Completing the DNS Service Discovery Architecture

Ted Lemon

What is DNS-SD

- A.K.A. Bonjour
- Automatic advertising and discovery of network services
  - Permissionless
  - Managed or unmanaged
  - Built on top of existing Domain Name Service
  - DNS without DNS-SD is also an advertising/discovery service; DNS-SD builds and improves on this platform

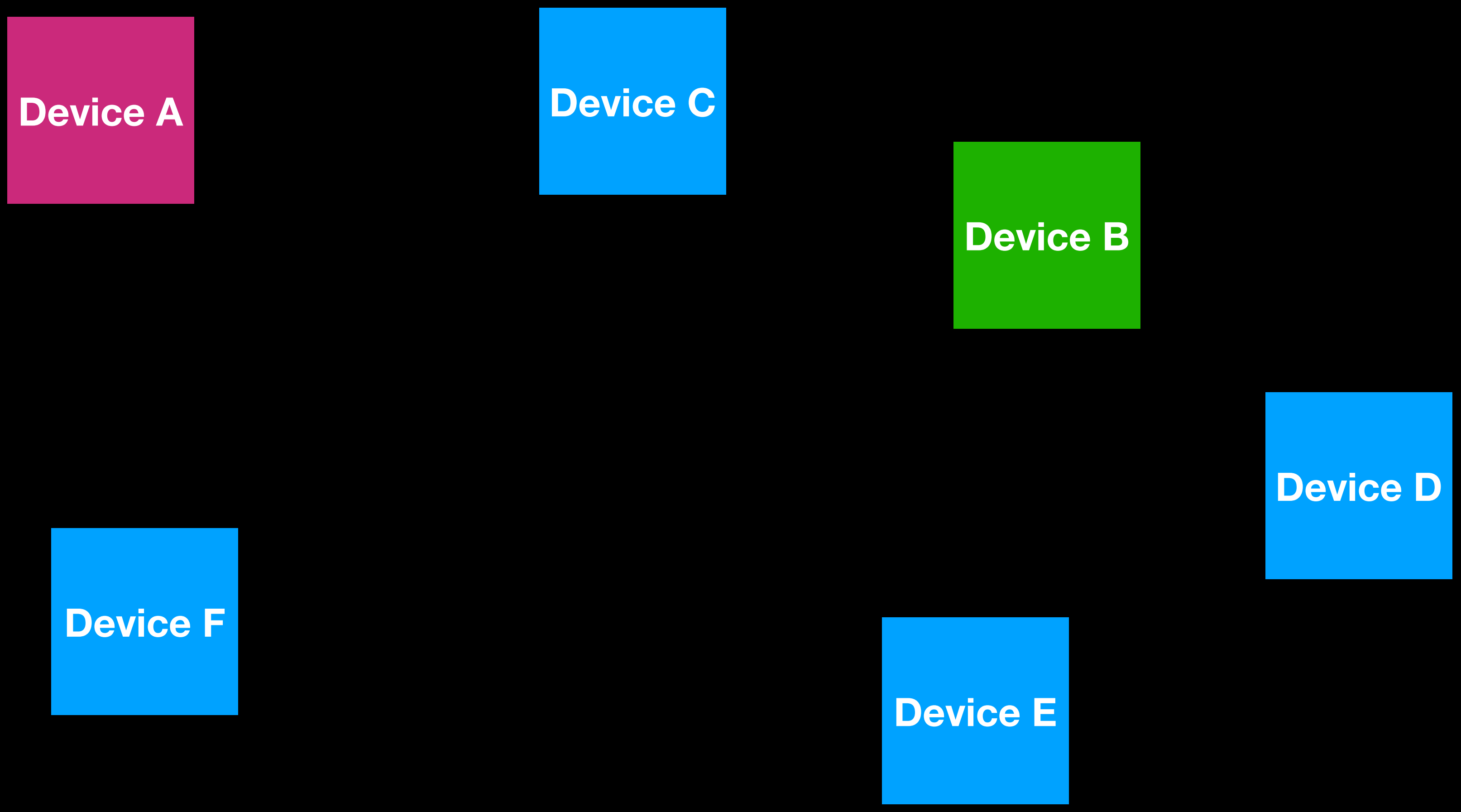
How it works

- Devices A, B, C provides a service
- Devices A, B, C advertise the availability of their services
  - e.g. `_hap._tcp.local IN PTR Device-A._hap._tcp.local`
- Devices A, B, C advertise info about their services
  - e.g., `Device-A._hap._tcp.local IN SRV 0 0 5432 Device-A.local`
  - e.g., `Device-A.local IN A 192.0.2.1`
- Device D needs that service
- Device D discovers servers:
  - `_hap._tcp.local IN PTR ?`
- Device D resolves a service:
  - `Device-A._hap._tcp.local IN SRV ?`
- Device D gets IP address of server:
  - `Device-A.local IN A ? IN AAAA ?`

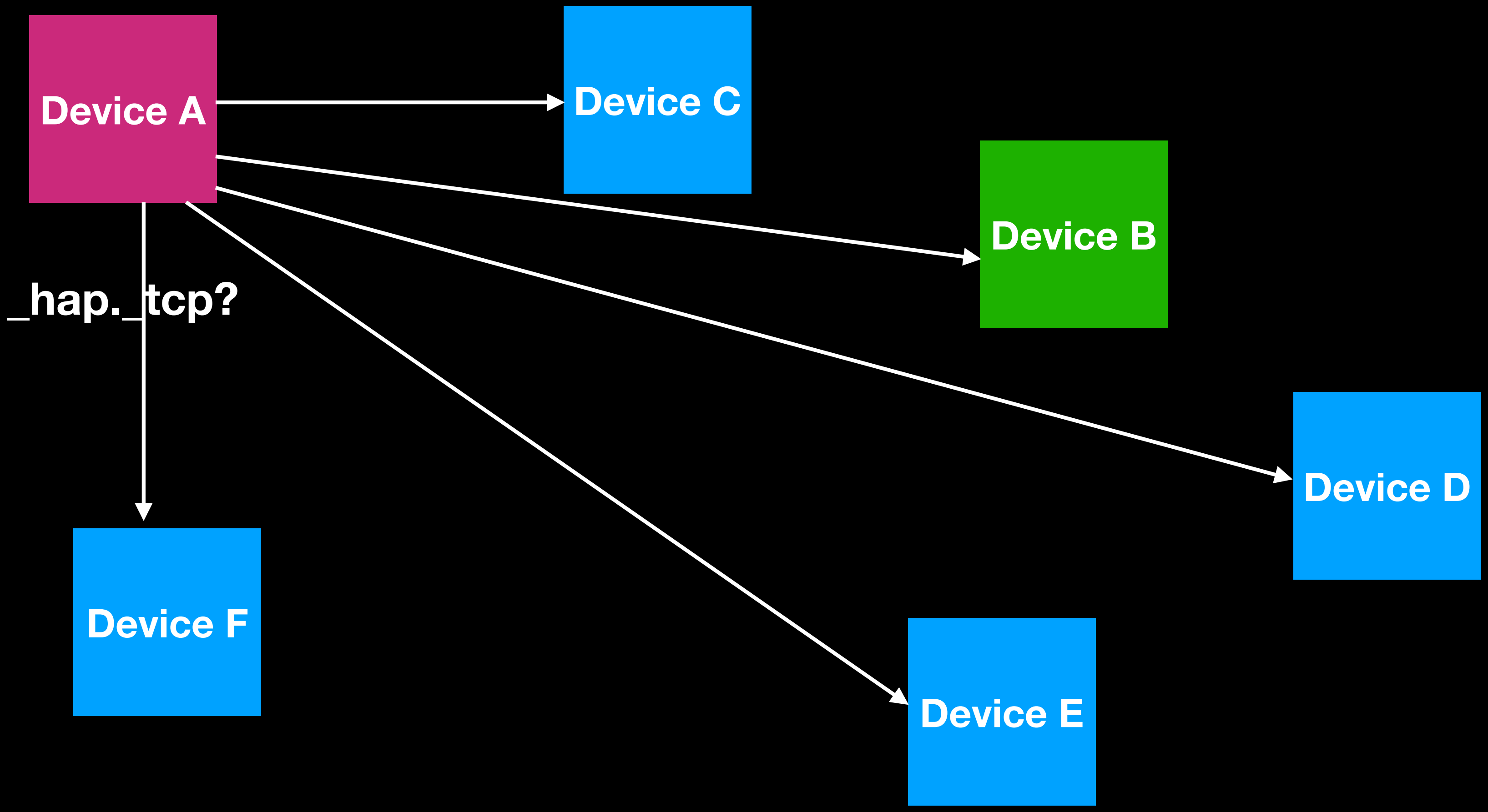
State of the art in 2018

- Two ways of doing DNS-SD:
    - Multicast DNS (mDNS)
        - Permissionless
        - Limited to a single subnet
        - RFC 6763 (DNSSD) + RFC 6762 (mDNS)
    - Unicast DNS (regular DNS)
        - Only supported for managed networks
        - Not automated
        - Not limited to single subnets
        - RFC 6763 (DNS-SD) + RFC 1034/1035 (DNS)

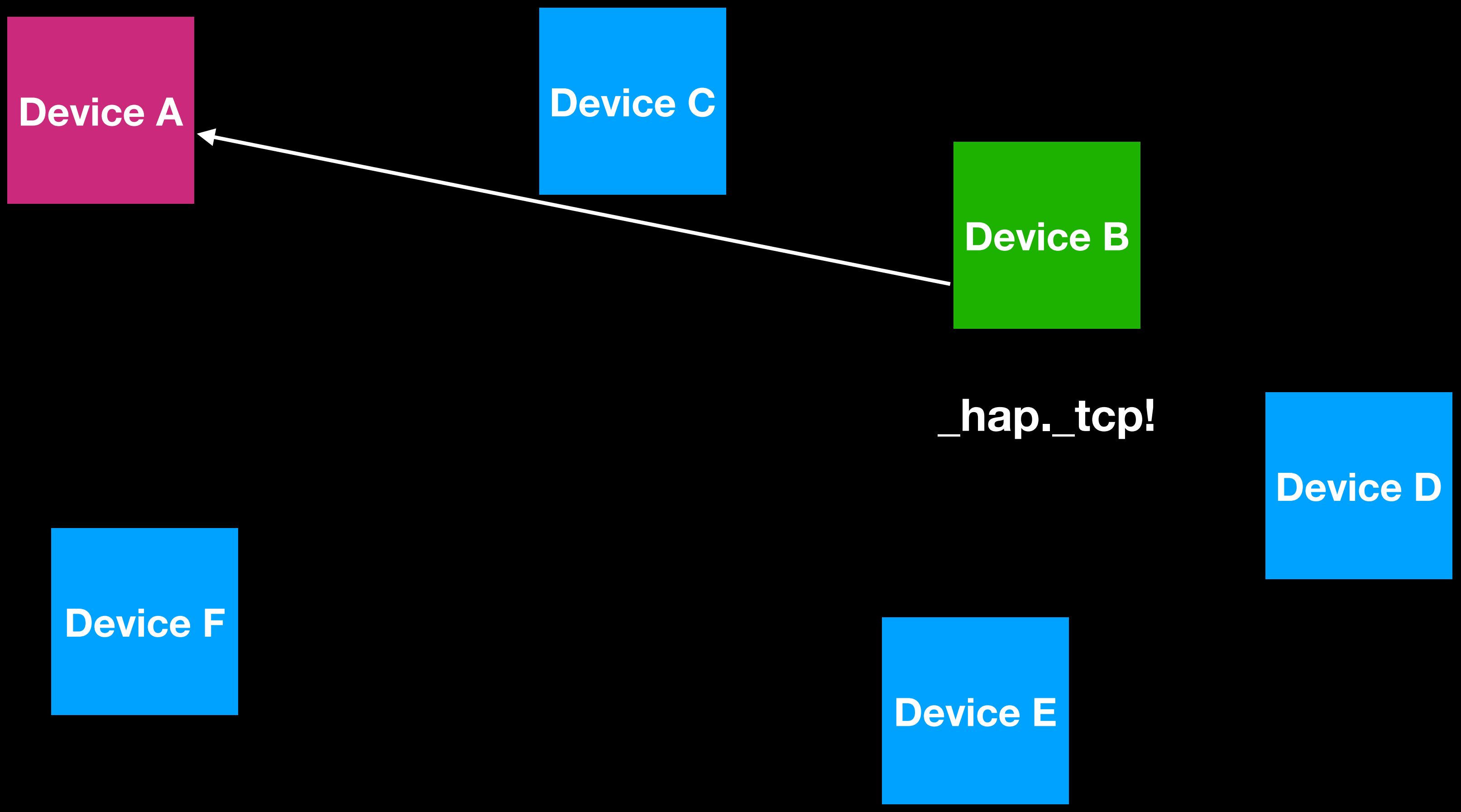
DNS-SD over mDNS



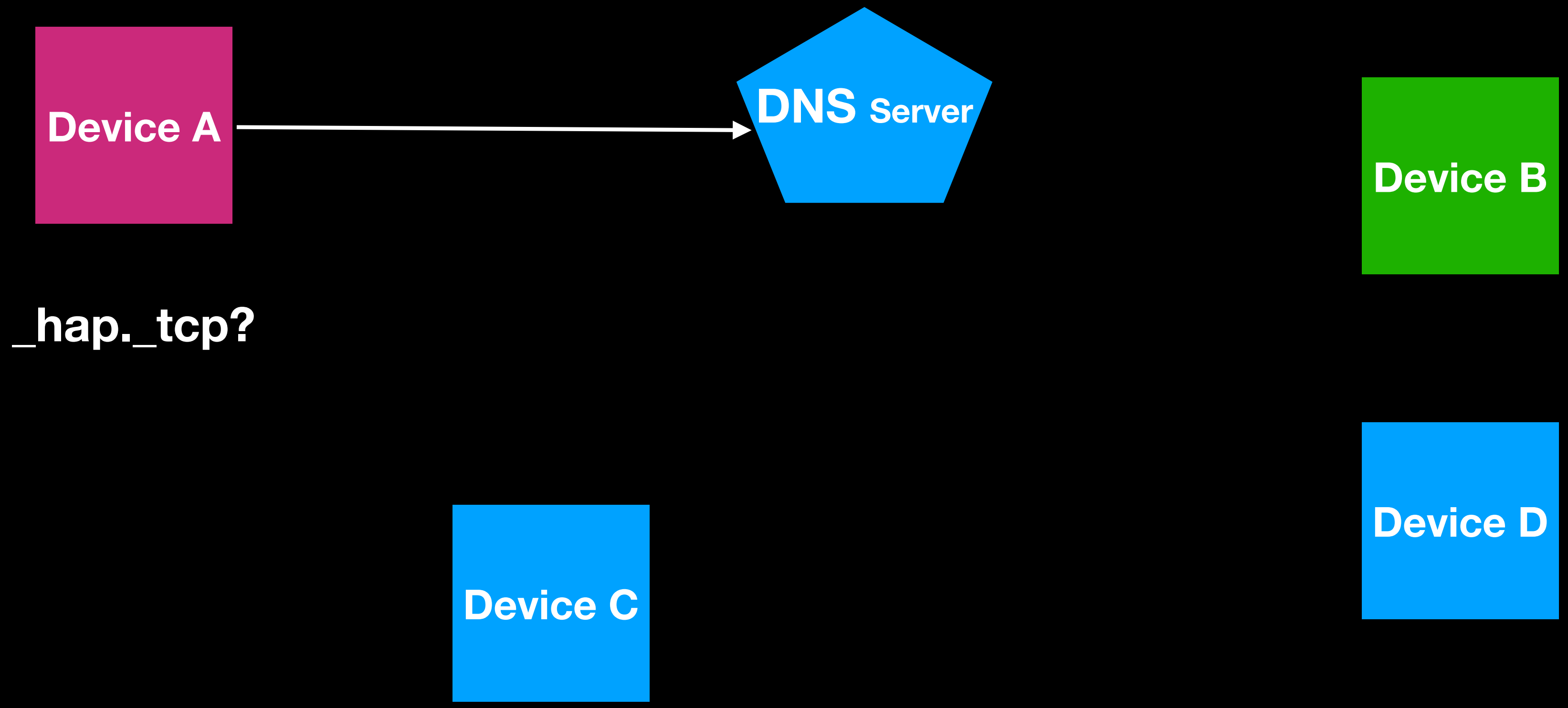
DNS-SD over mDNS



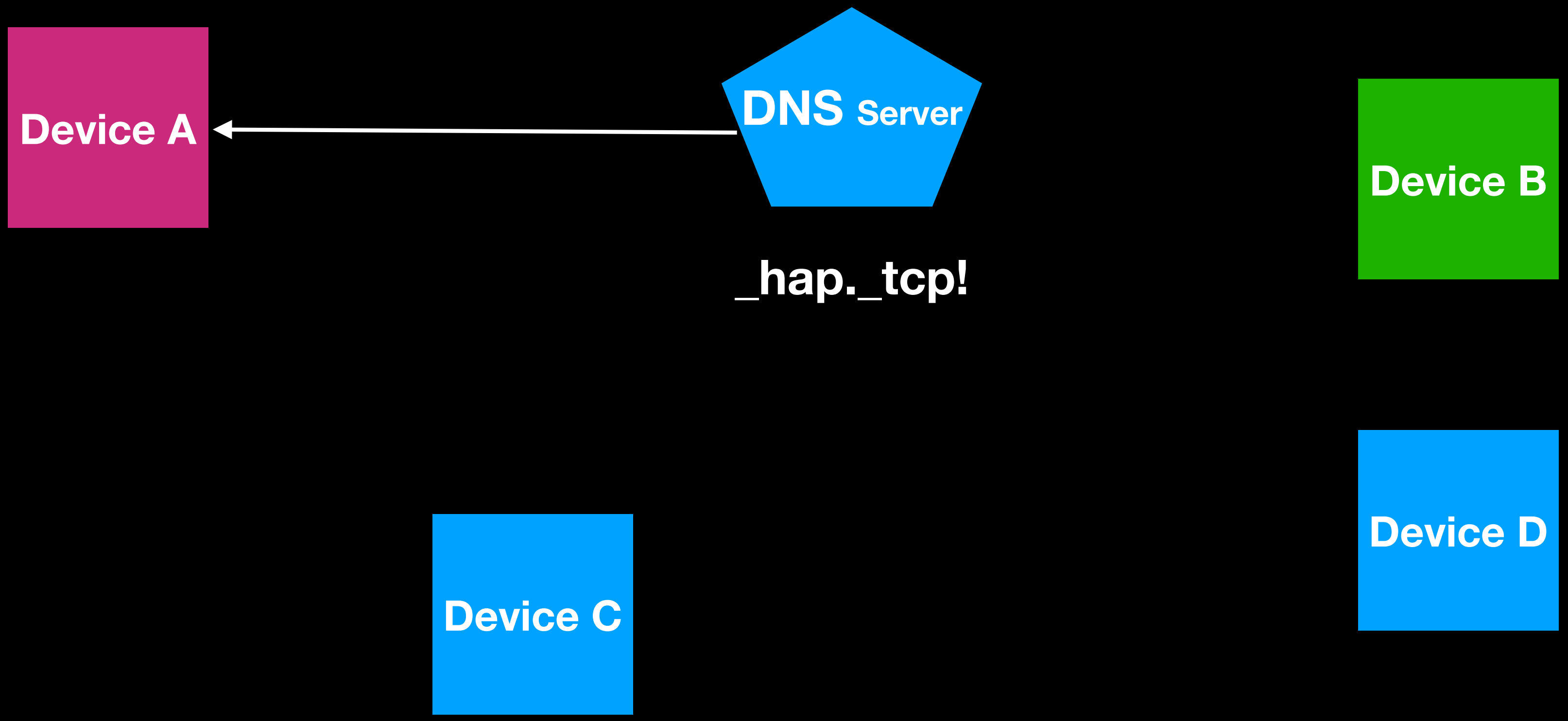
DNS-SD over mDNS



DNS-SD over DNS



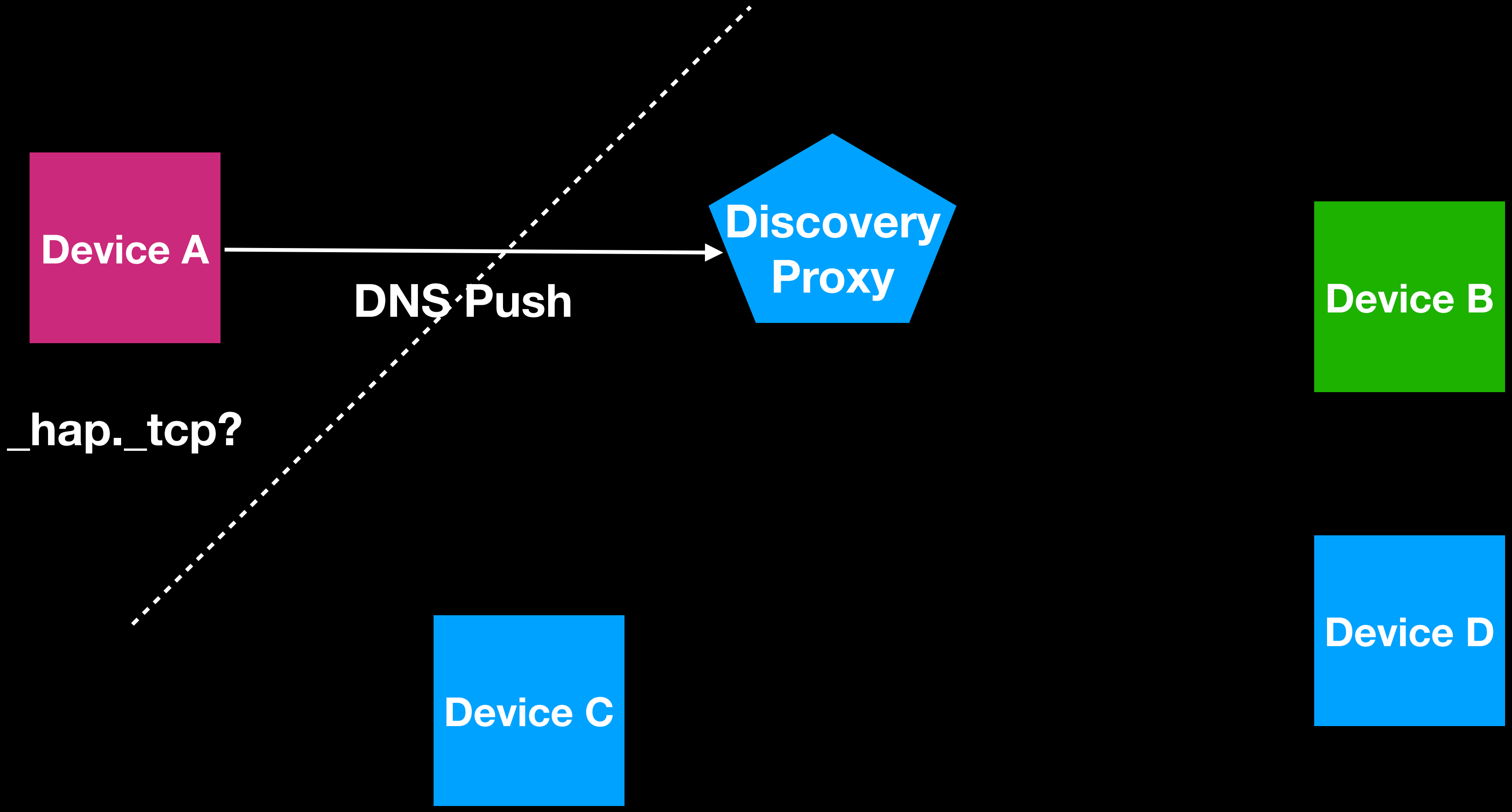
DNS-SD over DNS



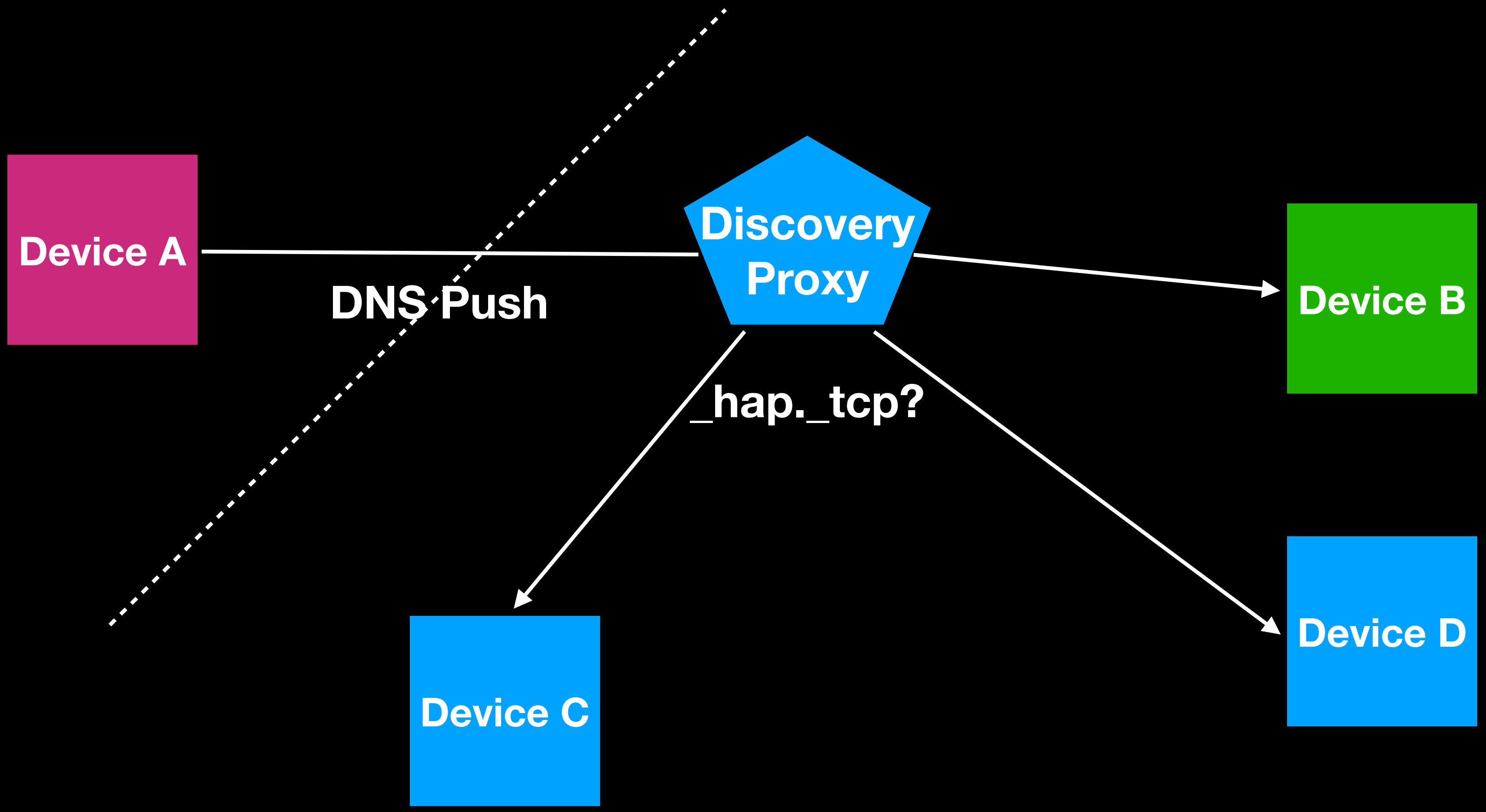
What we added in 2019

- DNS-SD Discovery Proxy (RFC8766)
  - A discovery proxy proxies between mDNS and DNS
  - Acts as an authoritative name server for queries
  - Acts as an mDNS client to collect answers for queries
- DNS Push (RFC8765)
  - Service browsing is asynchronous
  - DNS is synchronous
  - DNS Push is an extension to DNS that allows for a subscribe/push query model rather than an ask/answer query model

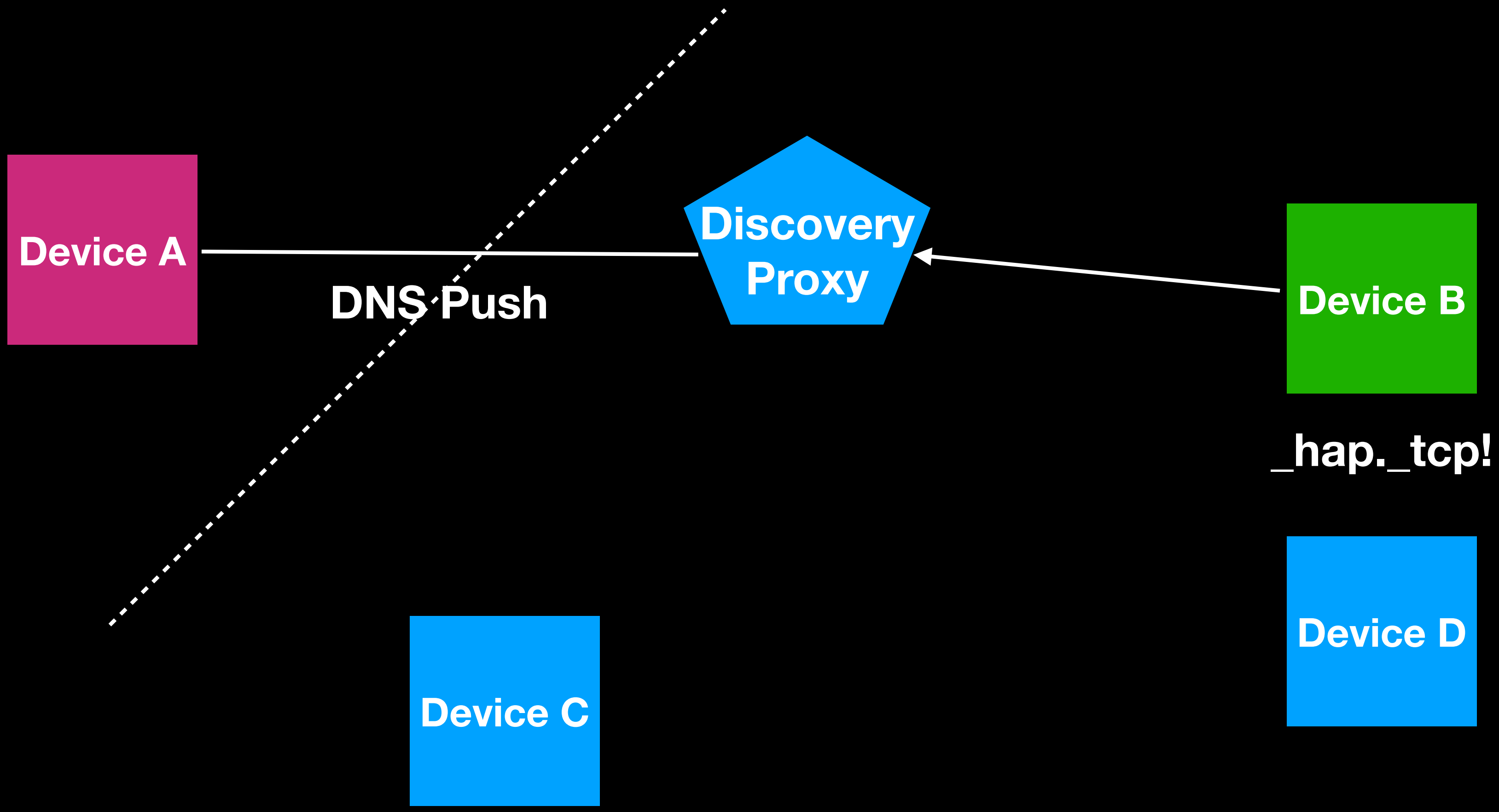
Discovery Proxy



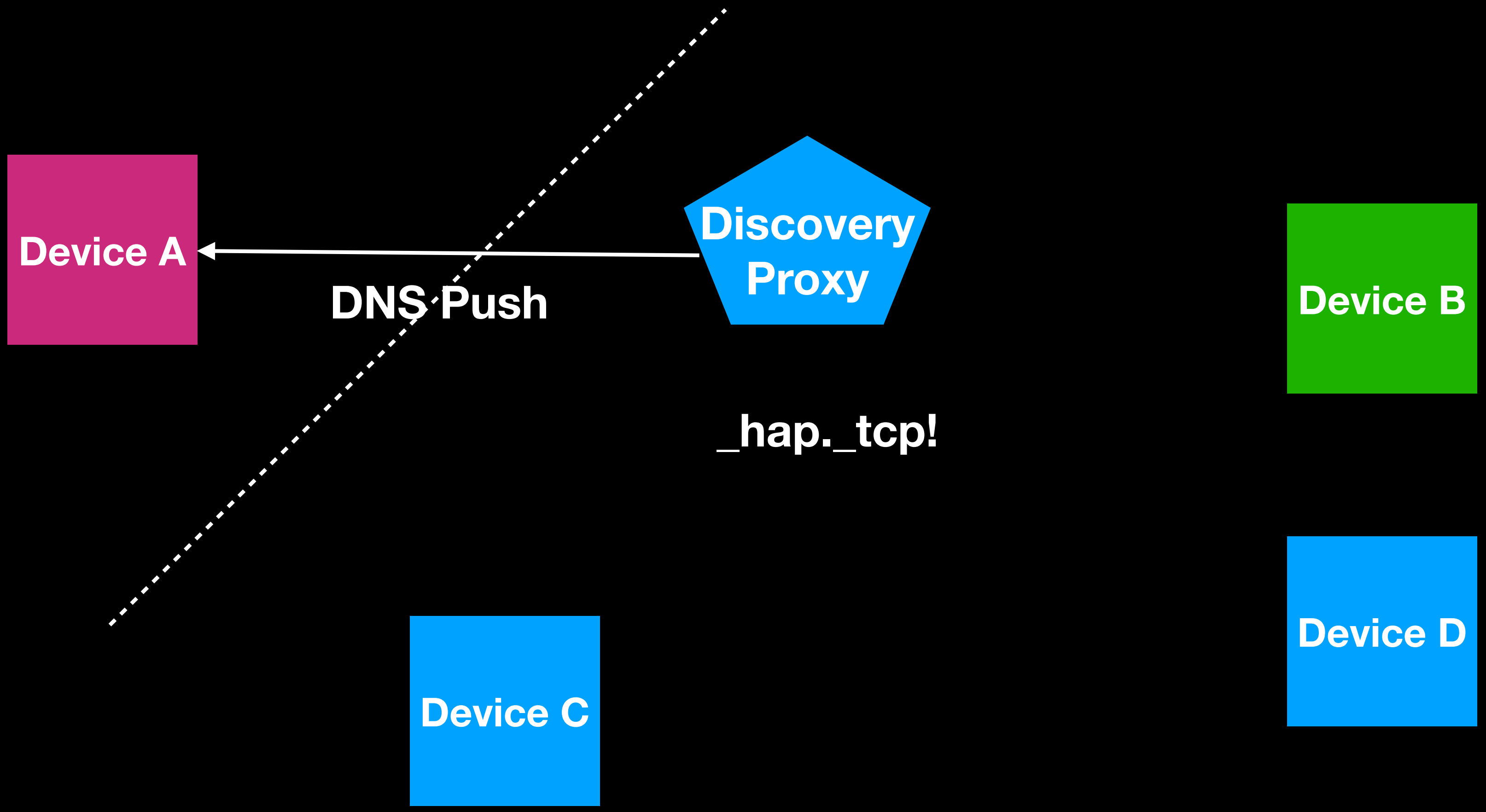
Discovery Proxy



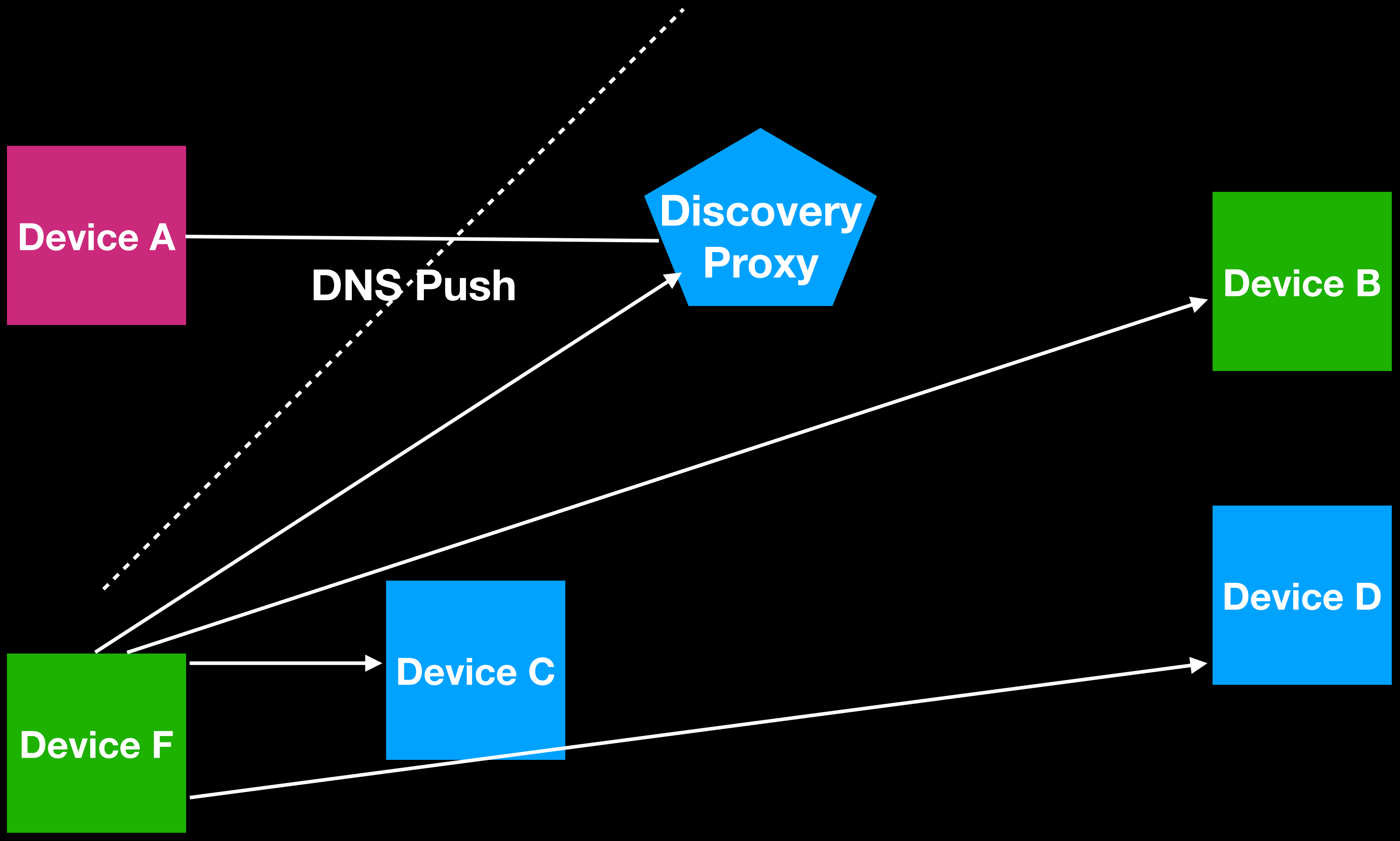
Discovery Proxy



Discovery Proxy

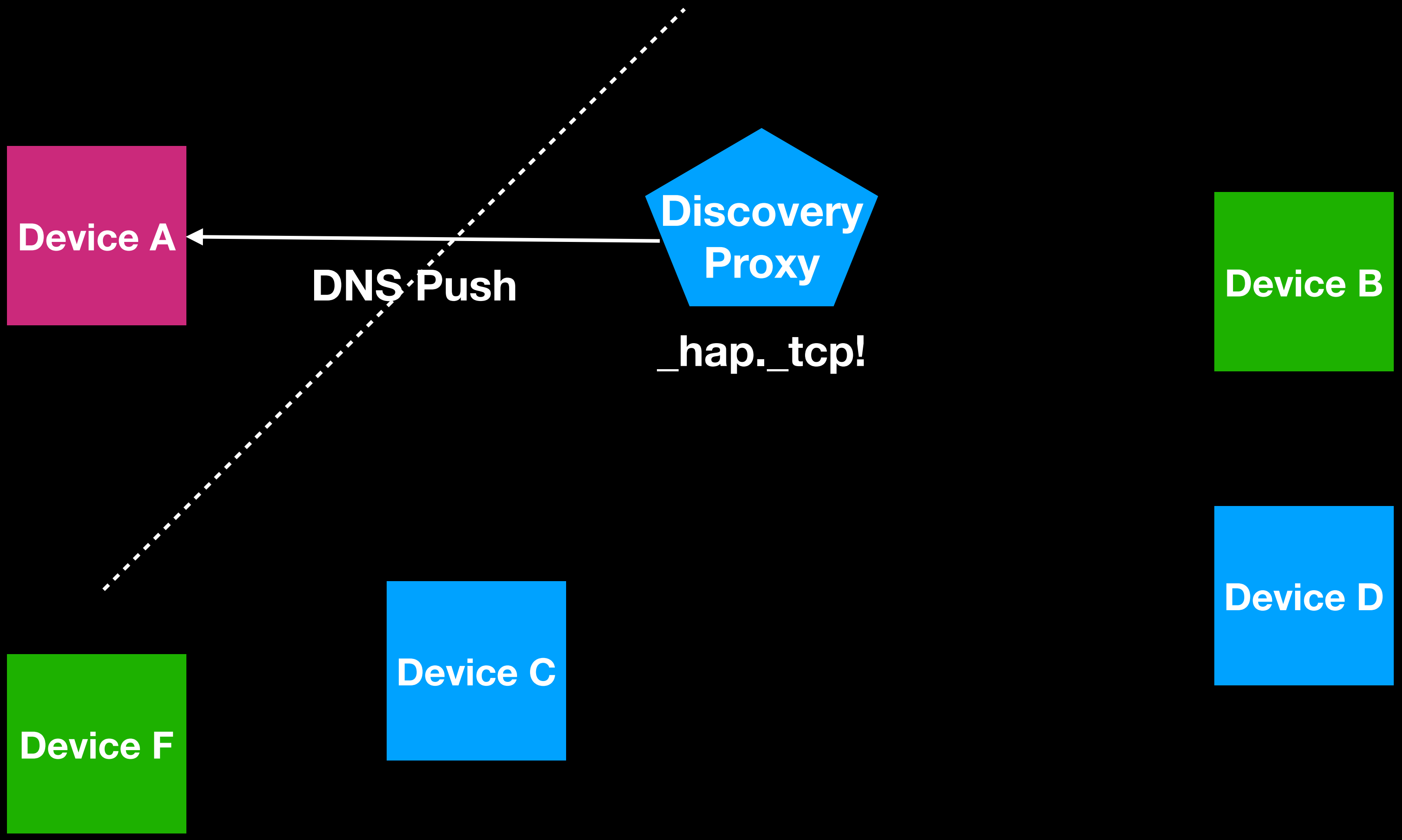


Discovery Proxy



_hap._tcp!

Discovery Proxy



What new capabilities did this enable?

- We can now use mDNS to discover services on other subnets
- DNS now supports asynchronous discovery, like mDNS:
  - we see new services as they are advertised
  - we see them leave when they are discontinued
- This means that the user experience when using DNS + Discovery Proxy is the same as when using mDNS

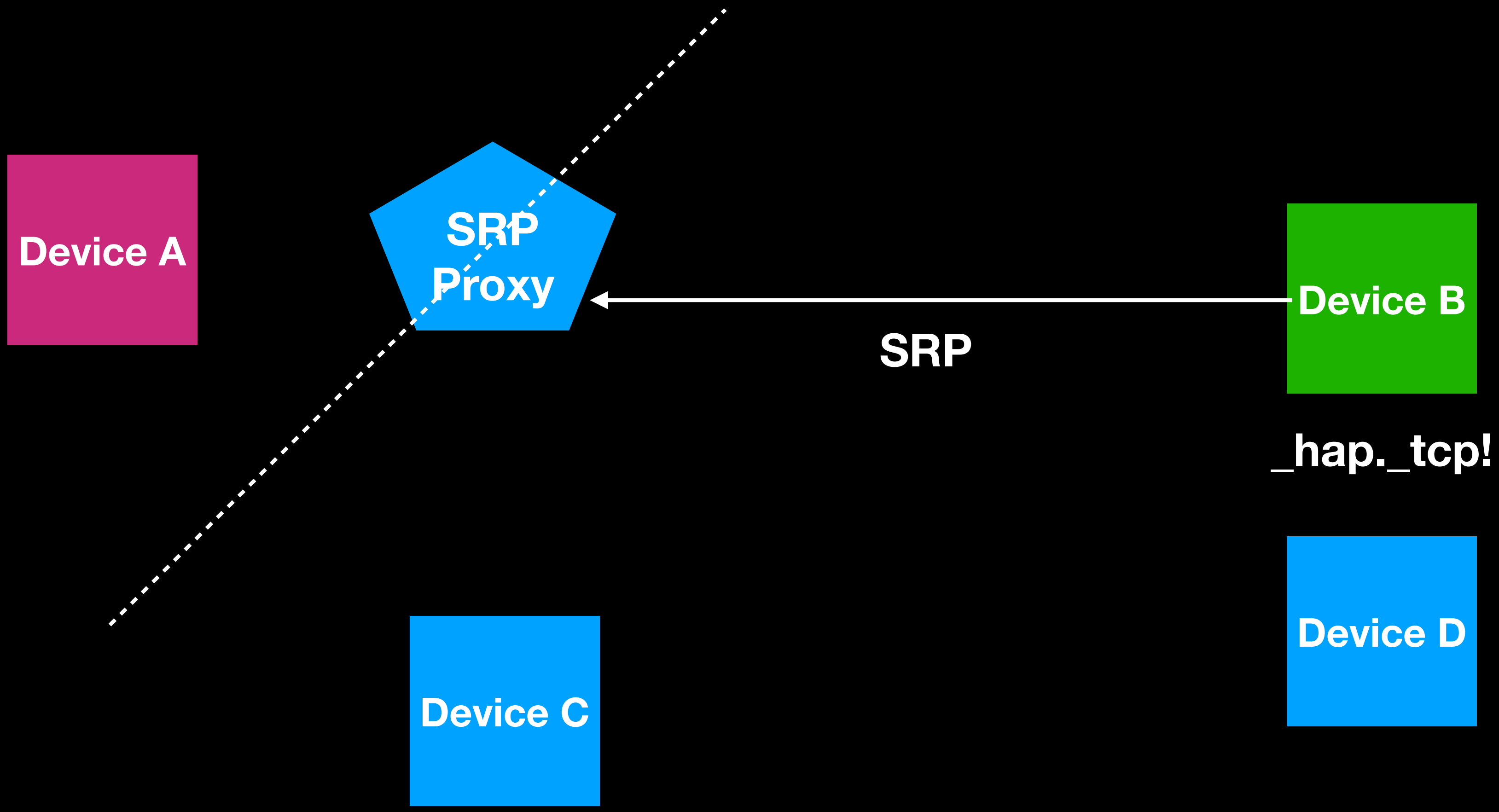
New problems that motivated SRP and Advertising Proxy

- We needed a way to support service discovery on constrained networks
- Constrained networks are subnetted from WiFi
- We could use Discovery Proxy
  - but that relies on mDNS
  - For constrained devices, spurious questions are costly
  - mDNS isn't the right solution for this problem
- So we need to register a service with a less-constrained proxy, and have that proxy advertise the service

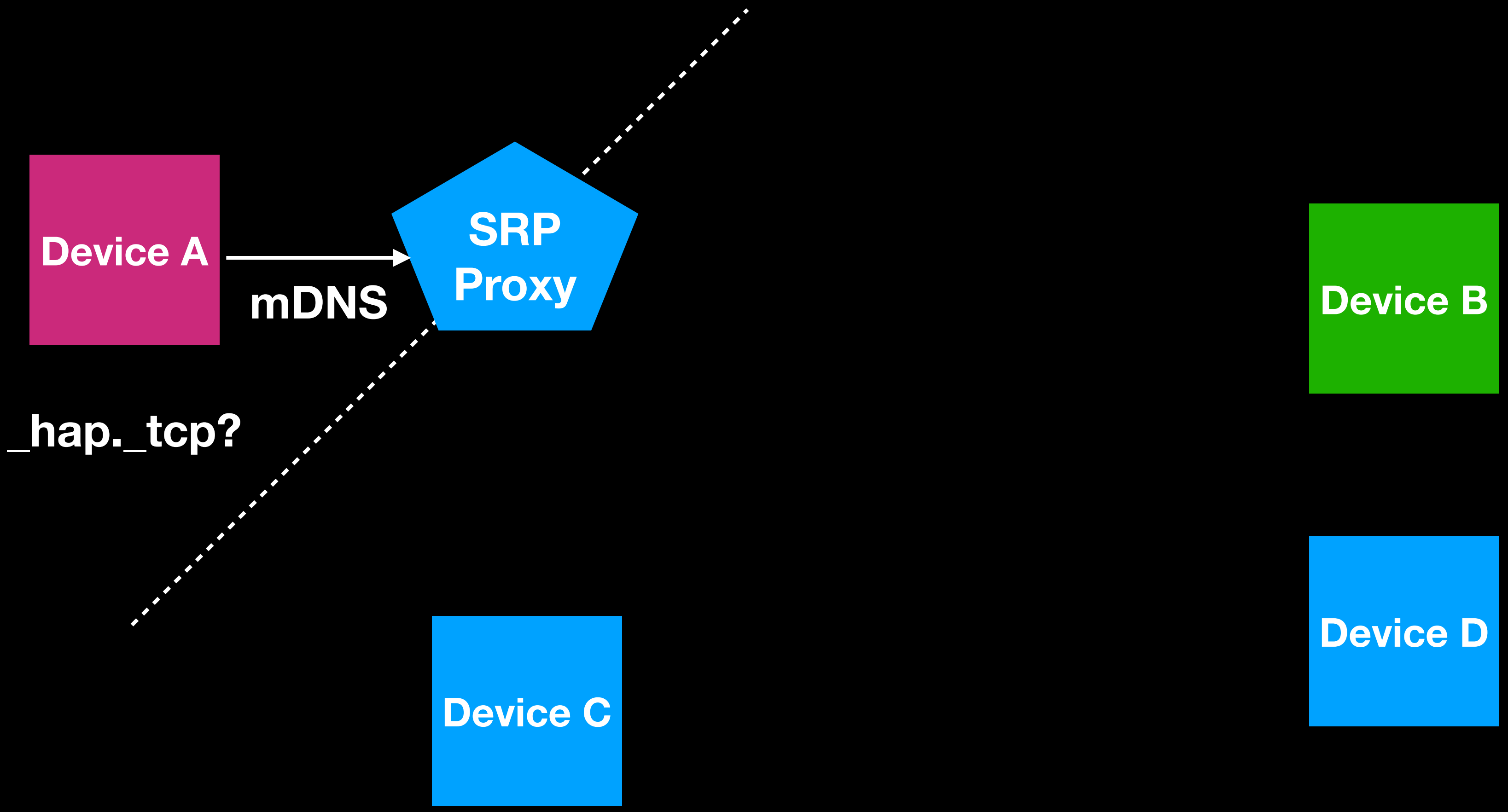
What we added

- “SRP Proxy”:
    - Service Registration Protocol (draft-ietf-dnssd-srp)
        - Services advertise themselves in a DNS zone
        - No need for mDNS as backing store for the DNS zone
    - Advertising Proxy (draft-sctl-advertising-proxy)
        - mDNS is permissionless, DNS requires configuration
        - Advertising proxy advertises the contents of a DNS zone using mDNS on an adjacent link

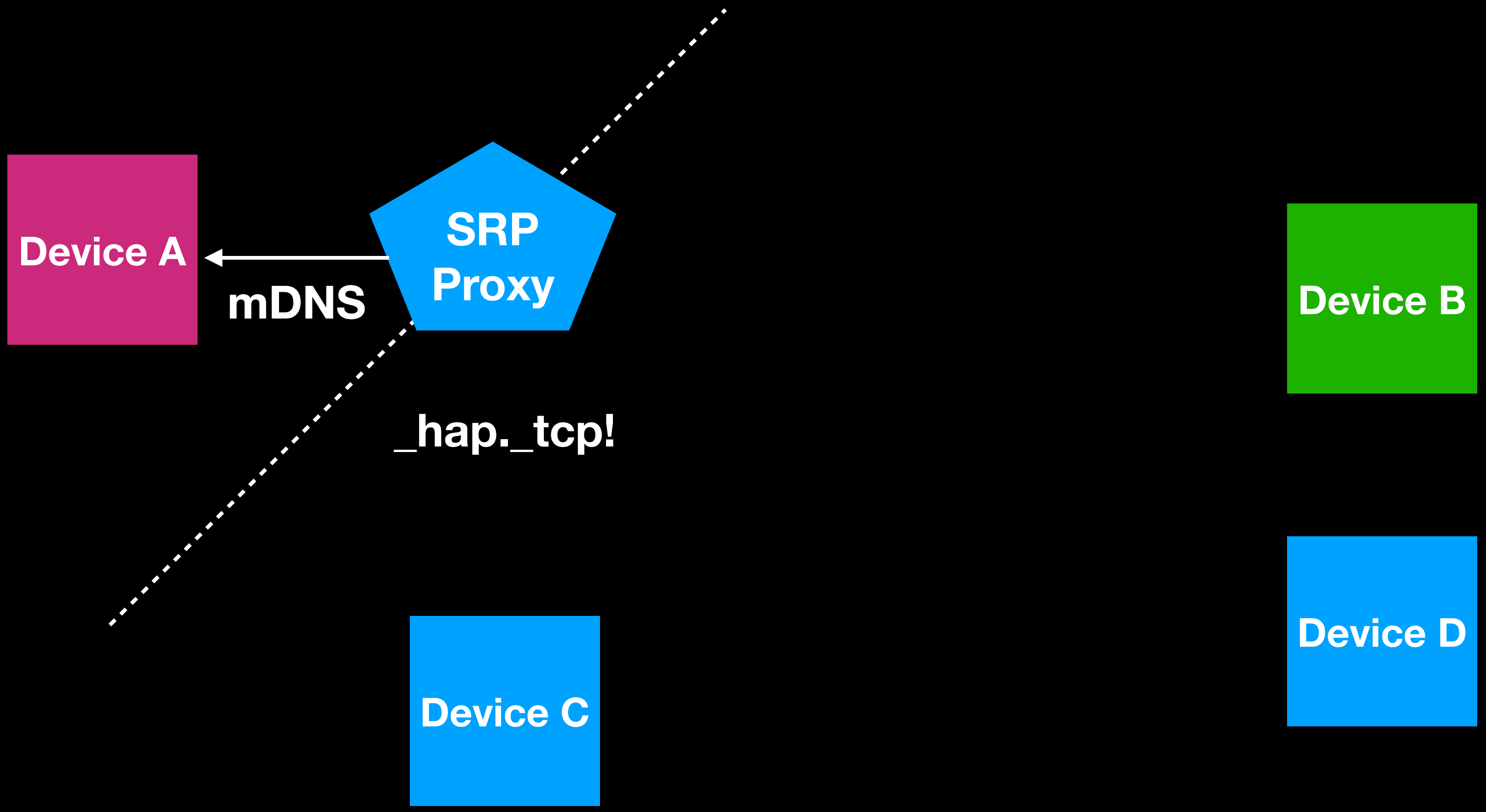
SRP mDNS Proxy



SRP mDNS Proxy



SRP mDNS Proxy



What we are asking of the WG

- draft-srp-10 is ready for last call
  - has a normative reference to draft-sekar-dns-ul, oops
- draft-sctl-advertising-proxy-02 is ready for adoption
  - There is at least one open issue: how to deal with name conflicts
  - Previous version of document said “rename.”
  - But SRP has FCFS naming, and renaming sucks
  - With SRP Replication (coming soon) we should be able to assert uniqueness
  - Discuss?
- draft-sekar-dns-ul-03 is ready for adoption

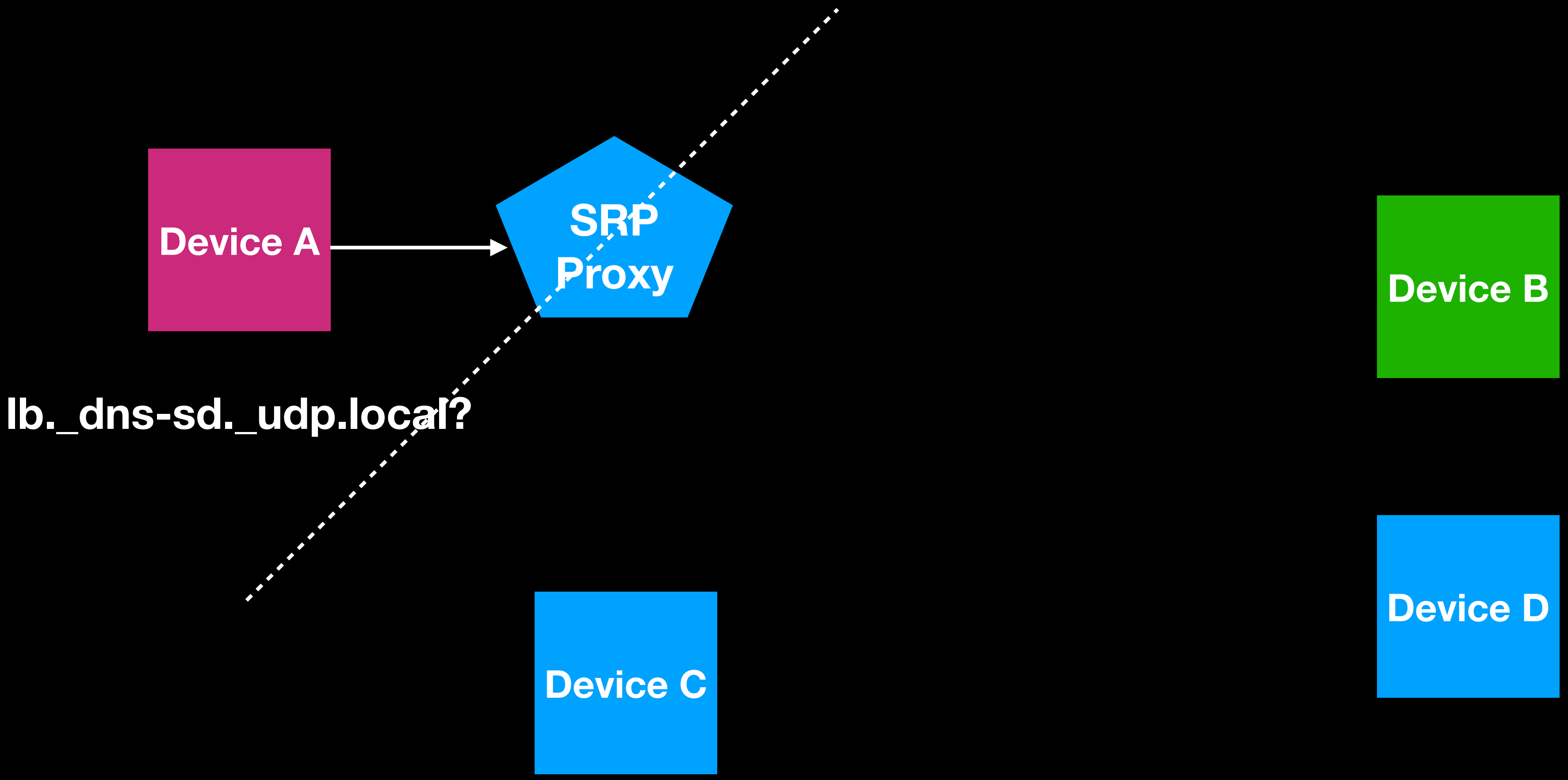
Problems with SRP Proxy

- Multiple servers do not cooperate, leading to name conflicts
- But we want multiple servers for redundancy
- Many devices per network mean lots of mDNS traffic in answer to a browse query
- Multicast on Wifi (common deployment) is expensive and unreliable

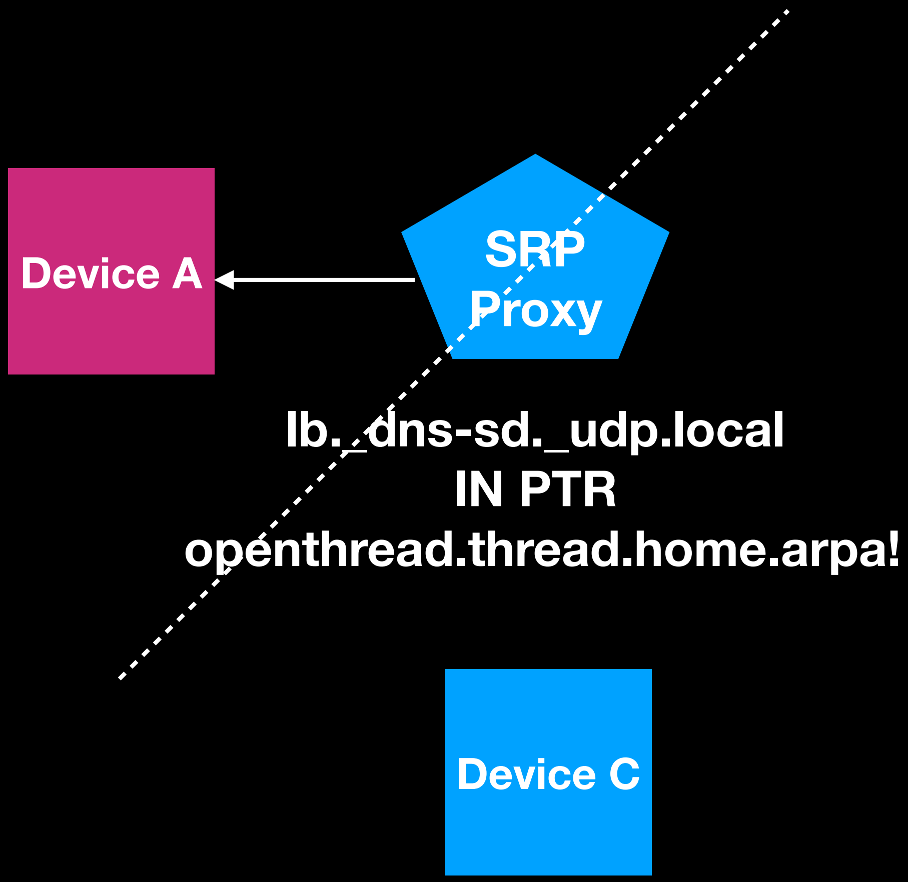
New work this year

- SRP Replication (draft-lemon-srp-replication)
  - More than one SRP server on same link will cooperate to maintain client list
- Advertising Proxy scalability
  - DNS Zone Discovery over mDNS
    - Discover the SRP server as a DNS server
  - Already suggested in RFC6763, but not much detail
  - draft-tljd-dnssd-zone-discover provides a lot of detail
  - SRP Proxy as authoritative server
    - SRP server answers authoritatively for SRP zone
    - Also described in draft-tljd-dnssd-zone-discover
    - SRP replication provides the common zone for DNS queries

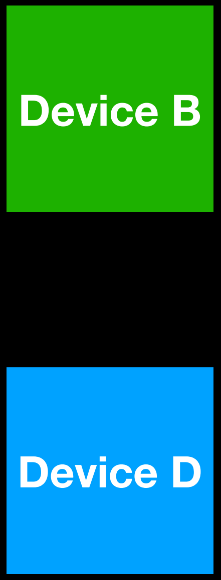
SRP DNS Proxy



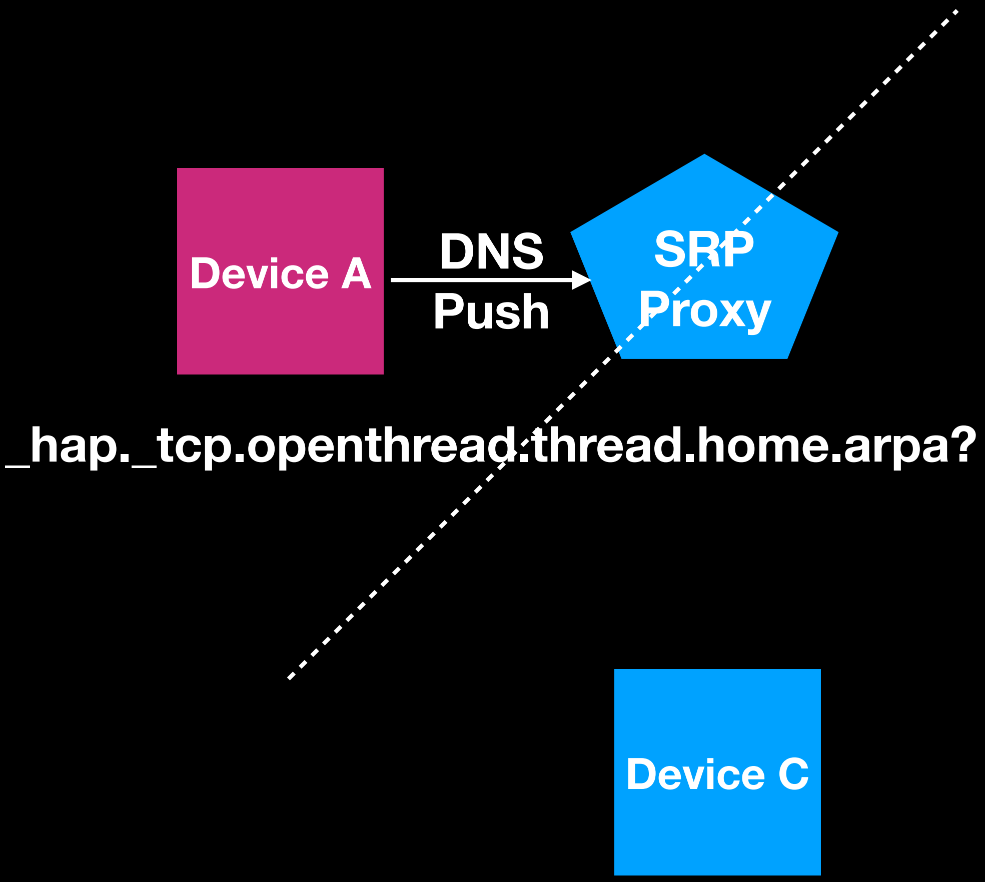
SRP DNS Proxy

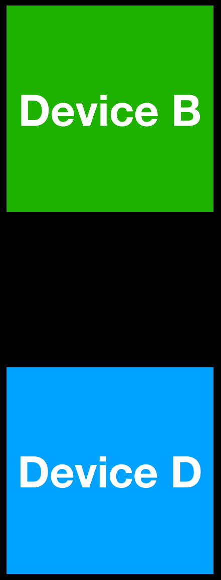


lb._dns-sd._udp.local
IN PTR
openthread.thread.home.arpa!

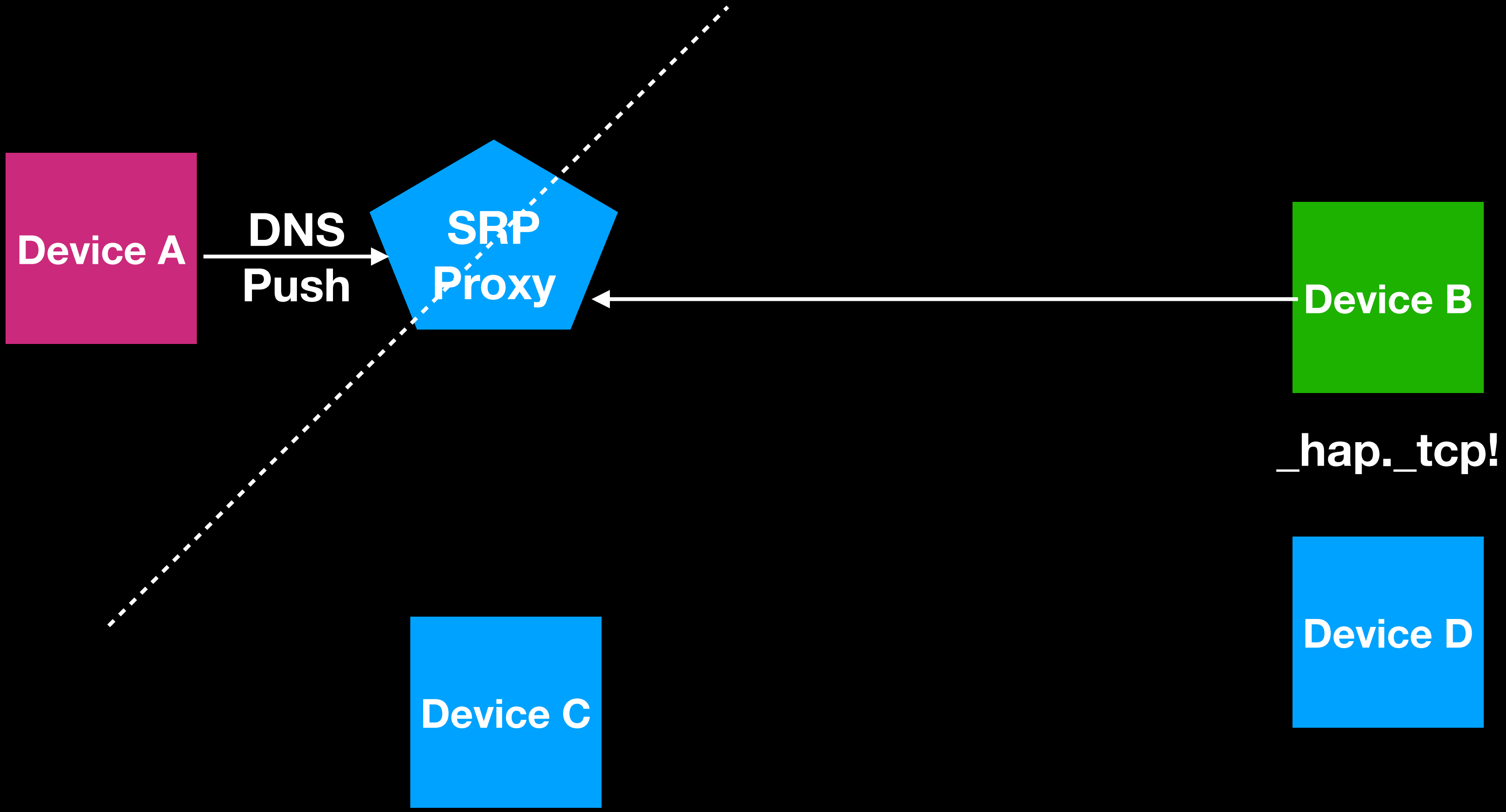


SRP DNS Proxy

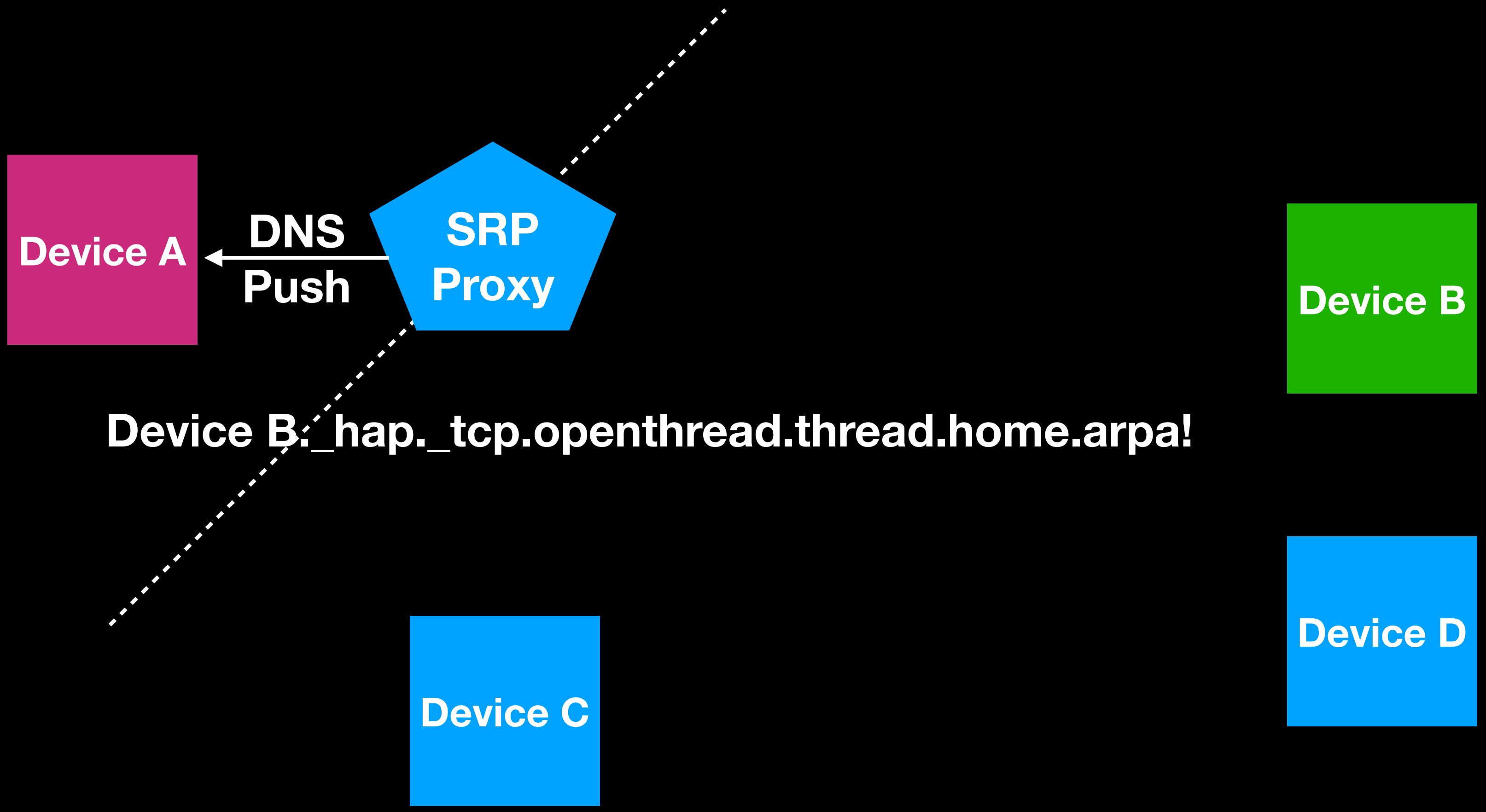




SRP DNS Proxy



SRP DNS Proxy



Device B._hap._tcp.openthread.thread.home.arpa!

Open issues

- SRP Replication is pretty solid, but could use more review
  - Review may turn up substantive issues
  - There may be opportunities for simplification
- Zone Discovery has some known issues
  - In the ad-hoc scenario, how we do we agree on a name?
  - Document describes several approaches
  - Some can coexist, but some decisions need to be made
  - Current implementation uses one of these alternatives

What we are asking of the WG

- Call for adoption on draft-lemon-srp-replication
- Call for adoption on draft-tljd-zone-discover

Remaining Issues

- Still reliant on mDNS
  - Can’t get away from this entirely because of legacy devices and networks,
  - but currently we don’t even provide a way for the infrastructure to support SRP zones and Discovery Proxy zones
  - mDNS is not reliable on WiFi
  - We see a lot of bug reports because of this
  - Some proprietary mDNS enhancements actually break mDNS even when multicast is working
  - So we really want to enable moving away from mDNS

What’s needed (1)

- Network infrastructure needs to be able to integrate Stub network DNS-SD proxies, such as Discovery Proxies and SRP proxies
  - Document how Stub Network DNS-SD proxies register with infrastructure automatically
- All devices should be able to do all service discovery using DNS-SD
  - Document how network signals that this is safe to do
- All services should advertise using SRP instead of mDNS when available
  - Document how network signals this is available

What’s needed (2)

- mDNS is only used to support legacy devices
  - Discovery Proxies on WiFi access points proxy DNS queries to mDNS, unicast individually to each connected WiFi device
  - No multicast on WiFi
  - mDNS queries from connected devices go to Discovery Broker, which decides which answers to give, and responds using mDNS only to the device that asked the question (not actually multicast, if possible)
  - Document how this is done