Communication Breakdown

making online conferencing work again in complex and highly secured company networks

Robert Hess
robert.hess@goto.com

Intro

- MacOS Sequoia severely tampers with Network interception software

Our industry is not well prepared to run reliable communication over the Internet in corporate networks

Neither security vendors

Nor communication vendors

And definitely not the customers

Let me introduce myself

Robert Hess

On a quest to help customers to fix their network communication issues

Working @goto – the makers of GoTomeeting

We will look at the problem from the perspective of a troubleshooter – requirements, problems, solutions

robert.hess@goto.com

What will I talk about today

- Requirements
  - Communication Software
  - Security
- Why does it conflict
- The great schism
- How to analyze
  - Things to look at
  - Practical examples
  - The test setup
- Will we all die

The Problem Space

Network requirements of communication software

- Access a lot of domains, vast IP and port ranges, sometimes dynamic
- Find best route (the one intended by the network admin)
- Avoid Deep Packet Inspection
- Handle high traffic in aggregation points (VPN Gateways, Proxies)
- Works world wide
- High bandwidth/low delay

Google Quote

Network best practices

Configuring default video quality

Using Wi-Fi

Using VDI

Avoid using Proxies

Avoid using QoS

Avoid using VPNs

Network Security Requirements

- Central management and inspection of all "unknown" traffic
  - Avoid download and execution of malware
  - Prevent access to malicious content
  - Intercept trojans calling back home
- Centralized management of endpoint traffic
- Office and remote work
- Regulatory requirements for specific industries

Network security practices

- Outsourcing of network security to SaaS providers
- Locked down devices as they are mobile, and the network is no longer assumed benevolent
- Multilayer approach
  - Agent on the device to inspect code before download/execution
  - Agent on the device to decide what happens to outbound network traffic
  - Traffic with no special rules is send to the cloud for DPI
  - Traffic that needs to access protected company resource (internal and SaaS) is send to the cloud to originate from specific IPs which are allowlisted on the company side
  - Ideally: high volume traffic should be broken out locally (split tunnel) to avoid high traffic concentration – that's not always possible.
    - Regulatory constraints
    - Bad access patterns of application make it hard to define such rules

Conflict

- Most secure solution is no communication
- Call for more security overrules communication needs
- Different departments with different agendas
- Centralized control vs. decentralized traffic
- Implications of security measures often not well understood

- Central monitoring of communication issues is tricky – have to rely on user reports
- Burden of proof is moved to the communications vendor

Statement

- Industrie is not well prepared
- Neither security vendors nor communication vendors, nor customers
  - Problem has escalated over the last 5 years
  - Most companies are made from bits and pieces, nobody knows network topology any longer, IT got outsourced
  - Communication companies have little deep network know-how
  - Security companies have odd ideas what they can do to the Internet
  - Changing grown structures and products takes very long

The great Schism

Communication ←→ Security

Central vs. Decentral

- Security calls for centralization of control, and as a result also of traffic.
  - This creates bottlenecks which are ill-suited for high bandwidth, low latency streams.
  - And we are not even talking about P2P here
  - Its a general problem in the world of SaaS services.
- One size fits all solution makes it hard to adapt to different needs, say developers vs. sales

Complexity

Communications solutions are inherently complex for a number of reasons:

- Phone legacy
- Systems cobbled together from unrelated components
- Lots of workarounds for the Internet 15 years ago are still in place
- We just love complexity

Combine this with a security solution, which is riddled with all kinds of workarounds to accommodate the quirks of real-world software, you have a recipe for failure.

→*Nobody (no single person) any longer really understands what's going on*.

Too many eggs in a single basket

- This type of complexity can only be juggled for a few products on the user side, and by a few large companies on the vendor side.
- I see a consolidation on a very few names, when it comes to communication, as well as security.

→This has its risks.

Will the sky fall down?

- Probably not

How do we deal with it?

What can we analyze

Terms

- Proxy
  - https
  - L7
  - DPI
  - DNS/SNI based filtering
- Packet filter
  - IP/Port
  - Protocol
  - L3 / 4
- VPN/Filter Agent
  - L3 -L7
  - combines inspection and filtering on all levels

Interception points

- local OS
  - Packet filter
  - local proxy config (PAC) – what gets send where
  - VPN/Filter Agent
- Company network
  - Proxy
  - Packet filter
- Cloud:
  Filtering on all layers, incl. heuristics and AI based rules

Policy for blocking or allowing based on

- DNS/SNI
- IP, Port
- Apps
- Protocols

How to analyse

- Capture either with Wireshark or CLI or with VPN Vendors built in capture
- Find the right place for your capture – in case of doubt capture on all interfaces and different places in the network

- Build yourself a test setup for analysing these things
  - Export TLS keys
  - Use Packet filter, proxy and DNS to reproduce problems

- Analyse with Wireshark

Telltale signs

- Blocked connections
  - No UDP connection possible
  - Proxy used instead of direct connection (check PAC file)
  - Direct or proxied TCP connections fail (check allowlists)
  - Proxy Auth fails
- Packet retransmission
  - TCP – retransmission and DUP ACKs
  - UDP - check RTCP or the respective RTP payload types for RTX
- Unintended routing
  - Use of VPN instead of split tunnel
  - SDWAN selecting wrong connections
- Change of Certs (different issuer)
  - Indication that connection undergoes DPI

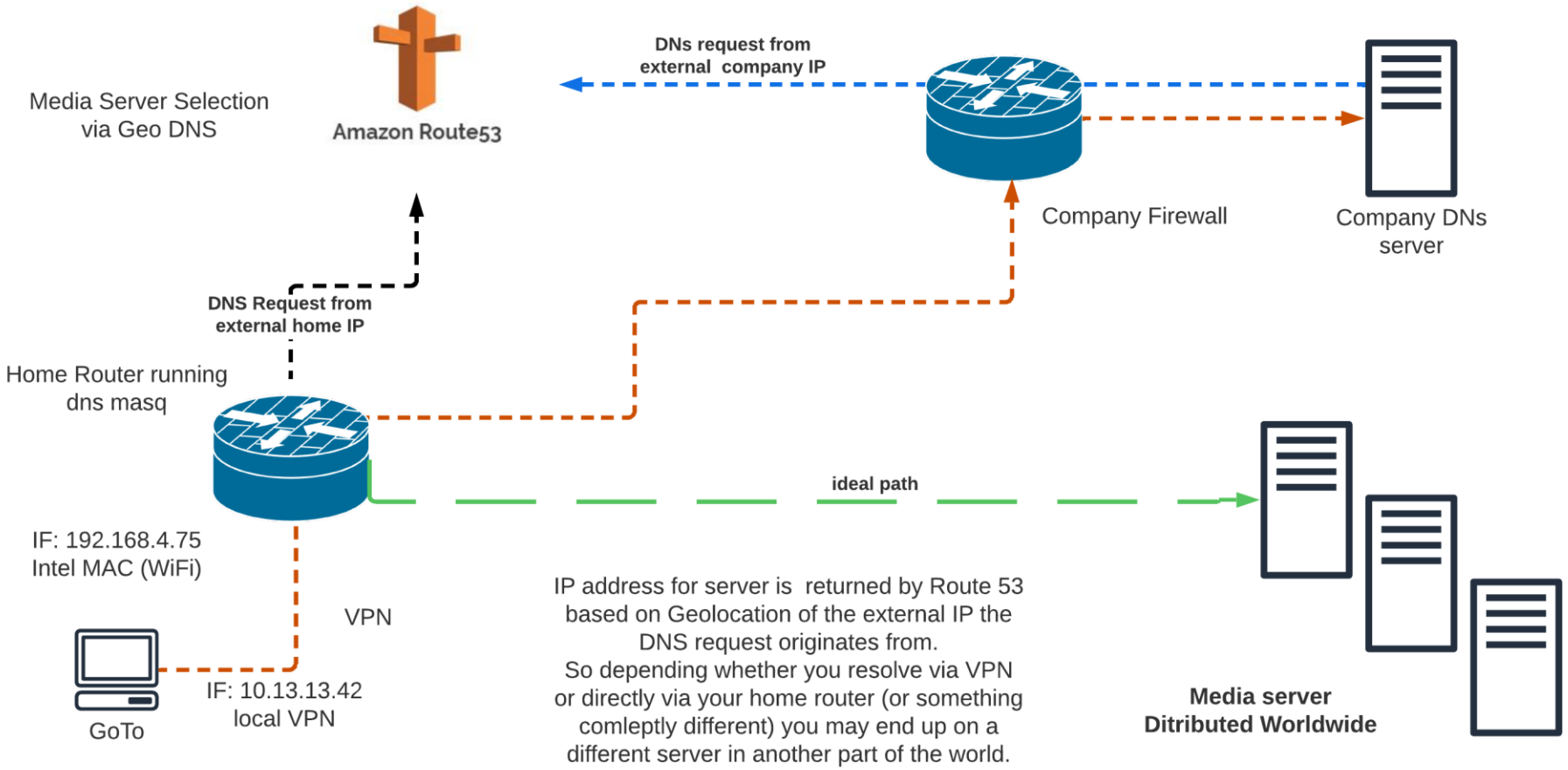
Telltale signs #2

- ICMP/Fragmentation issues
  - Video uses large packets – blocking ICMP "needs fragmentation" and restricting MTU size leads to black videos
- QoS
  - Google Quote: Best practice is not to use quality of service (QoS) for Meet in your network.
  - Exception: WiFi with heavy traffic
- DNS Issues
  - DNS timeouts because no external DNS
  - Wrong GeoDNS results because DNS request exits company network in the wrong region → call ends up on servers on another continent
  - Some security systems use DNS requests to learn which IPs are allowed → this can fail in so many ways

Real-world: No external DNS on clients behind proxy

- Assumption: All external traffic has to go through proxy, hence its sufficient to resolve internal DNS names, as the proxy will take care of the external resolution
- Reality: WebRTC tries a DNS resolution for the TURN server before connecting the proxy.
- Why: WebRTC running in the browser is not aware that a proxy is configured and hence tries to resolve the name for a direct connection.
- Problem:
  - Connection setup comes with a configurable timeout – you do not want to wait endlessly until you retry.
  - DNS timeout depends on a lot of factors, local resolver, DNS server config
  - → there are cases where it takes to long (i.e. 20 or 30 seconds) for DNS to return the failure – meanwhile the connection setup times out
  - Retry just runs into the same issue
- Solutions:
  - fake DNS entry
  - do not configure TURN because you already run traffic over a single port and your servers support TCP

GeoDNS challenge



Real-world: WebRTC using unintended Interface

- Remote workplace with Anyconnect Split tunnel
- Assumption: a route was used that the communication server should be broken out locally.
- Reality: WebRTC decided to send them over the VPN Link lading to higher delay as the topology was not optimized for this
- Problem:
  - WebRTC binds to all interfaces and sends packets, regardless of the routing. The ICE process found working connection over VPN as well as directly over the local Interface
  - When a connection can be made, the interface with the lowest interface metric is used.
  - Anyconnect uses metric 2 or 1 for its interface – ethernet runs at 10
- Solution:
  - Explicitly block STUN traffic over the VPN, so that no connection can be detected

Smoke and mirrors

- Tunneled DNS (DoH, "secure" browsing) → GeoLocation issues
- DNS filtering (Blackholing, i.e. PiHole) → functionality not fully available
- DNS rewritten by VPN agent
- Tunneled traffic adding overhead and delay

Wireshark Setup

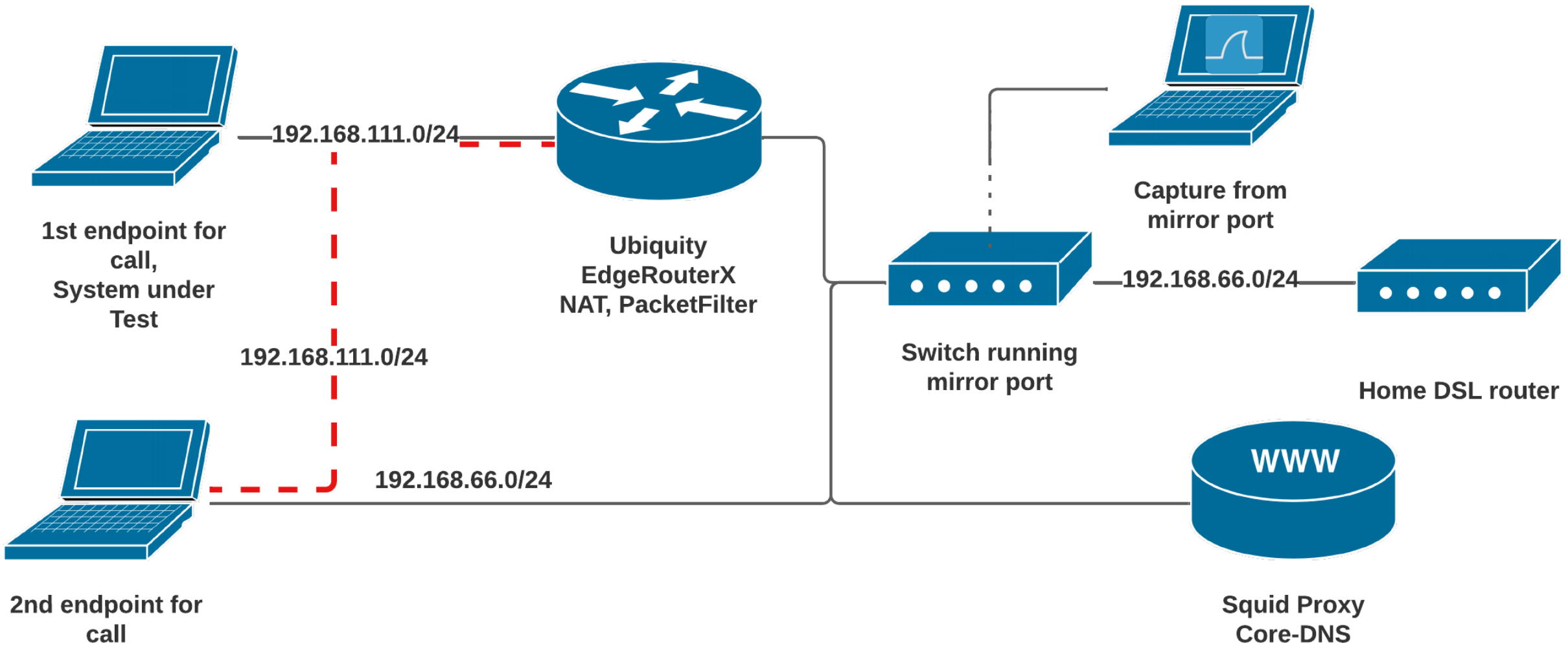
- Filter shortcuts
  - dns
  - stun
  - Proxy CONNECT – filter for proxy connect request
  - DeNoise – remove noise like arp, ssdp, llmnr – depends on your network
  - TLS Client Hello – Contains SNI
- Columns
  - T since prev frame in stream – look for timing issues
  - Server name – is SNI set
- GeoIP setup
  - Import Maxmind DB – check whether the servers used are close by
- Capture filter or header only captures
  - Only for really longterm captures or in extremely noisy networks
  - You may miss something unexpected

Test Environment

Gives you time to verify hypotheses and potential fixes

- client computers/phones with and without security setup
- use port mirror on mini switch
- external packet filter
  - the one on the OS is usually more complex to understand where exactly it intercepts the network stack
- Proxy server
  - With and w/o auth
  - DNS based allow lists
  - Proxy log gives indication what is really accessed
- DNS server
  - Test fake DNS entries
  - /etc/hosts is not always consulted depending OS/App behavior

My test setup



Capture on Mobile

You can always capture WiFi traffic from a mobile by hooking up your own AP to a switch with port mirroring and capture on the mirror port

If you need to see cell traffic, or the switch between traffic types

- iOS: Attach Mac and use the remote capture feature (see appendix)
- Android: capture directly on the device via an app (creates a VPN interface to intercept traffic)
- Specifics
  - High likelihood of connection interruptions due to cell handover or cell-wifi switches
  - Most likely IPv6 these days – provider does NAT64 and DNS64 if necessary

Outlook

Outlook

What can vendors fix?

- Smaller IP/Port/Domain Footprint
  - Exemplary: Google Meet
- Avoid dynamic addresses/host-names
- Minimize connection setup (e.g. ICE Lite) to reduce complexity
- Find better solution for determining locality than GeoDNS
- Avoid client-side redirects (http, Load balancer)

How can we mitigate impact

- Spread the word – educate folks about these topics
- Avoid "security" hacks in your network.
- Last Resort: run communication over separate access networks and devices – e.g. mobile. Screensharing stays on the computer as the requirements are way lower than for A/V
- Deploy everything in waves using canary deployments were possible
- Have early adapter groups
- Have test setups for the typical scenarios

Opportunities

- Working at the intersection of security and communication is a fascinating area of work for the future
- The various vendors can be relied on to create enough issues to analyse

- Things will just muddle along as they always have been

Thanks

https://conference.wireshark.org/sharkfest-24-eu/talk/TP88BZ/feedback

Appendix

iOs Capture - Method 1 using Airtool2 (slightly simpler)

There is a free test period, after which you have to pay something.
Generally, the capture process is a bit simpler, but it will capture stuff on all strange internal interfaces and the resulting capture is a bit cluttered. So unless you need to know which interface gets which packets, you are probably better off with method 2, which only uses free tools.
From these instructions

1. Connect iPhone with Mac - Unlock your phone - whenever prompted tell it to trust each other
2. Make sure Wifi is switched **off** on your iPhone if you wanna capture cellular traffic
3. On your Mac
    a. Install Xcode
    b. Install Airtool2
    c. click the AirTool2 Icon in your Macs status bar and select Capture iPhone Packet Trace (this is only available when your phone is connected to your Mac via cable)
    d. it will show a capture window
4. now start GoTo on your iPhone and try to join a session.
5. Once you see the problem you can stop the capture on the Mac. You should at least wait for like 2 Minutes to give it some time to try various reconnects.
6. the capture file is on your desktop (if Wireshark is installed it will open it automatically)

iOS Capture: Method 2 Using Wireshark

From these instructions

1. Connect iPhone with Mac - Unlock your phone - whenever prompted tell it to trust each other
2. Make sure Wifi is switched **off** on your iPhone if you want capture cellular traffic
3. On your Mac
    a. Install Xcode
    b. install Wireshark
    c. in Xcode open *Window → Devices and Simulators*
    d. you should now see your Iphone in Devices and Simulators - if not check that your phone is unlocked
    e. Copy the Identifier
    f. open terminal and run *rvictl -s [identifier]* → this will tell you about succeeded and on which interface (I have seen rvi0 and rvi1 so far)
    g. start Wireshark and have it capture on interface shown in the step before (Raw IP) - it might complain that it can do promiscuous mode -ignore
4. now start GoTo on your iPhone and try to join a session.
5. Once you see the problem you can stop the capture on the Mac. You should at least wait for like 2 Minutes to give it some time to try various reconnects.
6. In Wireshark, save the capture to a file.

Complications (there always are some)

if you use Wireshark (Method 2) to capture on rvi0 you get a pcap containing a raw IP capture without any further interface names. The god news however - this seems not to contain the stream between your iPhone and your Mac, which is useless anyways and just doubles the size of your capture file, which is always bad and adds the extra step to remove it from your capture.

If you use AirTool2 (Method 1) you get Interface ID on your packets. The bad news - there is no consistent definition for the naming in iOS as it seems + you get the stream between Phone and Mac which makes for some interesting comparing of IPv6 addresses to see whats what.

(This filter seems to exclude the connection between iPhone and Mac based on some guessing *!frame.interface_name == "anpi0"*)