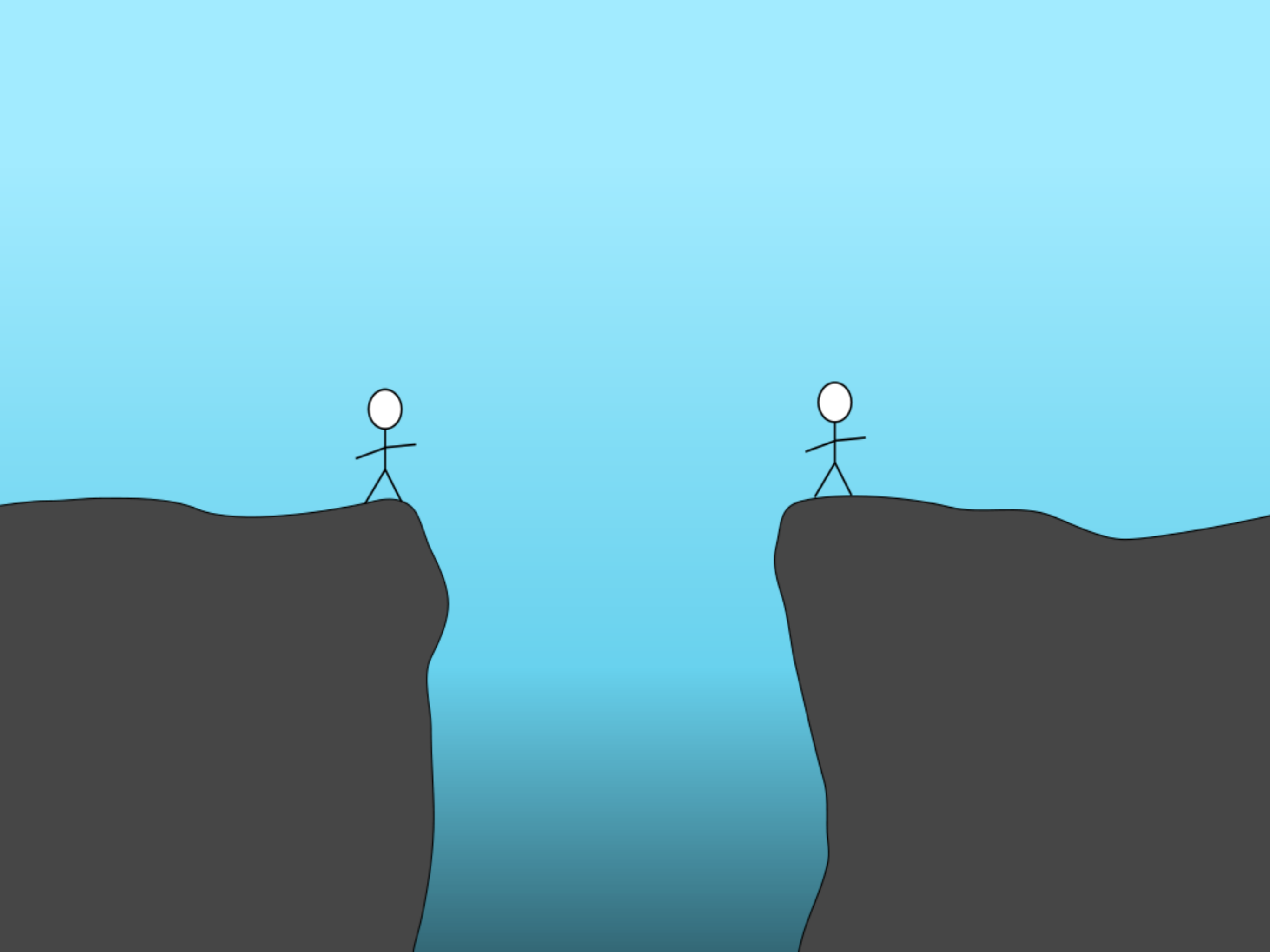
e.g. SIP

H.323

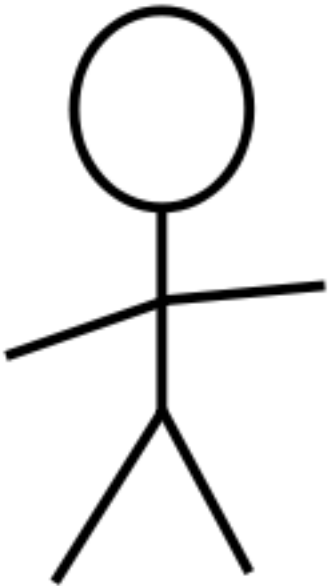
**XMPP (Jingle)**

games too

just exchanging IP addresses doesn't work



Alice



Computer

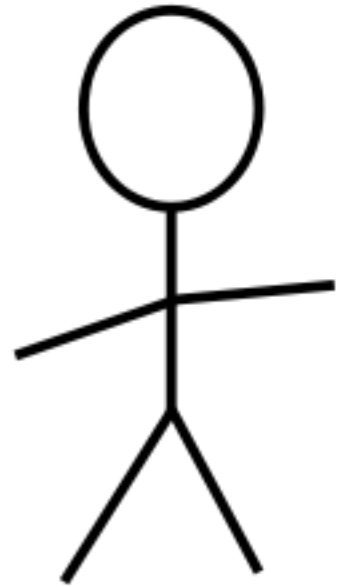


Network

Computer



Wendy

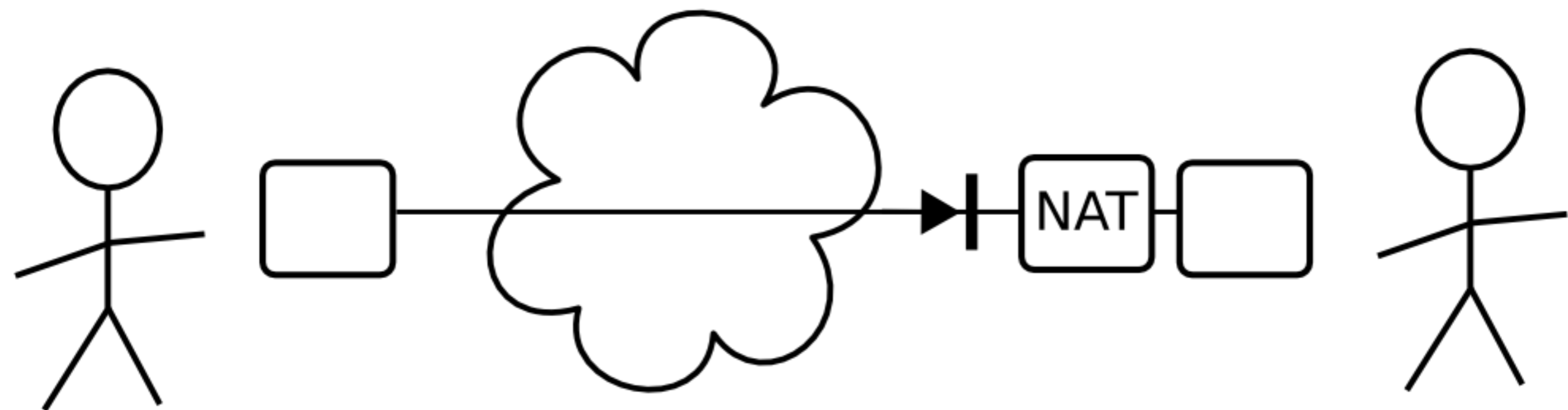


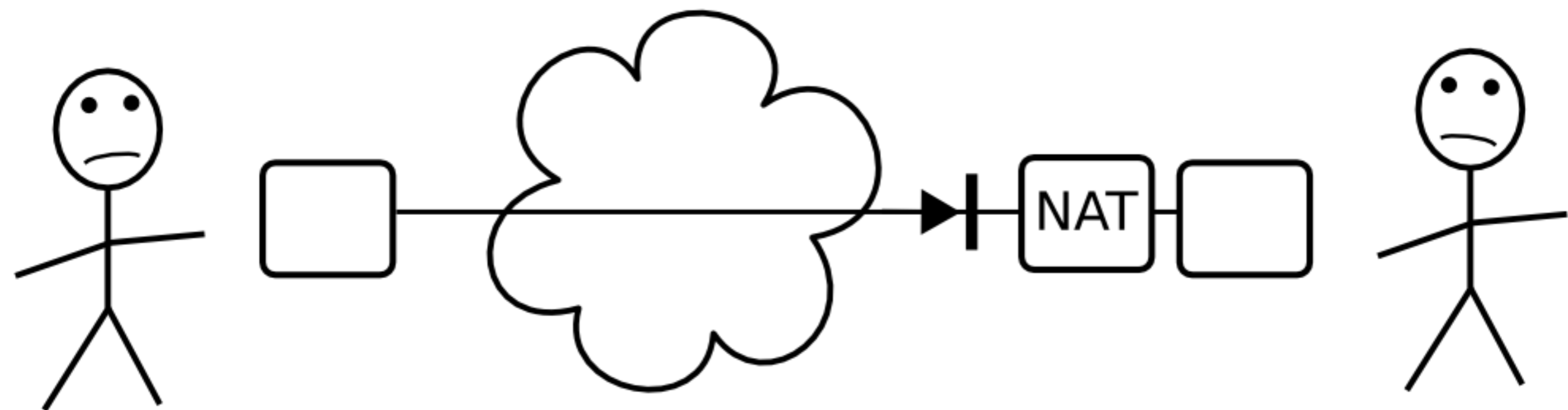
the problem

network address translation (NAT)

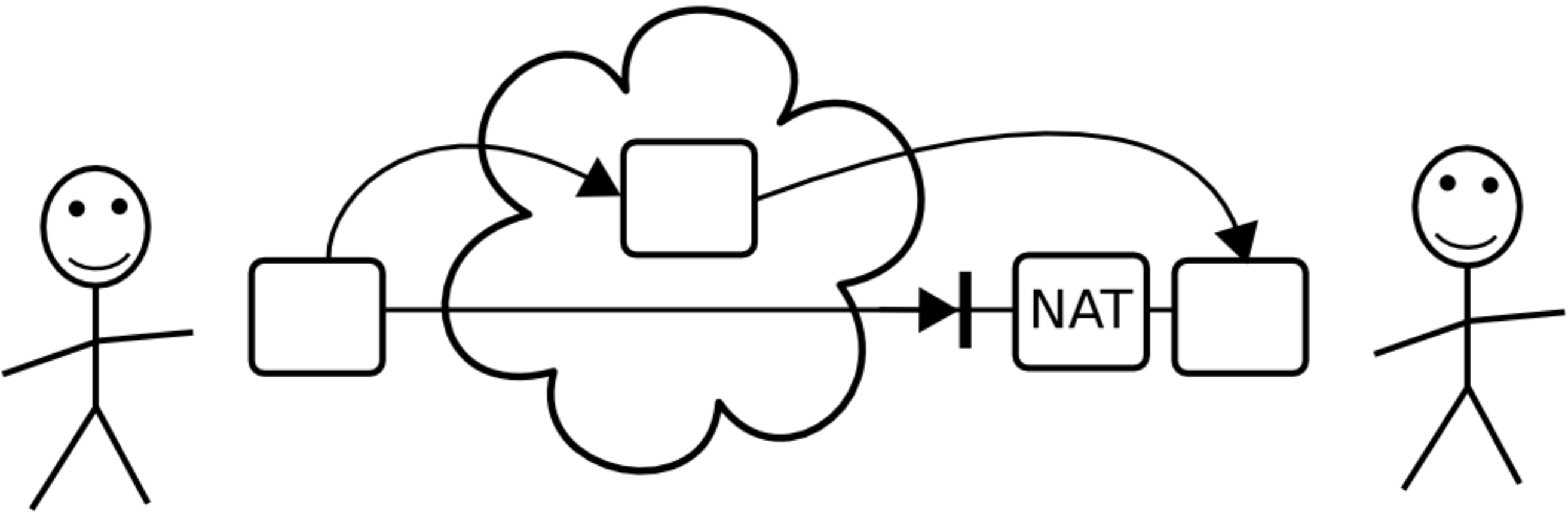
# IPv4 address space limitations

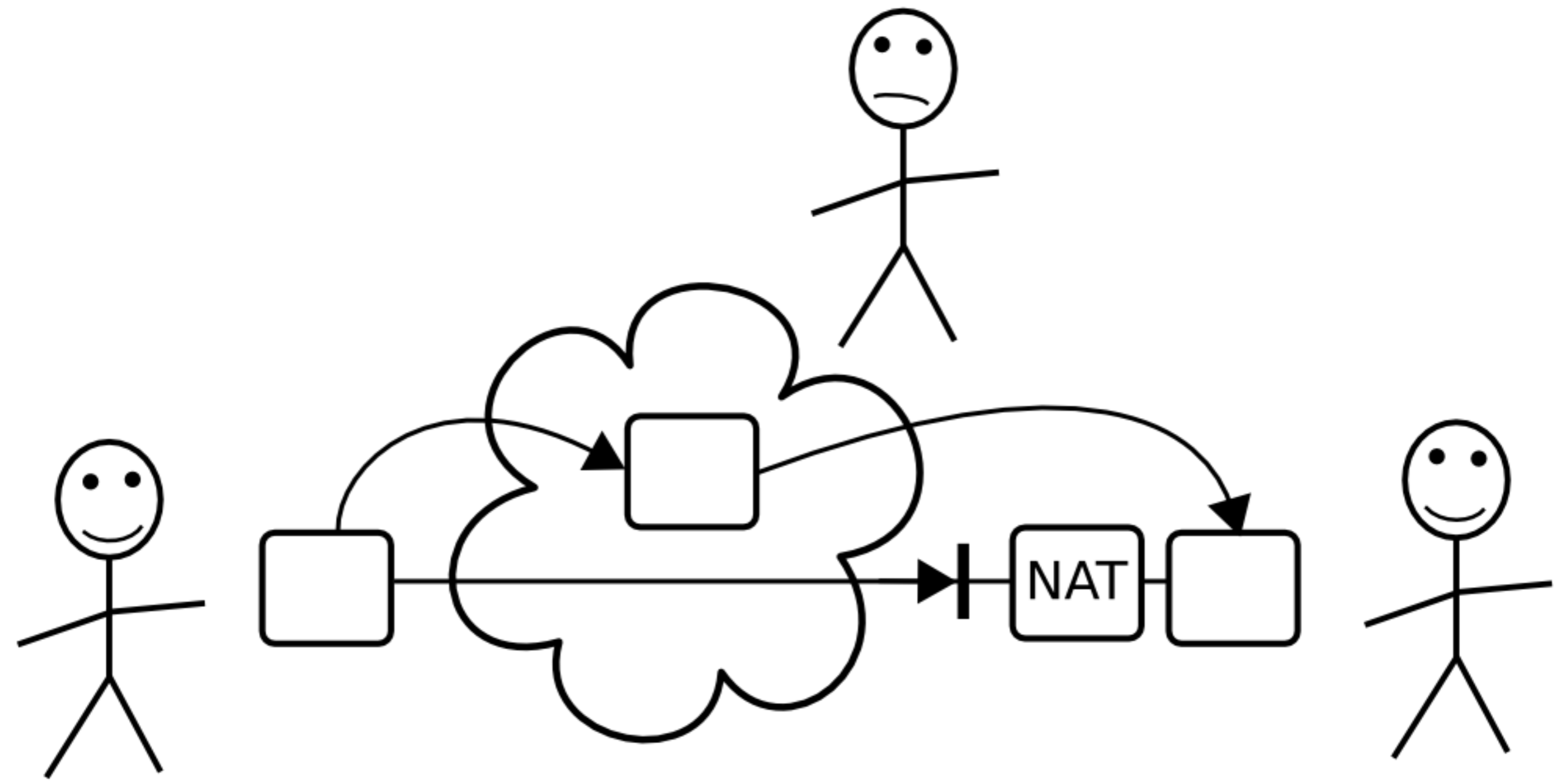
side-effect of blocking inbound traffic





you can relay media...





relaying is expensive

try to have media go peer-to-peer

many ad-hoc solutions

discover external IP

**STUN**

ask NAT for forwarding

**UPnP IGD**

**NAT-PMP**

Teredo

# MiddleBox Control Protocol

Realm Specific IP

# Interactive Connectivity Establishment (ICE)

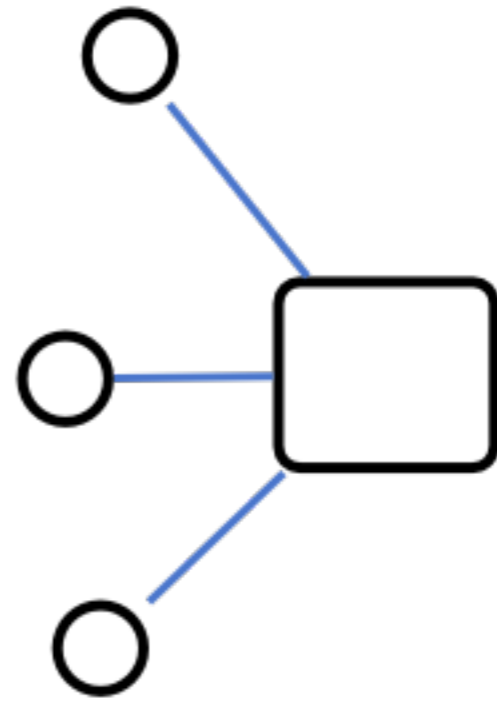
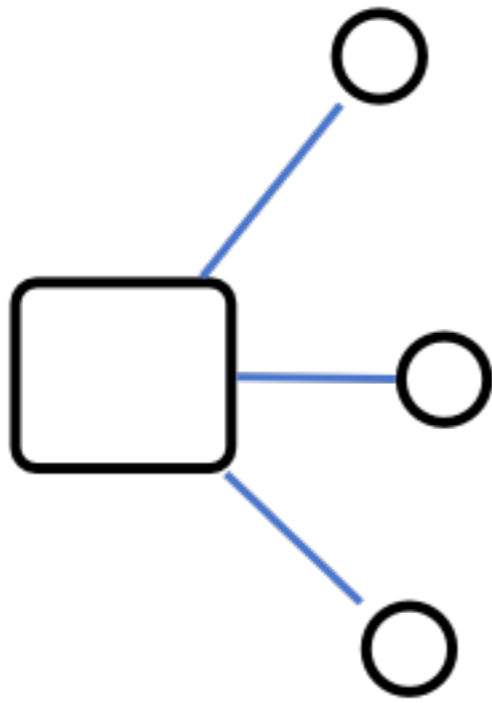
emerging IETF standard

methodology more than protocol

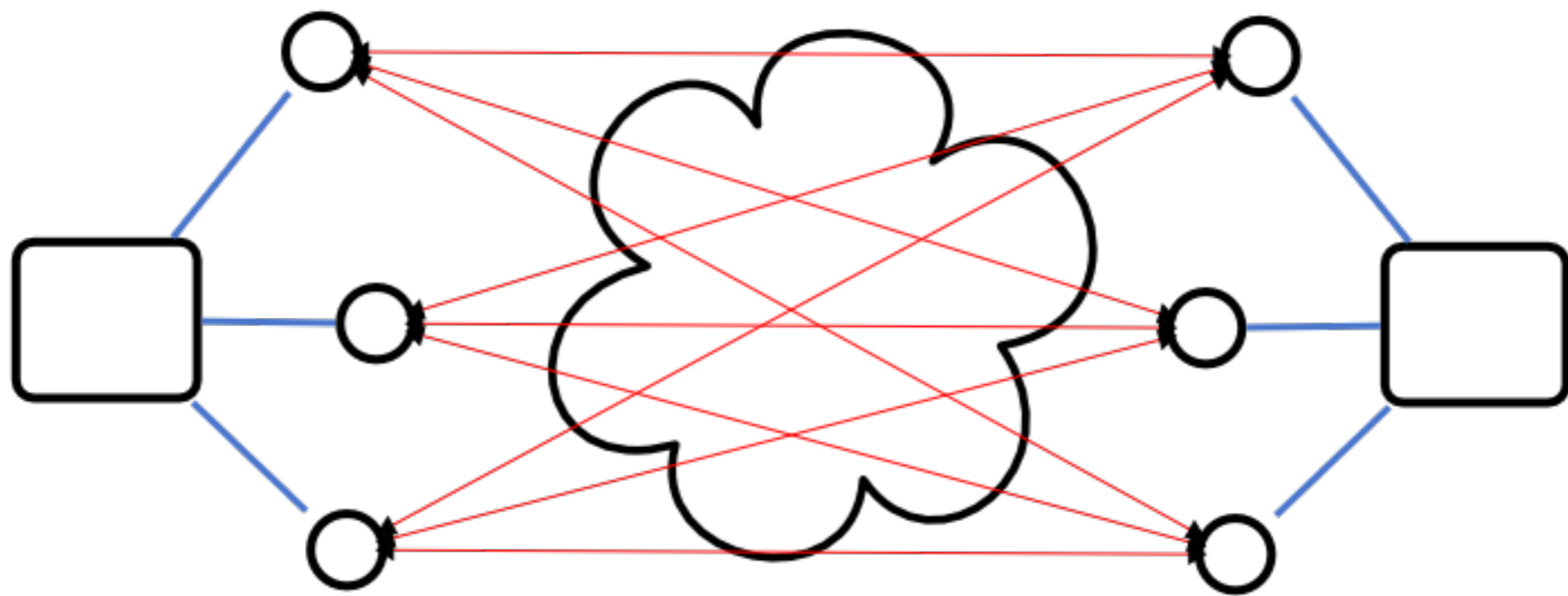
allows combining multiple techniques

in a nushell:

gather candidates (IP addresses)



test each pair of candidates  
until you find one that works



use STUN for connectivity checks

use UPnP IGD or NAT PMP to obtain extra candidates

try to finish quickly

but don't go too fast

keep checking in case something breaks

deployed (sort of) in Google Talk

> 90% NAT traversal rate

fall back to relay

libnice

free software implementation

goal: implement ICE standard

goal: make reliable streaming easy for  
application

goal: keep compatibility with Google Talk

goal: replace libjingle

**LGPL**

# GStreamer integration

# GLib mainloop integration

compatible with Google Talk

still to come:

# Farsight integration

full protocol support

Google relay support

conclusions

NAT makes peer-to-peer streaming difficult

ICE replaces ad-hoc solutions

**bring on IPv6**

questions