Reviving NAT-PT
-
two proposals

IETF 70, December 2007, Vancouver

Iljitsch van Beijnum

Content vs access

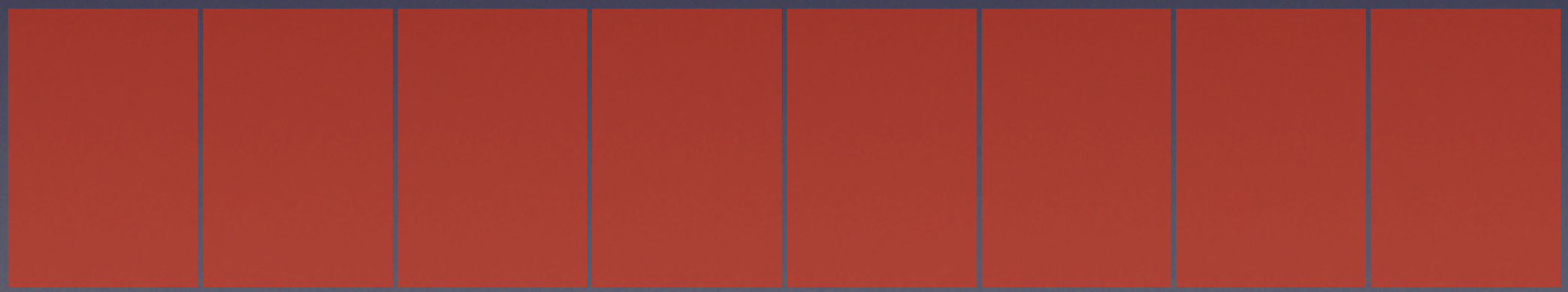
- Content providers:
  - use few addresses, depletion not an issue
  - either A or A+AAAA, all or nothing
- ISPs:
  - need new addresses for new customers
  - can give new users v6 without impact to installed base
- Could end up content on v4, eyeballs on v6

The dual stack problem

- Dual stack:
  - doesn't solve anything, still need v4 addresses
    - twice the work
    - if need to do v4 anyway, why add v6?
- NAT-PT solves the interoperability issue
  - v6 host can talk to v4 host

IPv4

API Host LAN Home GW ISP ... Ser-vice

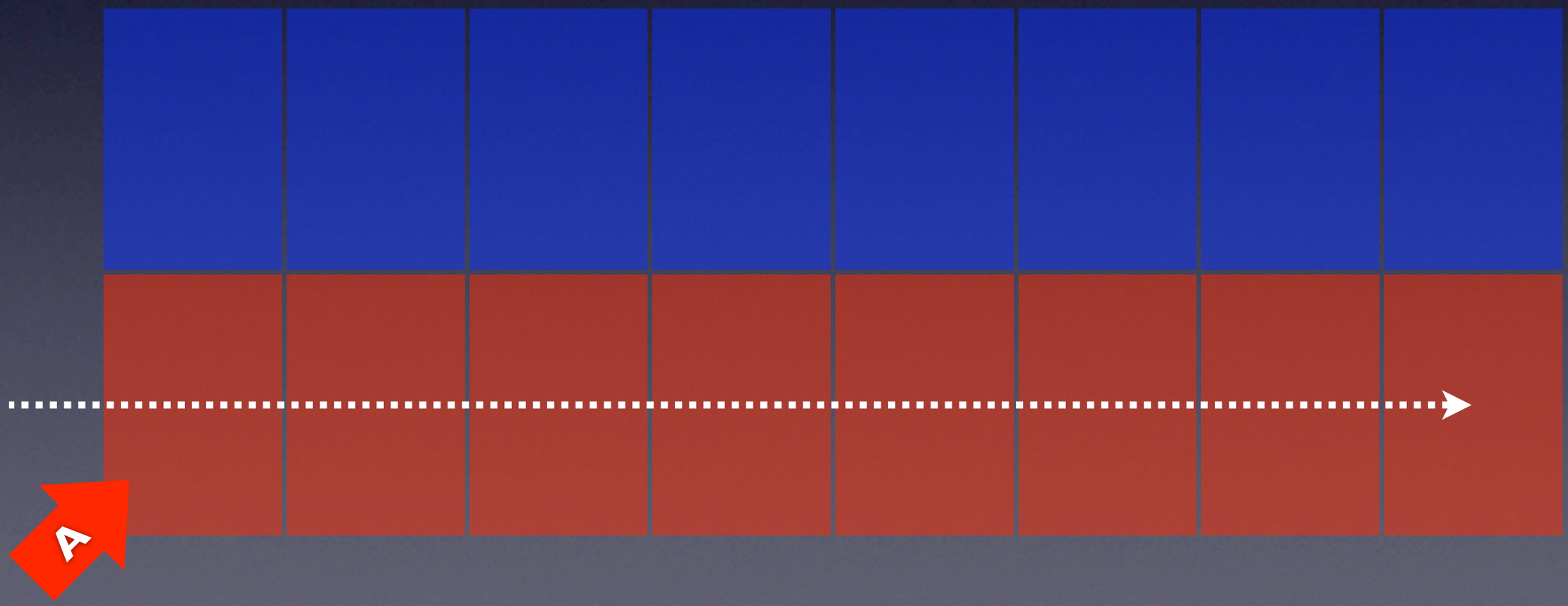


IPv4

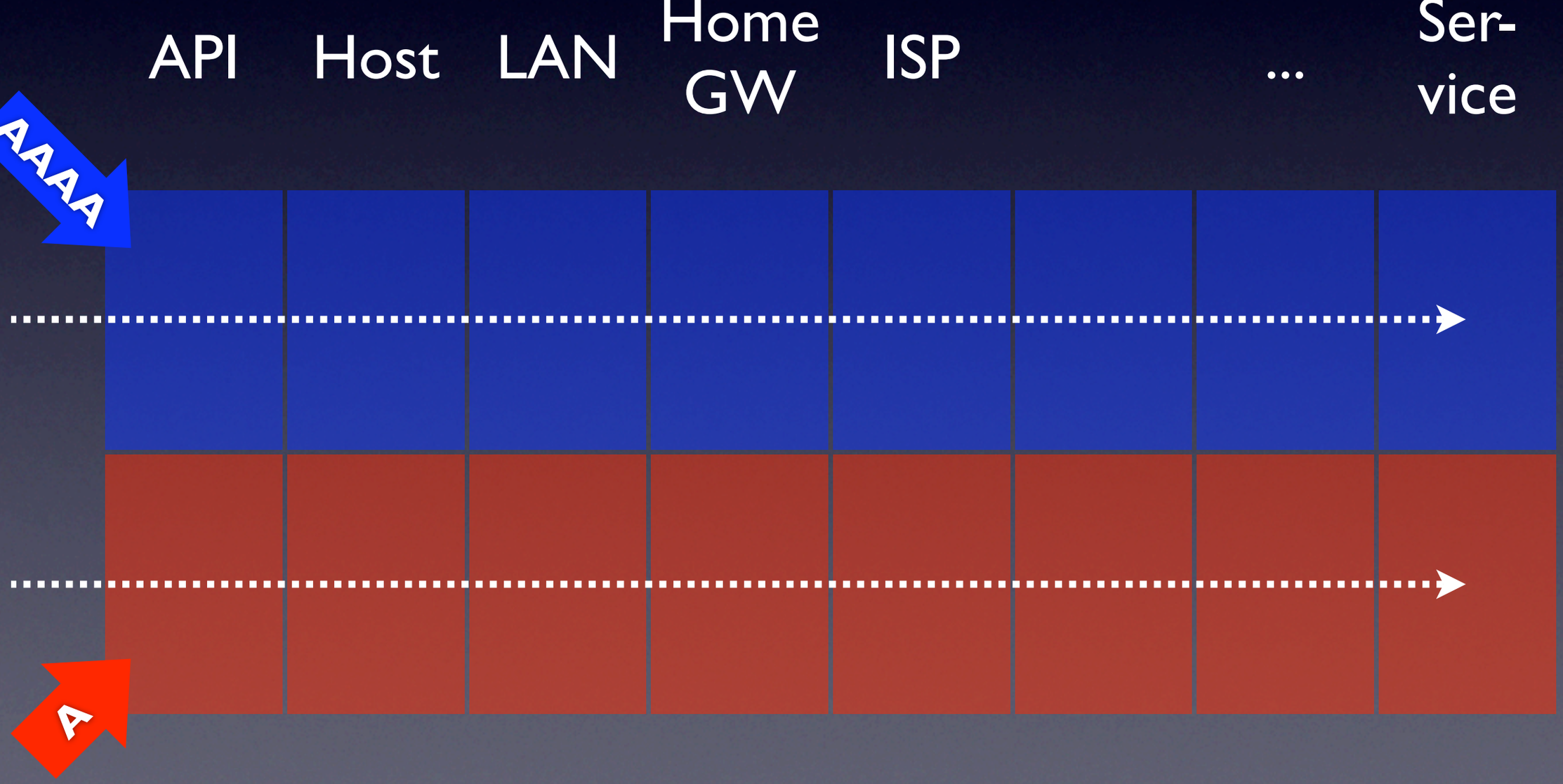
API Host LAN Home GW ISP ... Ser-vice

Dual Stack

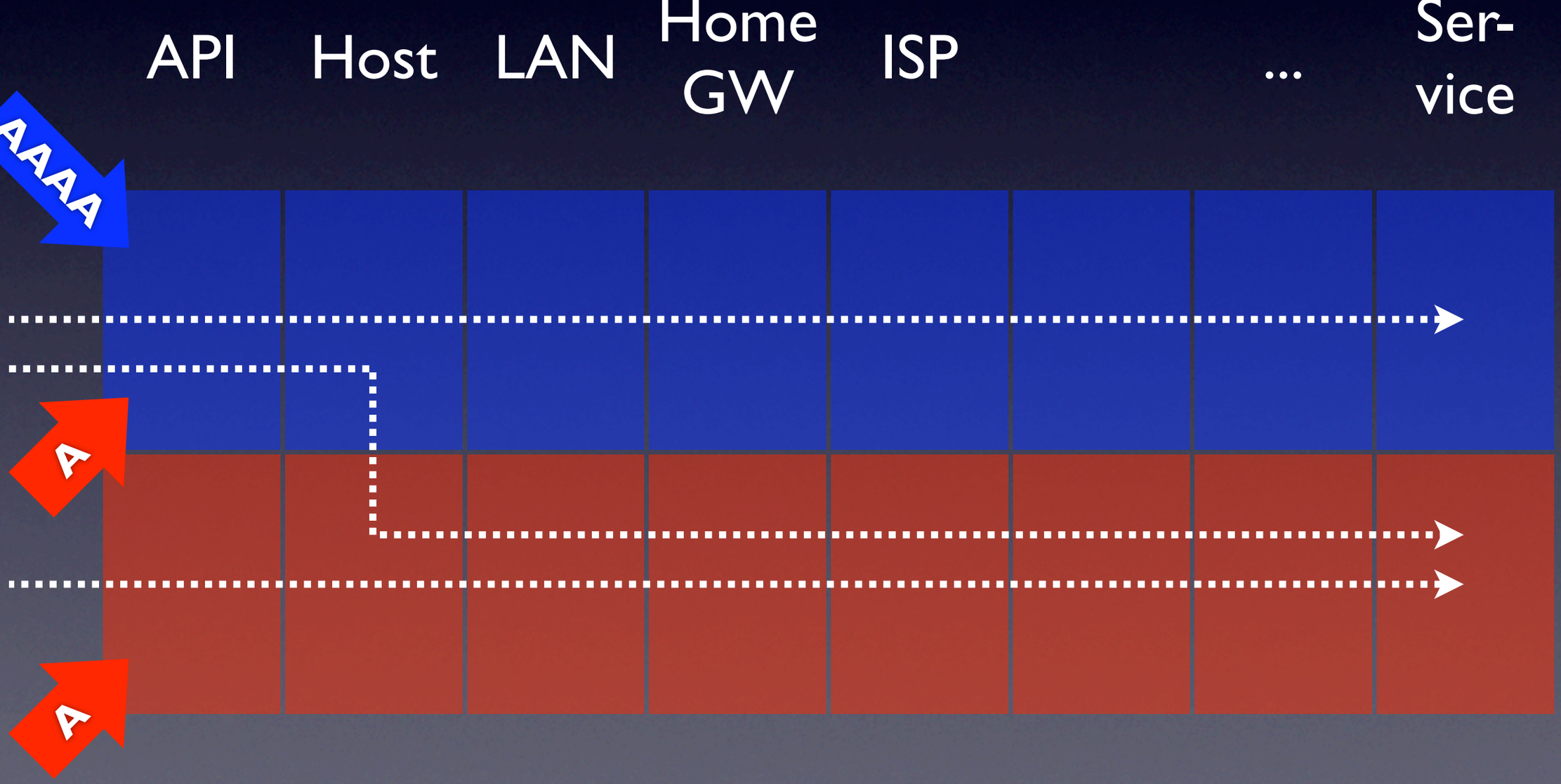
API Host LAN Home GW ISP ... Ser-vice



Dual Stack



Dual Stack



RFC 2766 NAT-PT

API Host LAN Home GW ISP NAT-PT ... Ser-vice

RFC 2766 NAT-PT

API Host LAN Home GW ISP NAT-PT ... Ser-vice

RFC 2766 NAT-PT

API | Host | LAN | Home GW | ISP | NAT-PT | ... | Ser-vice

RFC 2766 NAT-PT

Shimmed IPv4/IPv6 Address Network Translation Interface (SHANTI)

draft-carpenter-shanti-01

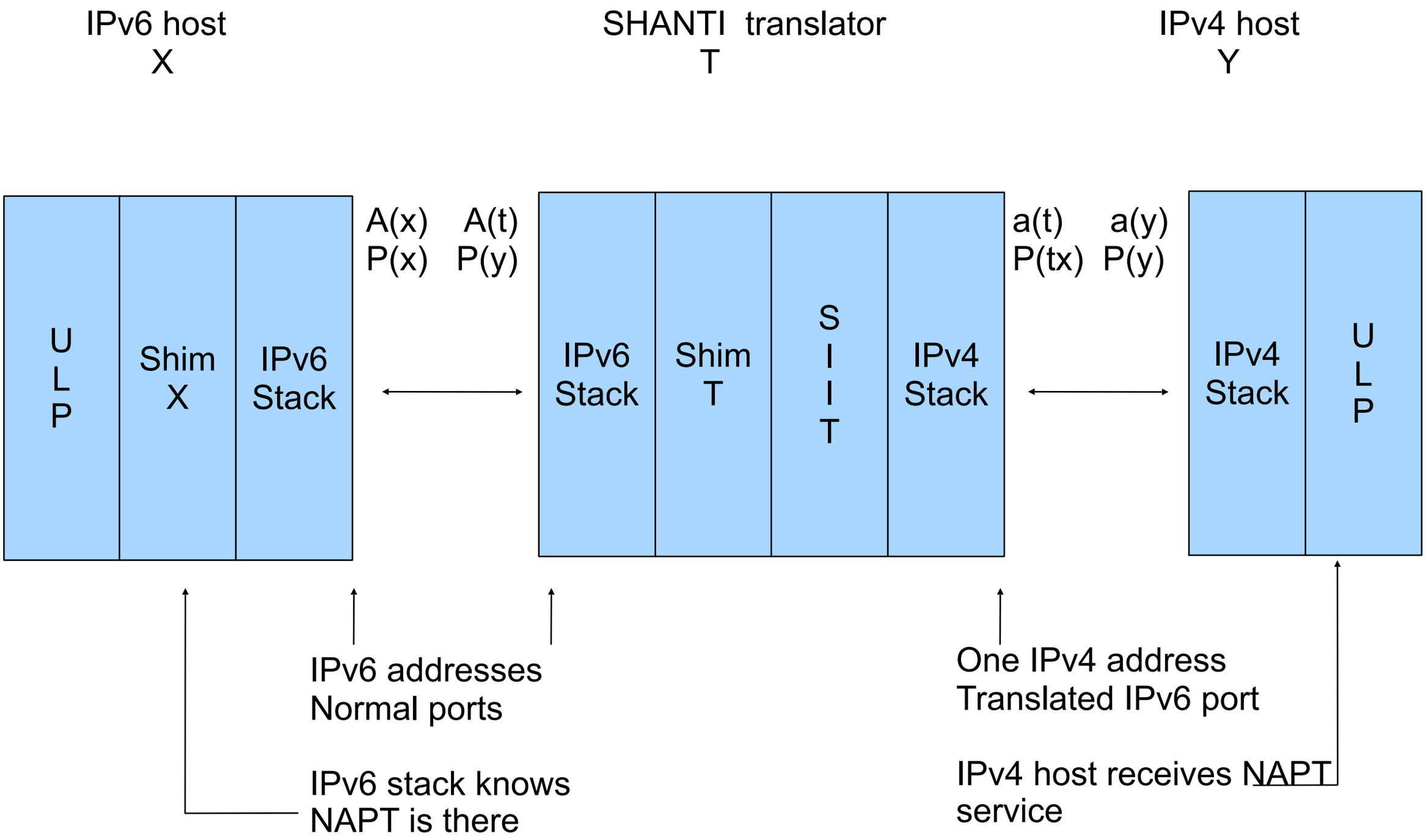
Brian Carpenter
channeled by
Iljitsch van Beijnum

IETF 70

Goals

- From the IPv4 host's point of view, nothing should be worse than IPv4-to-IPv4 NAPT.
- From the IPv6 host's point of view, information about the translation is available in the IPv6 host's network stack.
  - All ULP manipulations can be done in the host; no external ALG.
- IPv6 routing is not affected in any way, and no "entropy" is imported from the IPv4 routing tables.

Model: one IPv4 address represents many IPv6 addresses



What the shim does

- Shim tells the IPv6 ULP the IPv4 addresses and P(tx). [DNS tells it a(y) for outbound sessions.]
  - IPv4 addresses represented as IPv4-mapped IPv6 addresses.
  - Thus ULP has 4-tuple {a(t), a(y),P(tx),P(y)} for checksum calculations.
  - No ALG code required at the translator.
- The shim translates to and from regular IPv6 addresses for the IPv6 path.
  - No impact on IPv6 routing.
  - No topological restrictions.

  See draft for inbound & outbound walkthroughs

DNS

- Suggest a resolver that maps A records into AAAA records expanded with ::ffff:0:0/96
- Avoids DNS ALG and any strangeness in stored DNS records
- Robust (fate-sharing with the IPv6 host itself, zero impact on DNS server)
- Compatible (will not affect behavior of any non-SHANTI host)

NAT-PT issues from RFC 4966

- SHANTI removes issues created by DNS-ALG
- Mitigates issues of address-dependent ULPs (IPv4 addresses are available to ULP)
  - Ditto for port-dependent ULPs
  - No need for ALGs *en route*
- Removes topology constraints
- Cannot solve
  - issues with IPsec
  - binding state timeout
  - problems with fragmentation
  - single point of failure

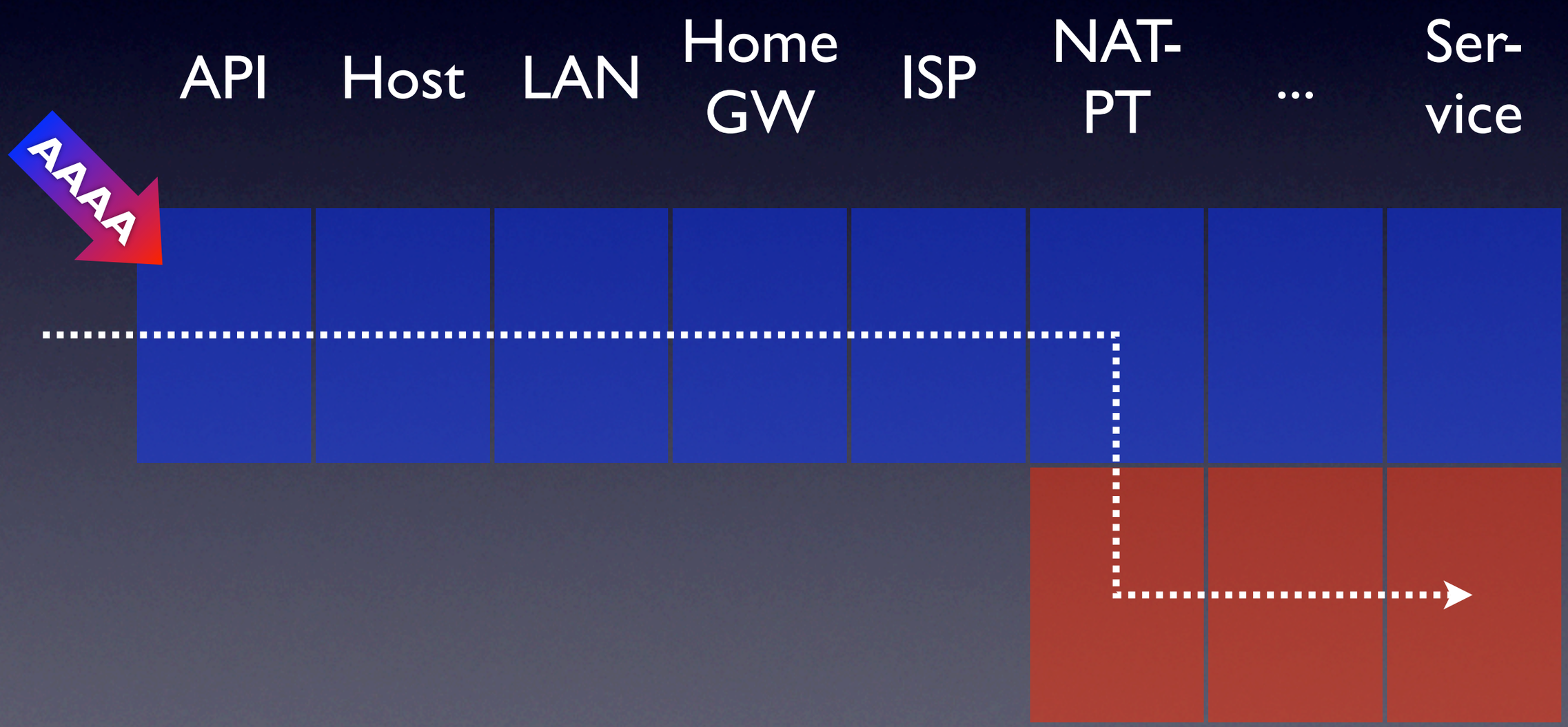
  See draft for detailed analysis

draft-van-beijnum-modified-nat-pt-02.txt

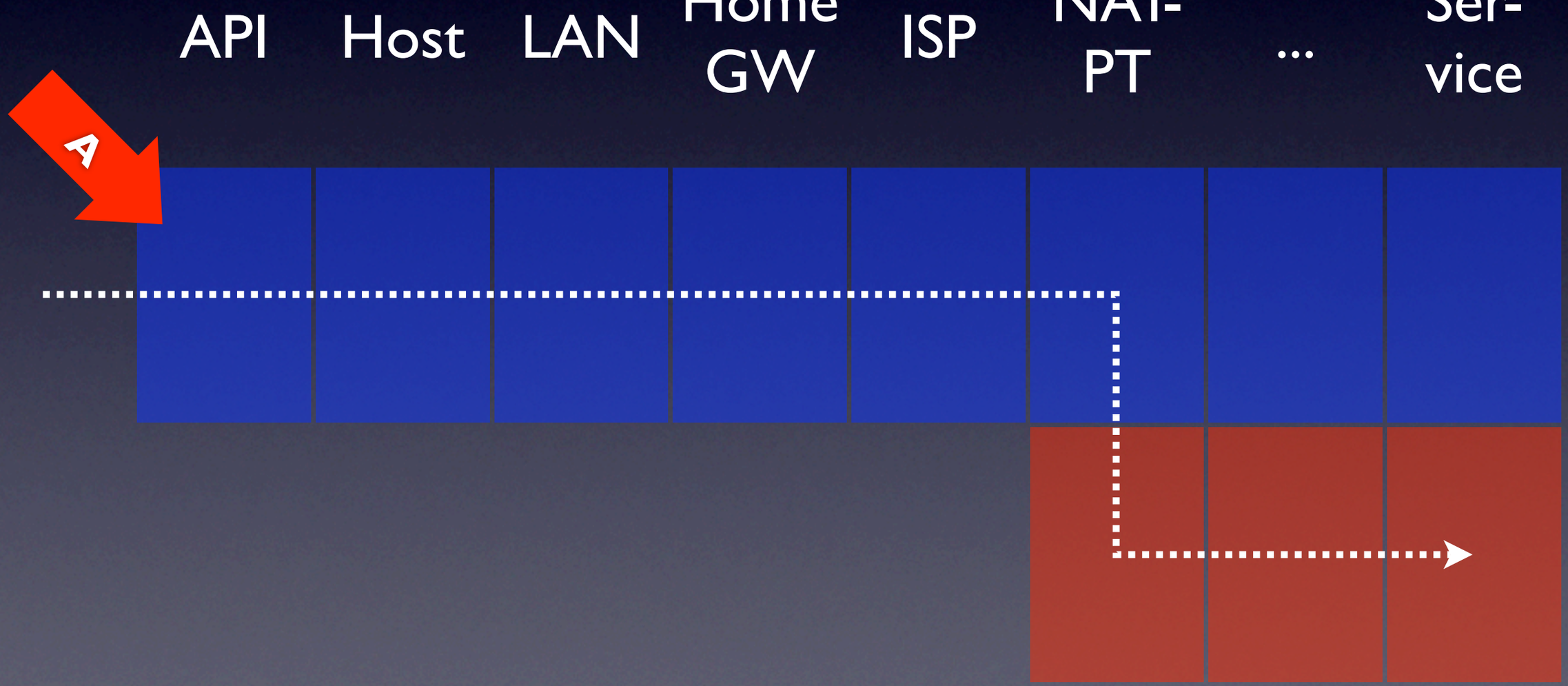
IETF 70, December 2007, Vancouver

Iljitsch van Beijnum

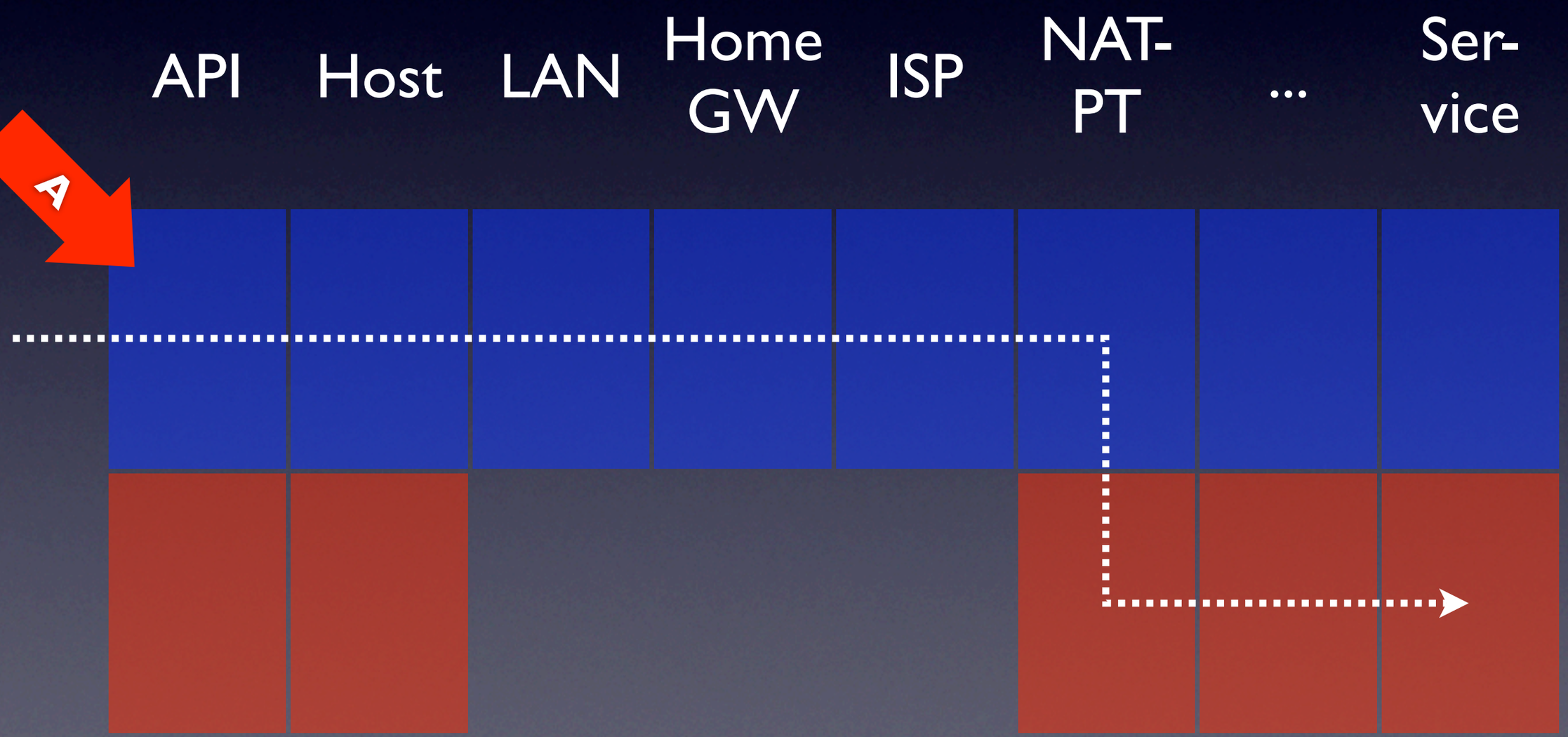
Modified NAT-PT



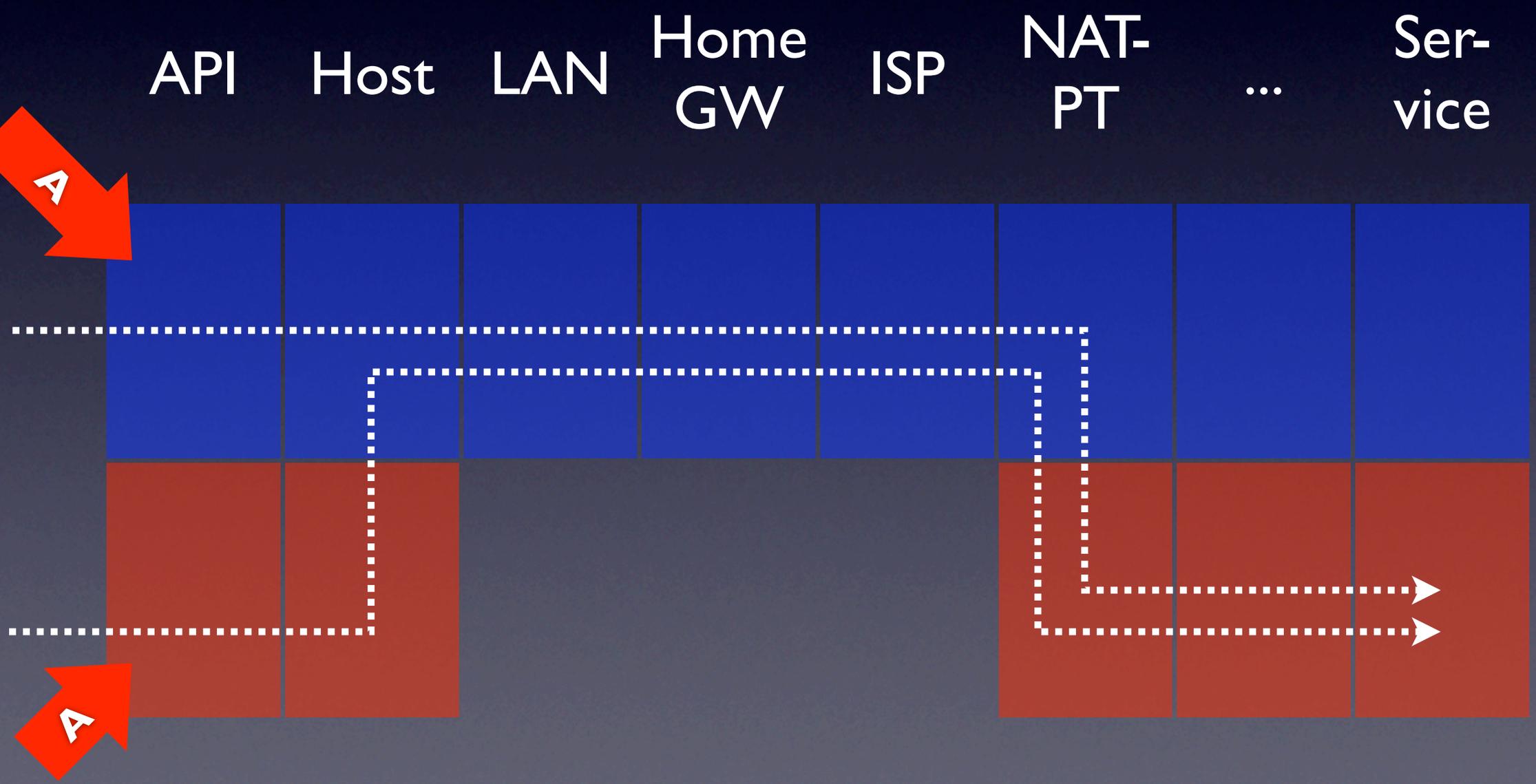
Modified NAT-PT



Modified NAT-PT



Modified NAT-PT

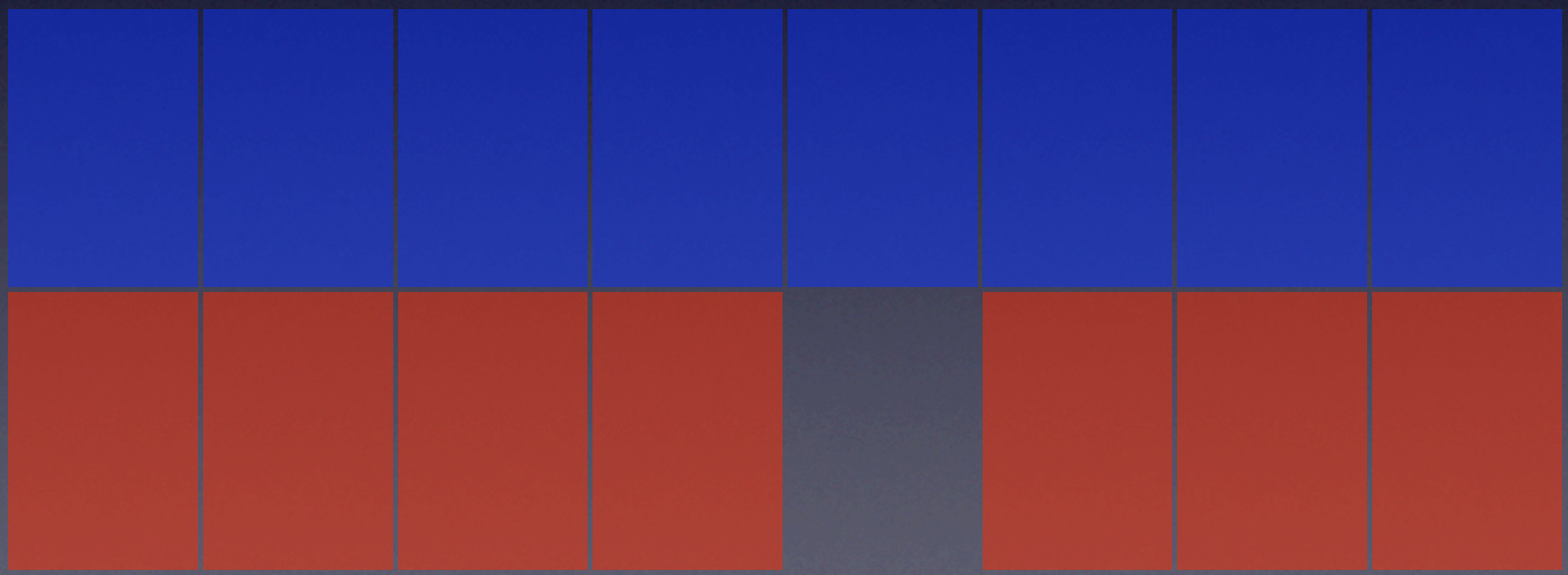


MNAT-PT

- No more DNS ALG: hosts do A lookup and create 96+32 bit address
- Means you can use IPv4 Socket API = IPv4-only apps
- Present fake RFC 1918 source address to apps to signal IPv4+NAT
- Anycast or configured /96 prefix

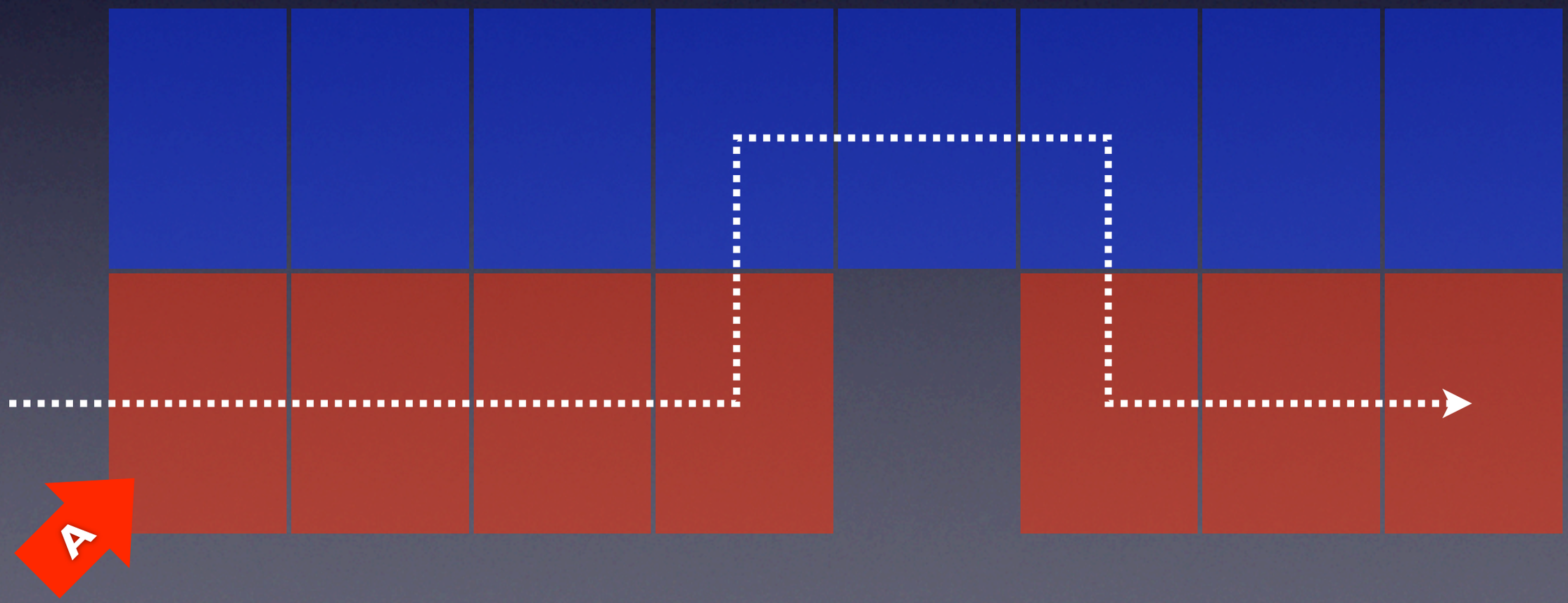
v4-v6-v4 MNAT-PT

API | Host | LAN | Home GW | ISP | NAT-PT | ... | Ser-vice



v4-v6-v4 MNAT-PT

API | Host | LAN | Home GW | ISP | NAT-PT | ... | Ser-vice



v4-v6-v4 MNAT-PT (2)

- Home gateway does SIIT
- NAT still only happens in remote translator
  - main advantage: no double NAT
- Compatible with IPv4-only hosts

IPv4-to-IPv6

- Set aside block of IPv4 addresses
- Map each IPv4 address/port to an IPv6 address/port in the DNS
- /8 gives you 2^{40} address/port combos
- Closest translator translates
- Encode IPv4 address in bottom IPv6 bits for IPv4-aware apps on IPv6 host

Pros of MNAT-PT

- No DNS ALG ugliness
- Compatible with IPv4-only apps/hosts
- Feed people IPv6 along with IPv4 NAT
- Neat tricks: new translator prefix for new sessions, keep old ones on old translator

Something different...

- draft-van-beijnum-v6ops-connect-method-00.txt
- HTTPS proxy is really a generic TCP proxy
- Simple and easy for all TCP apps
  - especially if talking to proxy done in OS
- Works both v6→v4 and v4→v6

My opinion

- Not *the* answer but *an* answer
- Fruitful work seems possible in this area
- But not much time, better make a protocol too many fast than one too few slowly
- IPv4 + more & more NAT: only gets worse
- IPv6+NAT-PT: gets better with time

Questions?

- draft-carpenter-shanti-01
- draft-van-beijnum-modified-nat-pt-02.txt
- draft-van-beijnum-v6ops-connect-method-00.txt