FA-IINAS: Functional Addressing (FA) for internets with Independent Network Address Spaces (IINAS)(*)

Toerless Eckert, Futurewei USA (tte@cs.fau.de)

07/28/2021

RTWG & INTAREA

(*) Will pay for better name/abbreviation: Coffee, Pizza, Sushi!

What could we do with variable long addresses better ?

- Start with NOT IETF widely explored benefits – operational issues of addressing
  - Network that would like to assign/grow its address space independent from other networks
  - No need to bother assigning / managing assigning global addresses
  - Build internetworks when required without need to manage global addresses

- Crossover issue for multiple WG/(RG?)
  - Probem part presented to iotops – maybe split out problem statement from draft and extend
  - Addressing – INTAREA, Routing - RTGWG

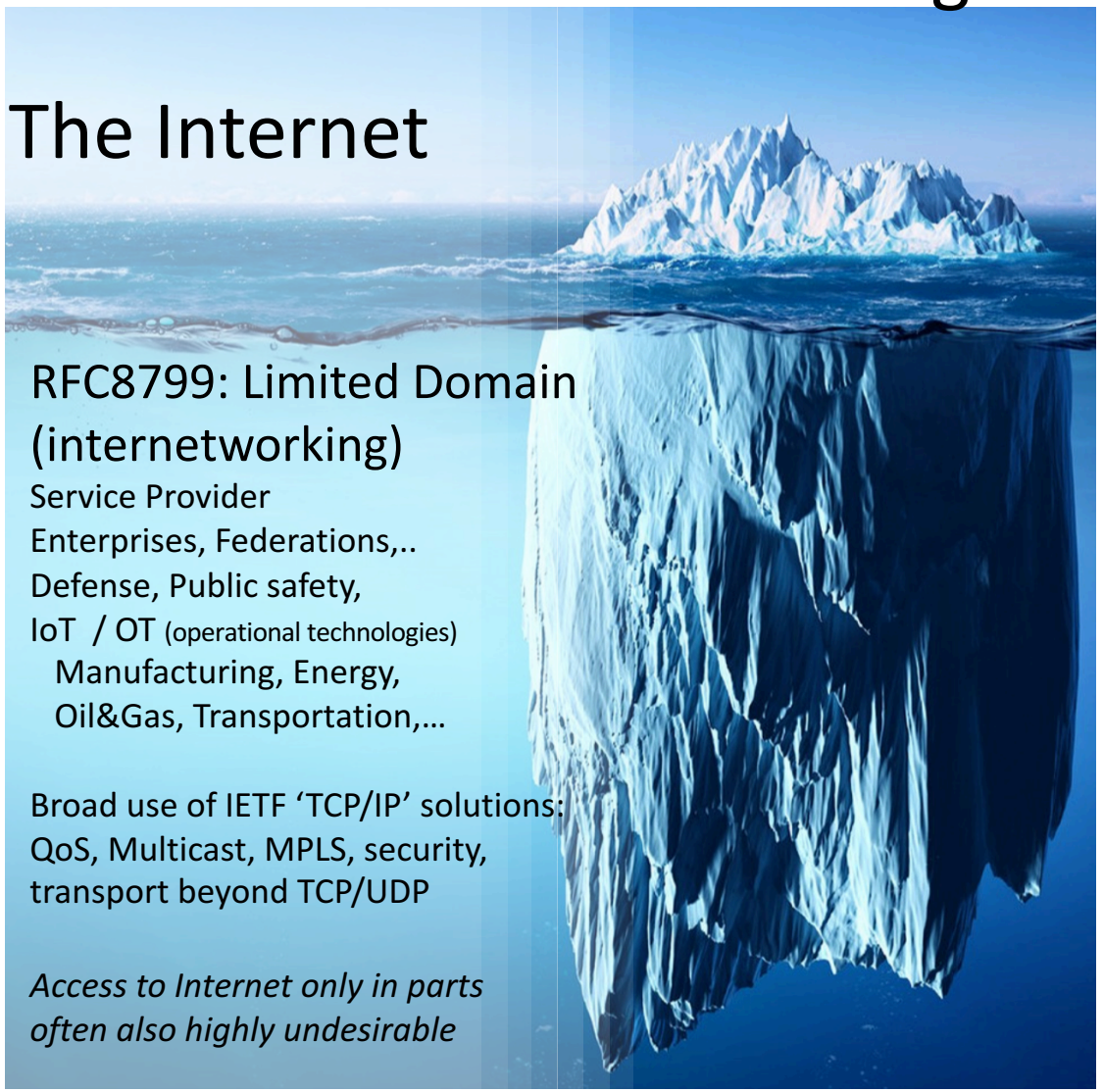
- Solution introduces additional proposal befits for known?! optimization/extensibility/unification goals
  - Efficient/flexible traffic steering (best of SR-MPLS / SRv6 ?!) – e.g. see CRH'
  - Efficient/flexible "programming" (SRv6++ ?)
  - New semantics without new separate network protocol for each
    - BIER, CCN/CDN, satellite,…
    - Remove duplication of infra efforts across multiple L3: SRv6, SR-MPLS, BIER, ...

- Research functionality ?!
  - Explore what else this scheme could do
  - Satellite addressing, security addressing, …
  - Author has not explored...

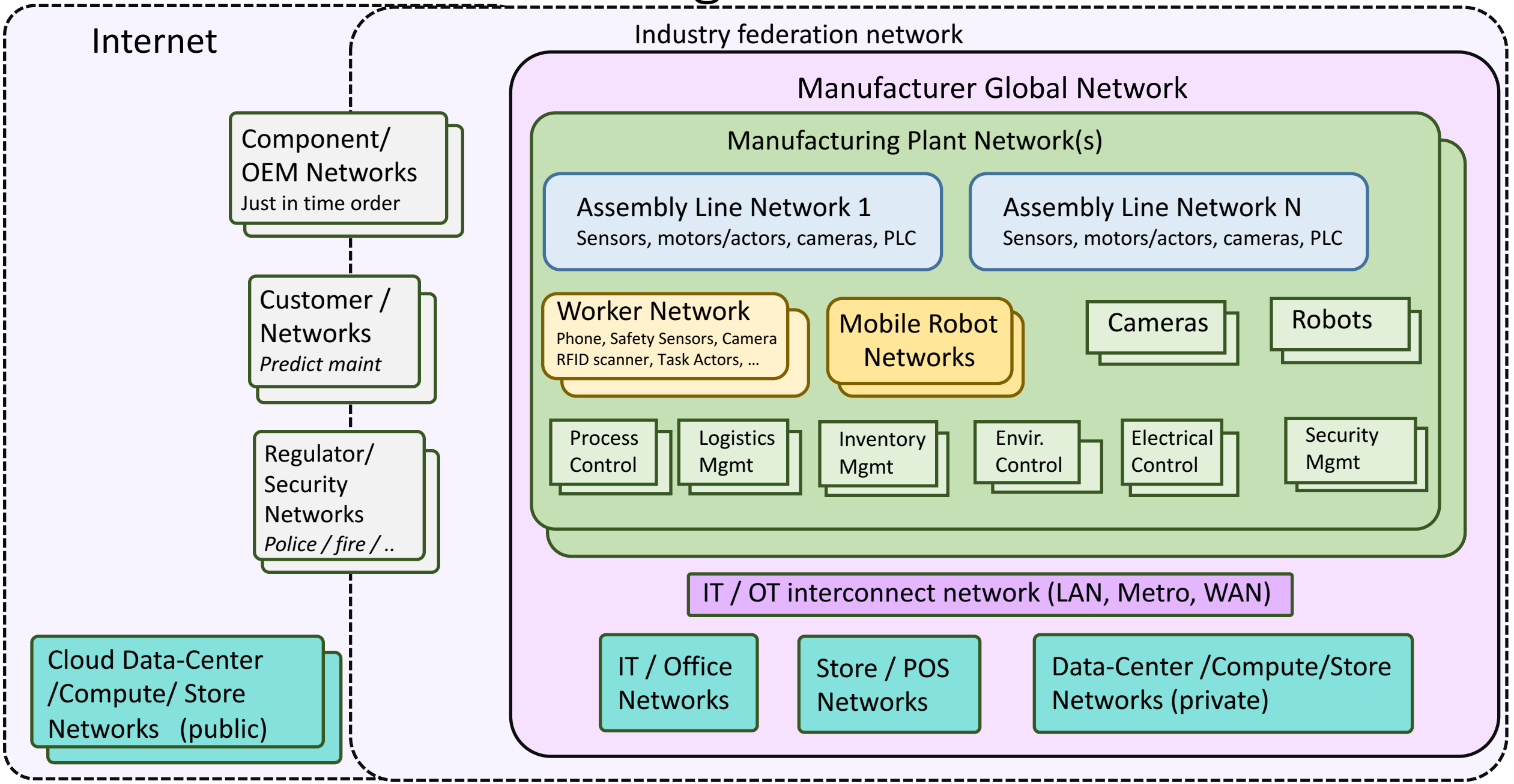
Problem: Arbitrary composition of networks with independent address spaces

- 90% TCP/IP systems are NOT "on the Internet"
  - Most not even "connected to the Internet" (heavy filtering / application gateways)
- Many challenges with IPv6 addressing.
  - PD – complex multihoming e.g.: IPv6 (S,D) forwarding, SP change causes renumbering, undesirable long prefix/host-parts (compression, management)
  - PI – loss of aggregation, admin/cost of assignment/delegation, limited space for extensibility (128 – 64 – prefixlength)
  - ULA – not "generic" addresses Worse than RFC1918 ?!
  - Site-local addr retired: rfc3879 (not disagreeing with that doc, but we still have to solve scoping of same addresses for link-local and multicast, so instead of giving up, we could have expanded) But it is "Not required for Internet"

The IETF Protocol Iceberg



The Smart Manufacturing "Network of Networks"



Inspiration and challenge:
Generic addressing, network composition via NAT

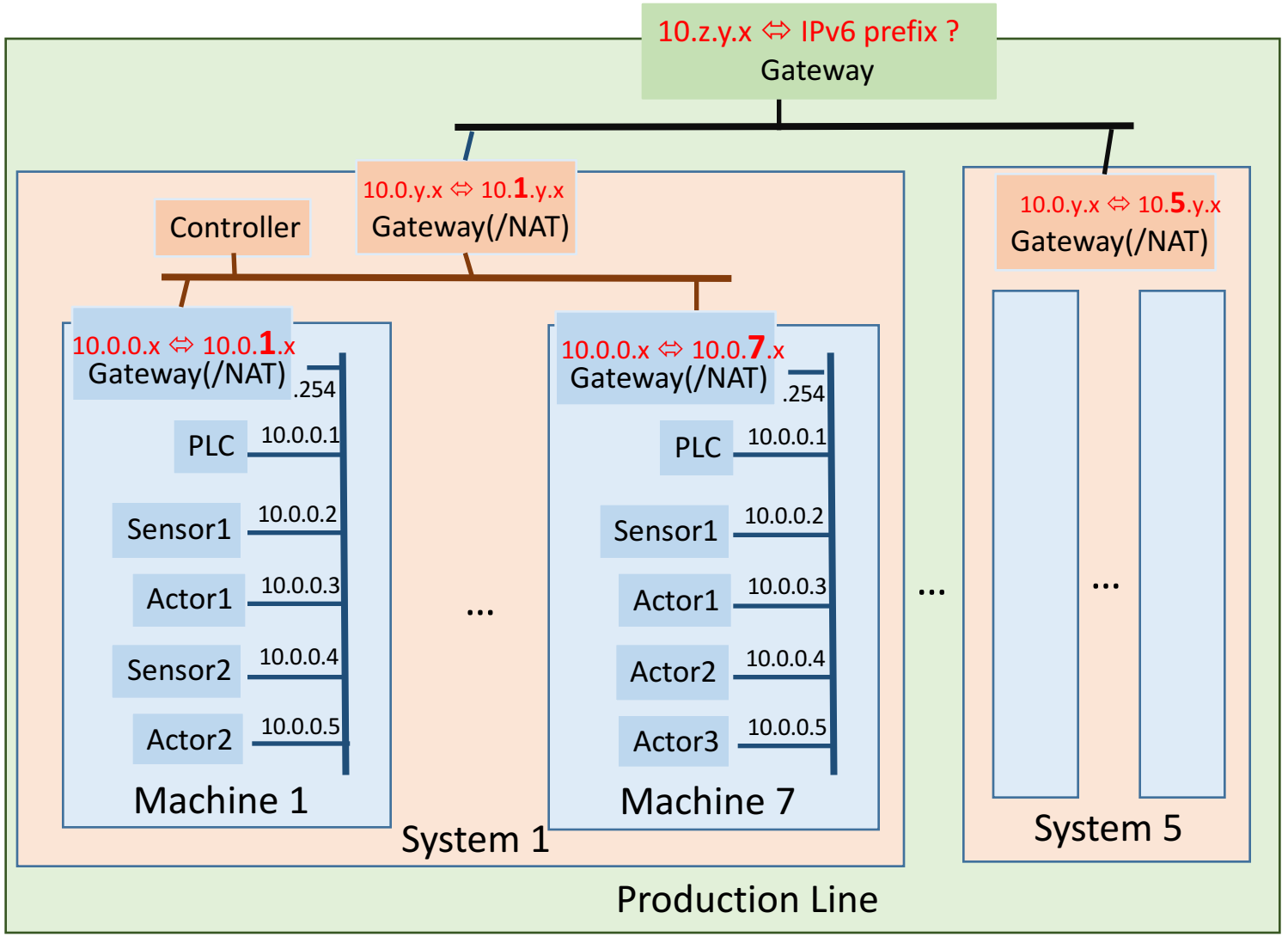
Global scope IPv4/Pv6 addresses should not be “burned in”. You don’t own them. You just lease them.

RFC1918 allows to manufacture equipment with “generic” addresses. Addresses you own forever. But must use NAT to interconnect.

Two tiers of composition can very easily be done with two tiers of stateless static NAT.

IPv6 worse ?! ULA (IPv6) not equivalent to RFC1918. Supposedly no NAT. Stochastical prefix collision. Still have only 16 bit “Subnet ID” to play with, same as Net 10.

Not just industrial issue.



FA-IINAS: A simple scenario (0): Two separate networks

Addresses have variable length

- Address lengths % 4 = 0
- Prefix lengths % 4 = 0
- Adresses and prefixes can be written as hex-digit number
- "." in address just for visualization

Address allocation:

- Assign address-prefix(es) to every node

- **No assigned address-prefixes may overlap**

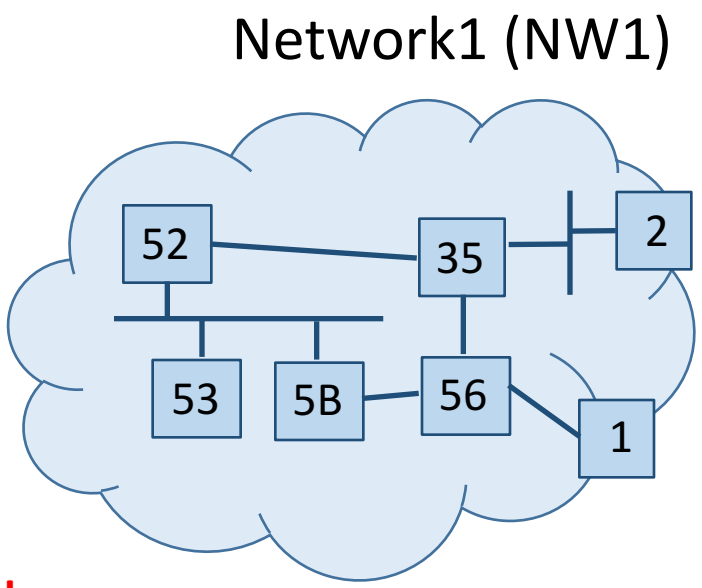
  every node "owns" any address equal to or longer than its assigned address-prefix.

- Every nodes address-prefix is routed in the network IGP
  - prefixes are "location" independent

FA-IINAS: A simple scenario (0): Two separate networks

Addresses have variable length

- Address lengths % 4 = 0
- Prefix lengths % 4 = 0
- Adresses and prefixes can be written as hex-digit number
- "." in address just for visualization



Address allocation:

- Assign address-prefix(es) to every node

- **No assigned address-prefixes may overlap**

  every node "owns" any address equal to or longer than its assigned address-prefix.

- Every nodes address-prefix is routed in the network IGP
  - prefixes are "location" independent

FA-IINAS: A simple scenario (0): Two separate networks

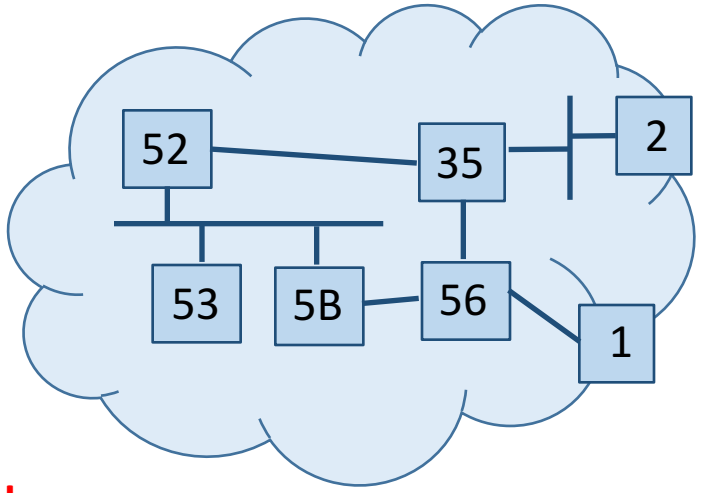
Addresses have variable length
- Address lengths % 4 = 0
- Prefix lengths % 4 = 0
- Adresses and prefixes can be written as hex-digit number
- "." in address just for visualization

Address allocation:
- Assign address-prefix(es) to every node

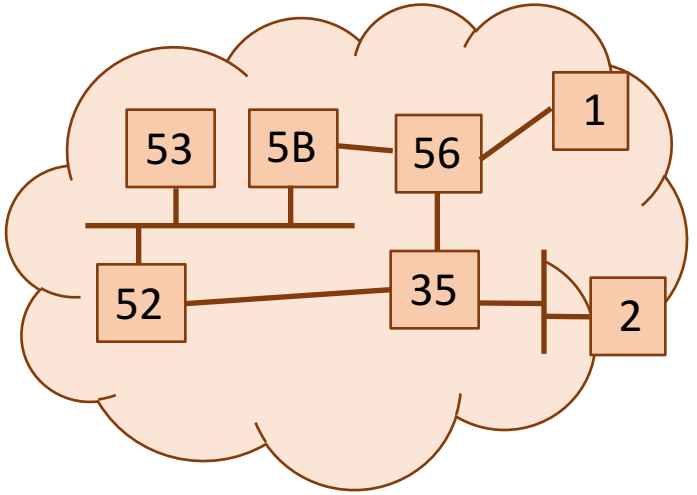
- **No assigned address-prefixes may overlap**
  every node "owns" any address equal to or longer than its assigned address-prefix.

- Every nodes address-prefix is routed in the network IGP
  - prefixes are "location" independent

Network1 (NW1)



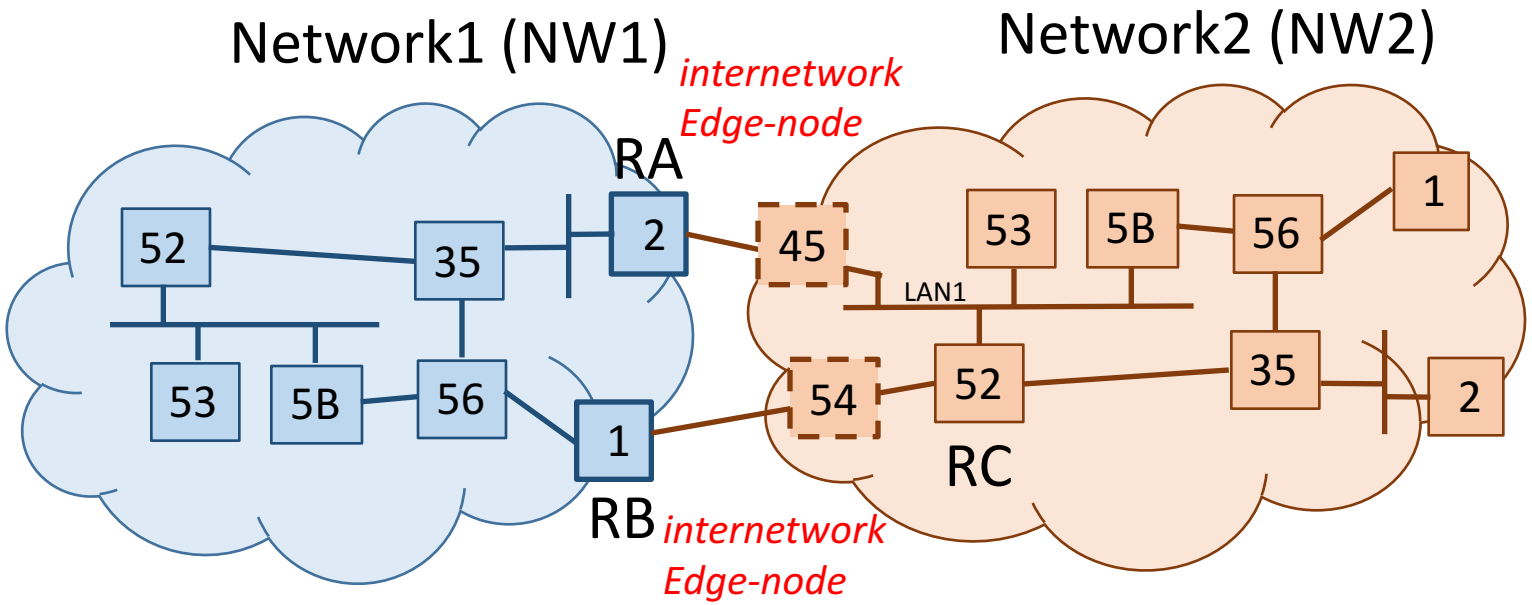
Network2 (NW2)



May have been built from same template as Network 1.
Aka: addressing may overlap.

Internetworking Q: How do we connect Network 1 and Network 2 ?

FA-IINAS: A simple scenario: (1) connect the networks

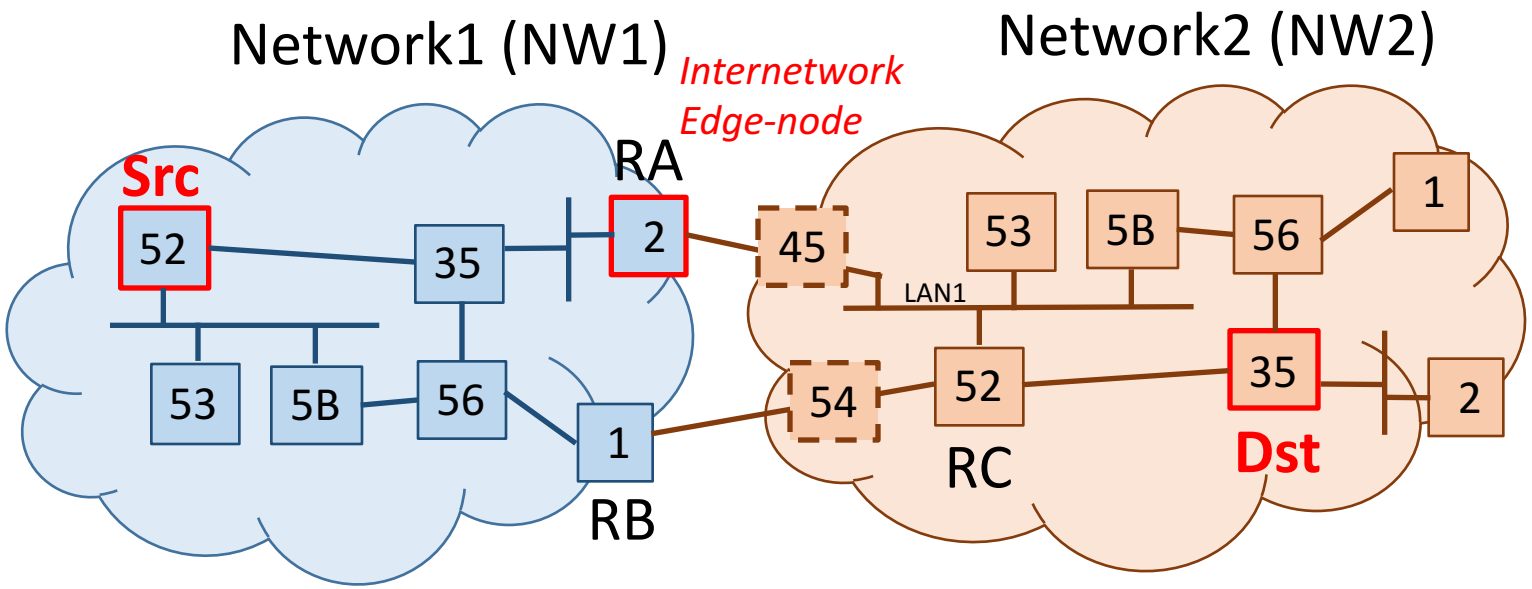


Network1 wants to connect into Network 2, so it uses NW1:2 (RA) and NW2:1 (RB) to do this.
Aka: Only one network may need to care about the connection and configure something on its routers (NW1)

Add link from RA to NW2: LAN1
Configure RA for this link to be in separate (namespace/IGP) Network NW2 with address prefix 45

Add link from RB to RC
Configure RB for this link to be in separate (namespace/IGP) Network NW2 with address prefix 54

A simple scenario: (2) send a a packet



Packet from NW1:52 to NW2:35: **DstAddr**: **2.2.1.35**, **SrcAddr**: **52**

Packet forwarded by NW1:52 via NW1:35 to NW1:2 (RA) because of NW1 IGP

RA examines function after its prefix 2/4: **2.2.1.35**: – forward to separate network

2: function to forward into different network, **1**: parameter for function: Network 2 via LAN1

RA rewrites packet DstAddr and (optional) SrcAddr to send it into NW2:

Strip examined function prefix from DstAddr: **2.1.1.35 -> 35**

Prepend return prefix to SrcAddr: **52 -> 45.2.2.52** - "2" indicates link into NW1 for RA.

RA sends packet into NW2 link, next-hop (52 on LAN1) from IGP.

A simple scenario: Review

- Only additional edge-node functionality/configuredion on nodes IN ONE NETWORK required to interconnect multiple networks
- Simple forwarding for edge-node functionality

  Longest mask lookup followed by (NEW): strip DstPrefix, prepend SrcPrefix:

  RA:NW1 forwarding table:

  2.2.1/12 -> DstAddr: strip /12 prefix, SrcAddr: prepend 45.2.2, "recirculate" into NW2 forwarding table

  RA:NW2 forwarding table:

  45.2.2/12 -> DstAddr: strip /12 prefix, SrcAddr: prepend 2.2.1, recirculate into NW1 forwarding table

- Stateless, per-prefix address rewrite

  Best scalable/performance ?! Address rewrite option

- Works for arbitrary topologies of interconnected networks!

  As long as address fields are long enough to indicate inter-network path through edge-nodes

Functional Address semantics, generalized (example)

- Addr = function.{function…}
- function = (<semantic-prefix> | <node-prefix>) . <function-code>{.<parameters>}
- <function-code>{.<parameter>}:
  - 0.<next-protocol> - receive packet, pass up to <protocol>
    Eliminates IP next-protocol packet field
  - 1.<node-prefix> - source-routing. Remove prefix, steer packet to <node-prefix>
    Steering as in SR-{MPLS,v6}. Maybe add mor functions (loose, strict,…)
  - 2.<link>.<address> - internetwork edge function – as described
    Removes prefix, appends prefix to SrcAddr
  - <other>.<params> - any other programming, as from SRv6
    Prepend before any other “packet ejecting” (0,1,2) function codes when combining multiple functions
- <semantic-prefix>
  - Prefixes for non-unicast forwarding (Multicast, CDN, BIER,…)
  - Not further discussed here.

Naming / Address Resolution
how to know DstAddr, how to determine internetworking path?

- Often: We do not need other identifiers than addresses !
  - Node Address-Prefixes within networks are sufficient “Names”/”Identifiers” (persistent)
    - No need for additional name space ?! (machine 2 machine networks)
  - Path addresses in hierarchical networks can also be “Names”/”Identifiers”
    - Edge-node function codes can be fixed/anycast/designed

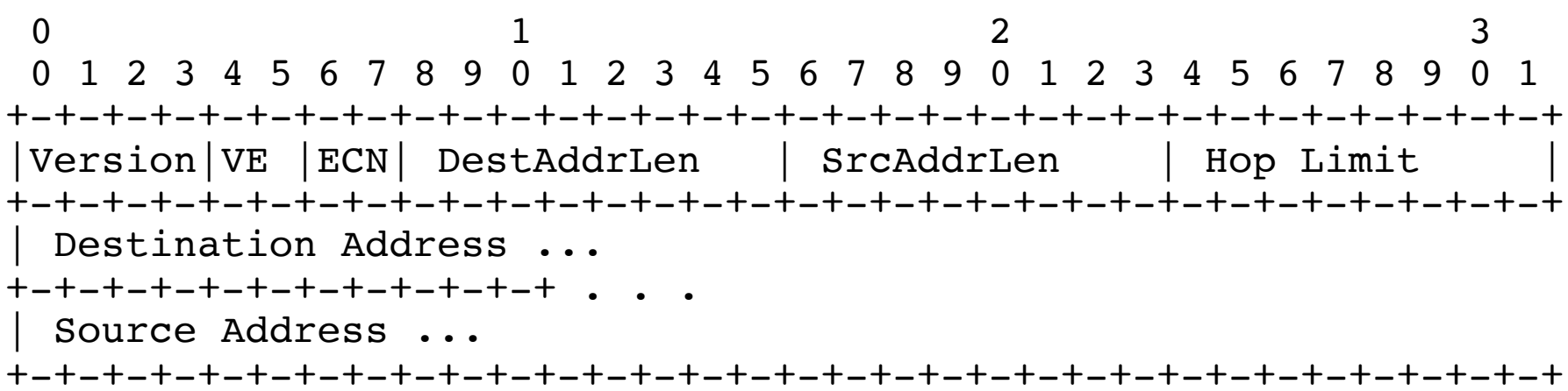
- More often: We have PCE/SDN-controllers/Orchestrators, many are humans
  - Viable approach in many internetworks (embedded, industrial)
  - Many internetworks have predefined internetwork interconnect topologies

- But also: Great opportunity for next-gen internetworking routing control plane
  - Global Network Names + path resolution control plane
    - Network name ~= AS ? BGP extensions ?
    - We have been doing this forever, even before IP (UUCP path table for email)
  - Path-coupling of name->address lookup with edge-node control plane agent
    - See e.g. 21 year old IPv4 Enterprise NAT multihoming (Yakov Rekhter et. al.)
  - More generic ideas in draft

FA-IINAS: Example header *(motivational!)*

32 bit header + address(es)

Strip & enhance IPv6 header:



Version = not 4 or 6 ! VE – 2 bit to identify addtl. E.g.: extension headers

Functionality covered by Destination Address:
Next protocol, SRH segment list/destination functions/TLV

Source address optional (length can be zero)
May not be used in limited domains operating like MPLS

Moved to TBD QoS/”Service-Level-Objectives” extension header (with more features)
DSCP, Flow Label - Not universally used, waste in base header.

Summary

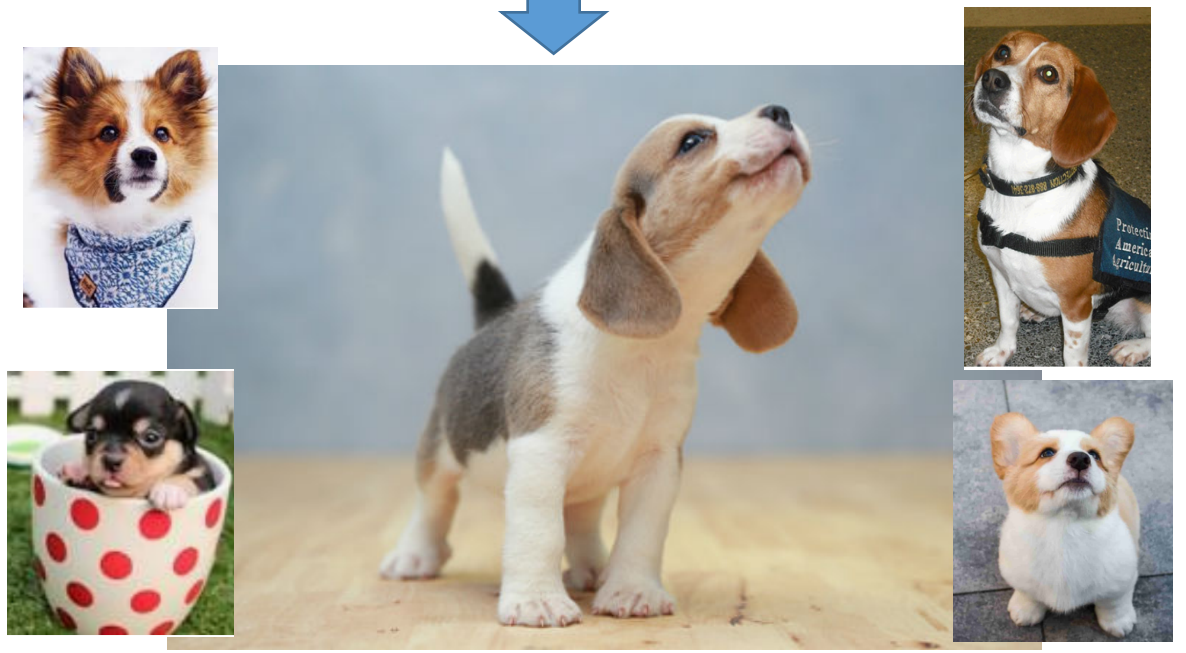
- IPv? needs to serve the other 90% (of networks/systems) better
  - Use-cases (rfc8799, internetworking, semantics, steering, programming, address spaces)…
- Make forwarding more useful – simpler but more extensible
  - Rely on known working high speed concepts – prefix lookup, stateless address rewrite
  - No per-feature extension header when we can do it better in the address
  - Customizeable, rich semantics defined via control plane (SR-TheNextGeneration ?!)
- TBD: Adoption strategy and interoperability
  - Do not repeat IPv4->IPv6 “kill & replace” goal.
  - Needs support for IPv4/IPv6 backward compatibility/integration
  - Prime issue: Host stacks
    - When you build it, how else could they come ? (“anything-new-over-UDP” ?!)

Domesticate addressing

NAT: IPv4, adressing, RFC1918, ULA,…
26++ IPv4/IPv6 transition (NAT) mechanisms:
https://en.wikipedia.org/wiki/IPv6_transition_mechanism

But benefits from **functional structures in IPv6** already (scopes/zones, unicast prefix multicast, RP,…)

Experiences with **address processing** of MPLS and **source-route** processing in SR-MPLS/SRH



FA-IINAS: **Multi-purpose functional address processing**

The End

- Check draft for many details / aspects that did not fit here
- E.g.: Interplanetary E.164 Addressing with FA-IINAS