2STiC

An introduction to future internet architectures

Caspar Schutijser and Joeri de Ruiter
SIDN Labs

Operator of the .nl TLD

- *Stichting Internet Domeinregistratie Nederland* (SIDN)
- Critical infrastructure services
  - Lookup IP address of a domain name (almost every interaction)
  - Registration of all .nl domain names
  - Manage fault-tolerant and distributed infrastructure

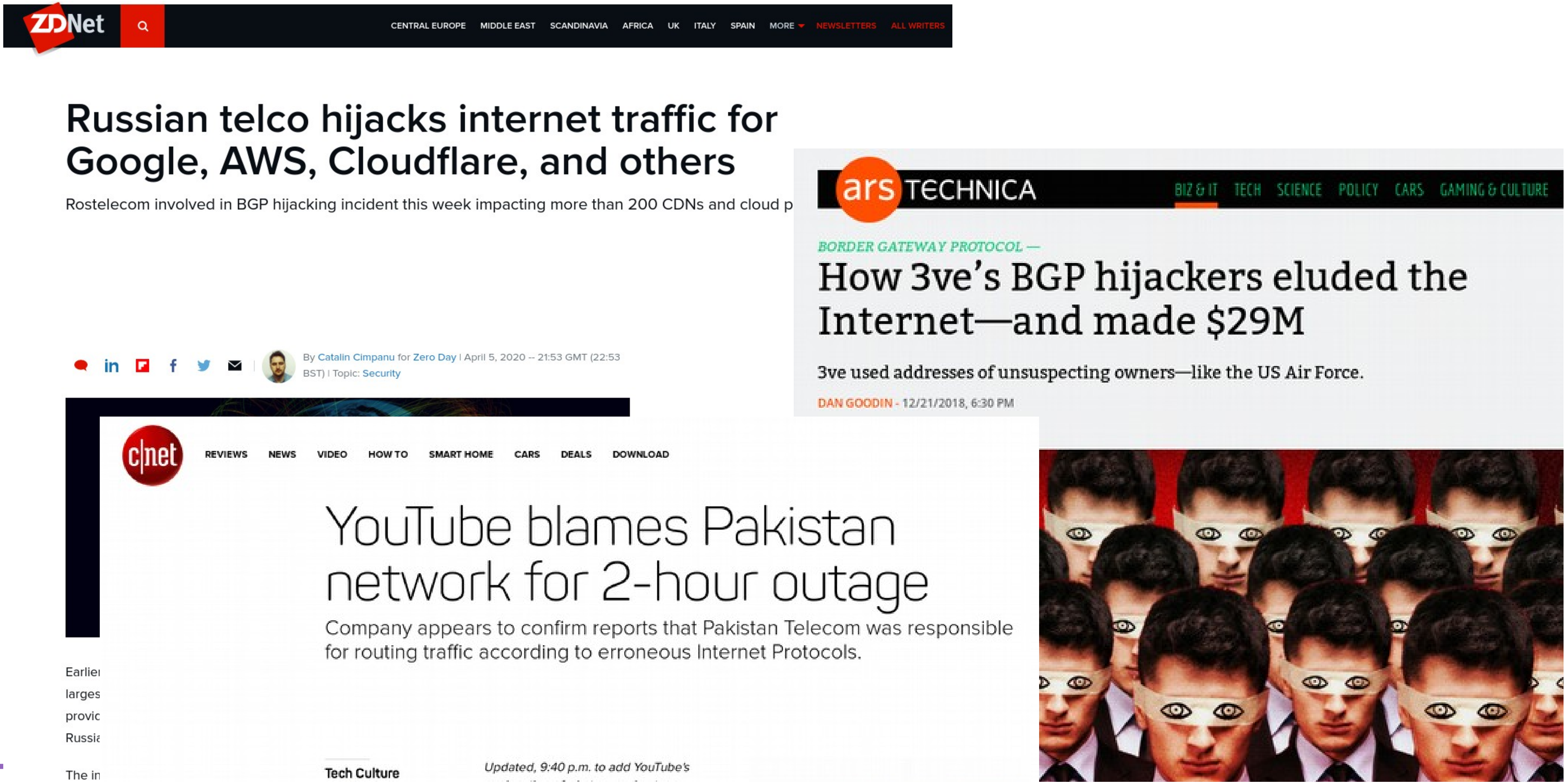
.nl = the Netherlands
17M inhabitants
6.1M domain names
3.4M DNSSEC-signed
2.5B DNS queries/day

SIDN Labs

- Goal: increase the trustworthiness of our society’s internet infrastructure
  - Measure, prototype, evaluate mechanisms that increase the trustworthiness of the Internet and for new internet infrastructures that complement the Internet
  - Reinforce the Dutch, European, and global research and operational communities
- Daily work: help operational teams, write open source software, analyze vast amounts of data, run experiments, write academic papers and tech reports, work with universities

The internet

- Started as small scale experiment
  - Nowadays a basic infrastructure
- Not designed with current usage in mind
  - For example, in the area of security
- Reactive approach to issues
- New infrastructures can offer solutions to this
  - Address issues fundamentally and pro-actively



Russian telco hijacks internet traffic for Google, AWS, Cloudflare, and others

Rostelecom involved in BGP hijacking incident this week impacting more than 200 CDNs and cloud p

By Catalin Cimpanu for Zero Day | April 5, 2020 -- 21:53 GMT (22:53 BST) | Topic: Security

Earlie
larges
provic
Russia

The in

BORDER GATEWAY PROTOCOL —

How 3ve's BGP hijackers eluded the Internet—and made $29M

3ve used addresses of unsuspecting owners—like the US Air Force.

DAN GOODIN - 12/21/2018, 6:30 PM

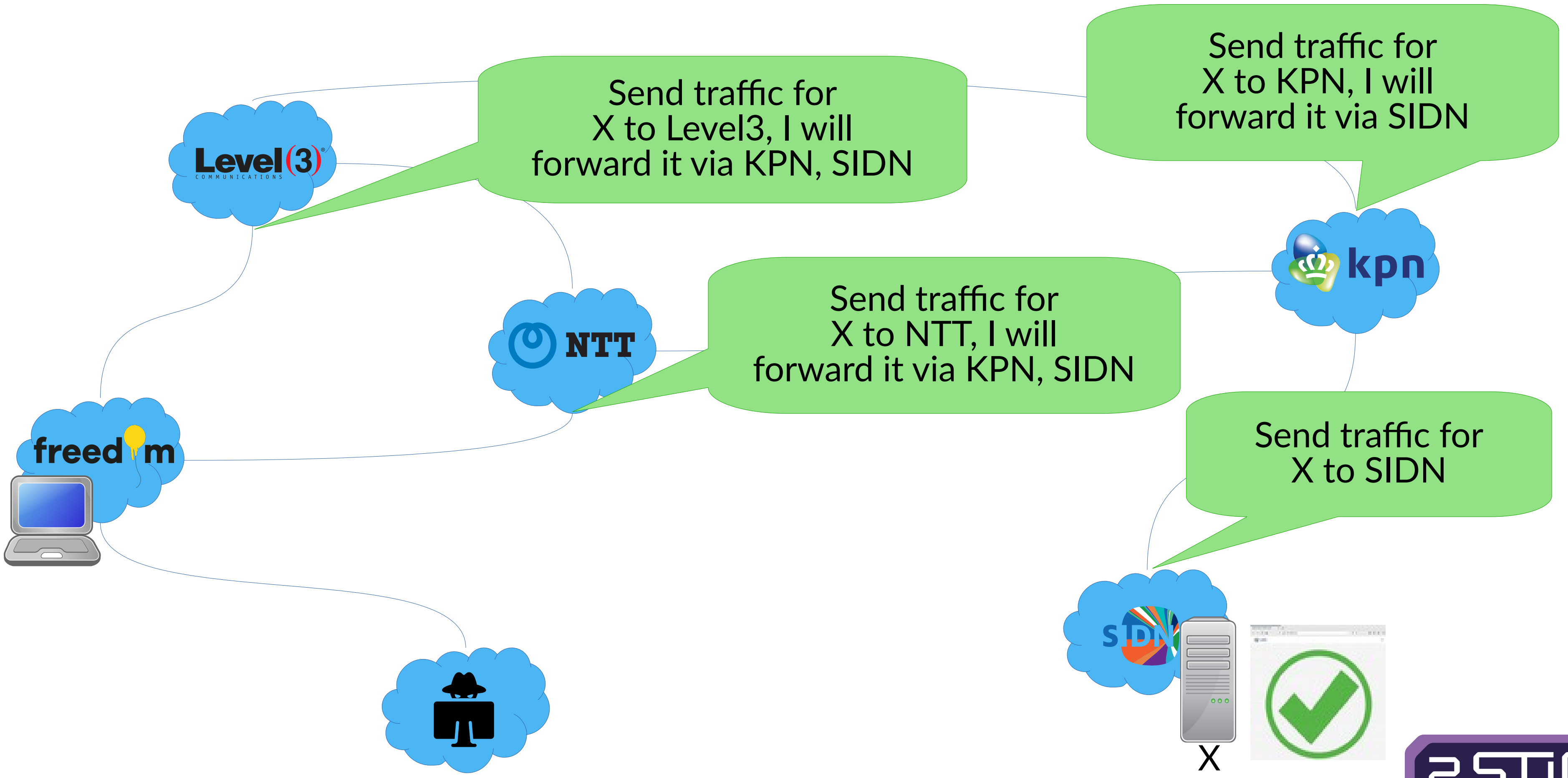
YouTube blames Pakistan network for 2-hour outage

Company appears to confirm reports that Pakistan Telecom was responsible for routing traffic according to erroneous Internet Protocols.

Tech Culture

Updated, 9:40 p.m. to add YouTube's

How does the internet work?



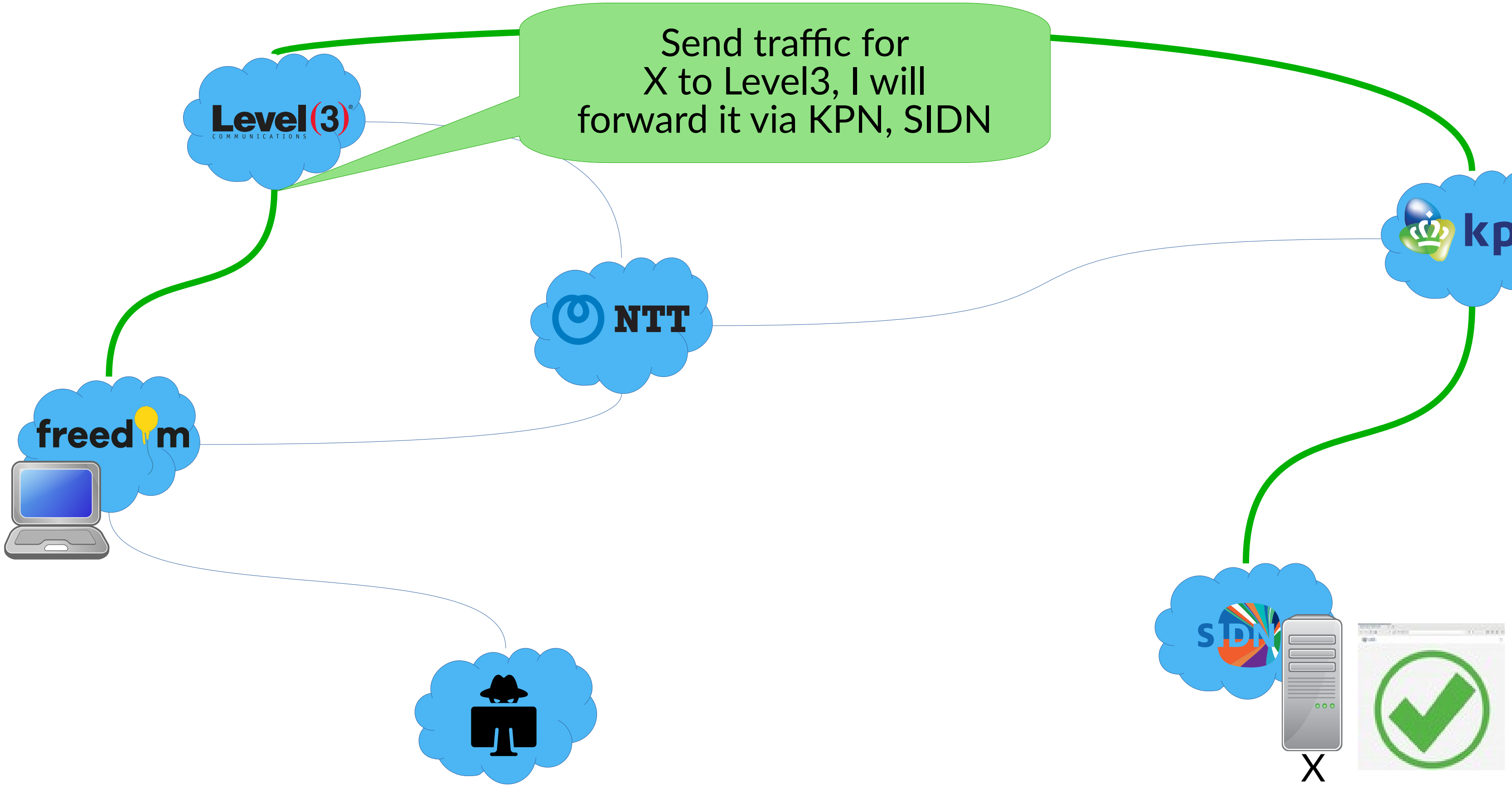
Send traffic for X to Level3, I will forward it via KPN, SIDN

Send traffic for X to KPN, I will forward it via SIDN

Send traffic for X to NTT, I will forward it via KPN, SIDN

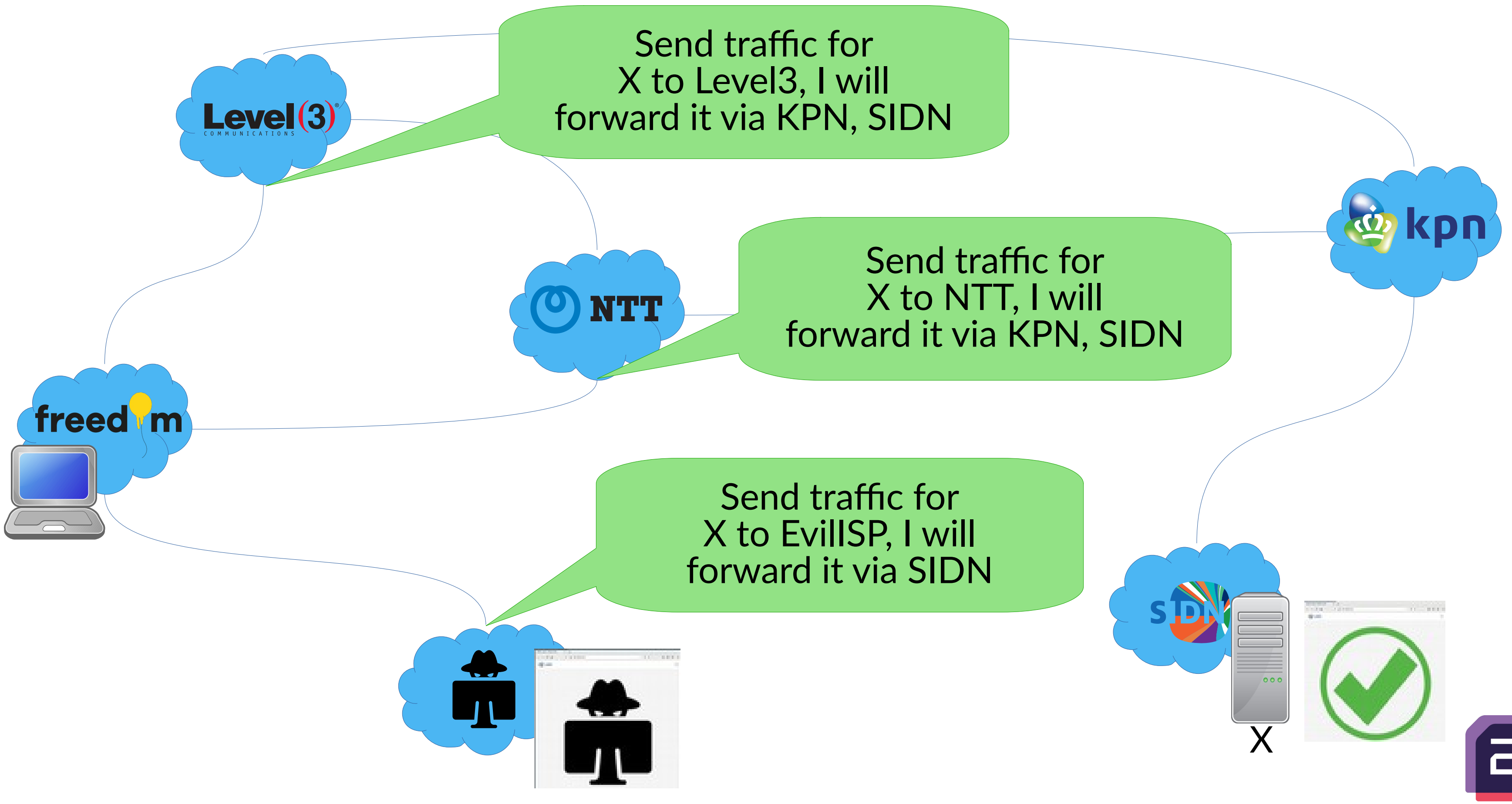
Send traffic for X to SIDN

How does the internet work?



Send traffic for X to Level3, I will forward it via KPN, SIDN

Route hijack

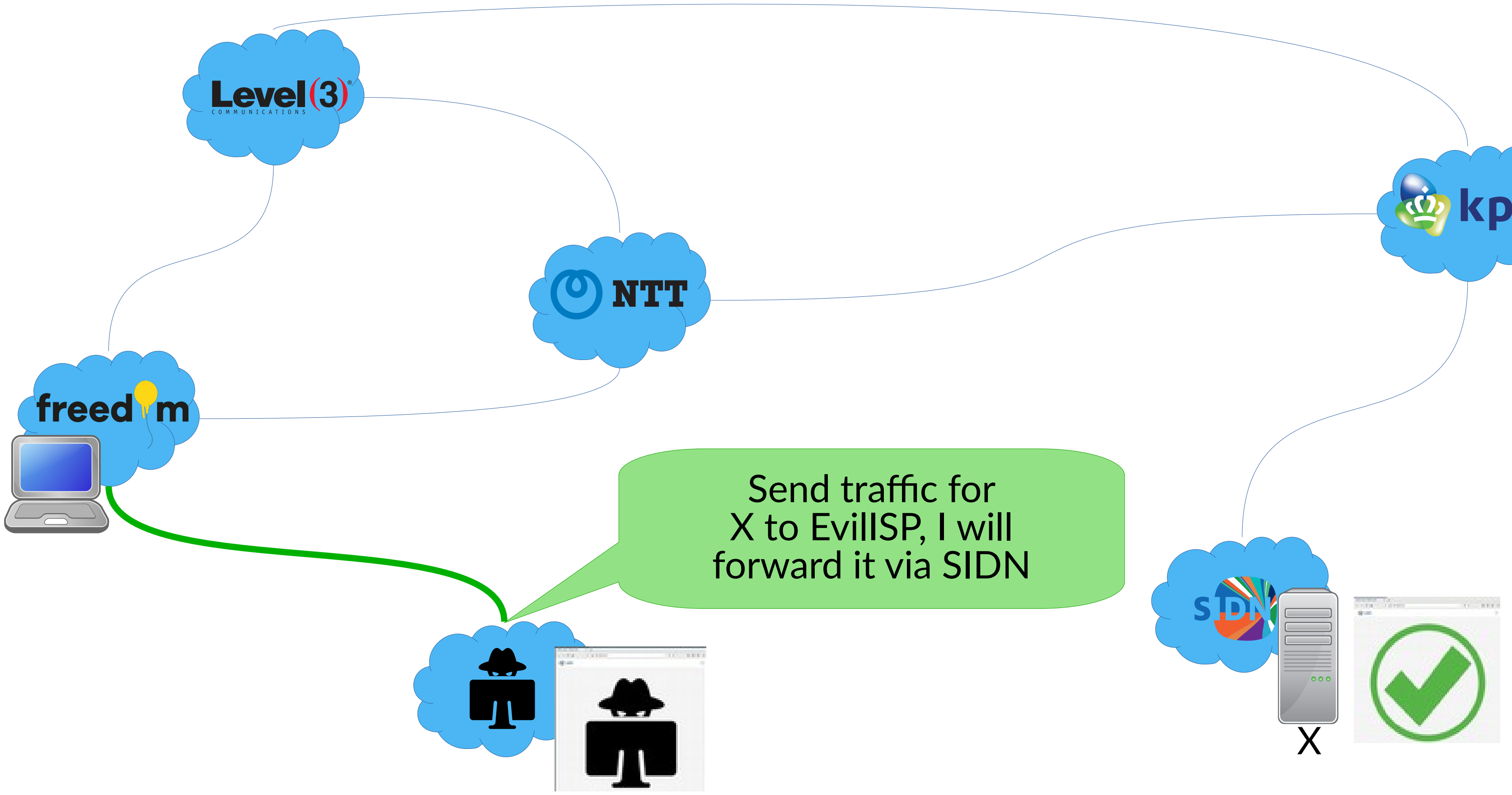


Send traffic for X to Level3, I will forward it via KPN, SIDN

Send traffic for X to NTT, I will forward it via KPN, SIDN

Send traffic for X to EvilISP, I will forward it via SIDN

Route hijack



Send traffic for
X to EvilISP, I will
forward it via SIDN

Security, Stability and Transparency in inter-network Communication

Put Dutch and European internet communities in leading position of secure, stable and transparent inter-network communication

2STiC

amsix

ndix

NLNETLABS

SIDN LABS

SURF

TU Delft

UNIVERSITY OF AMSTERDAM

UNIVERSITY OF TWENTE.

2STiC

- New applications have new security, stability and transparency requirements
  - More interaction with physical space (e.g., transport, smart grids, drones, remote surgery)
- Open programmable network equipment is becoming commercially available
  - Eases adoption
- Experiment with and evaluate emerging internet architectures
  - For example: SCION, RINA and NDN

SCION

SCION

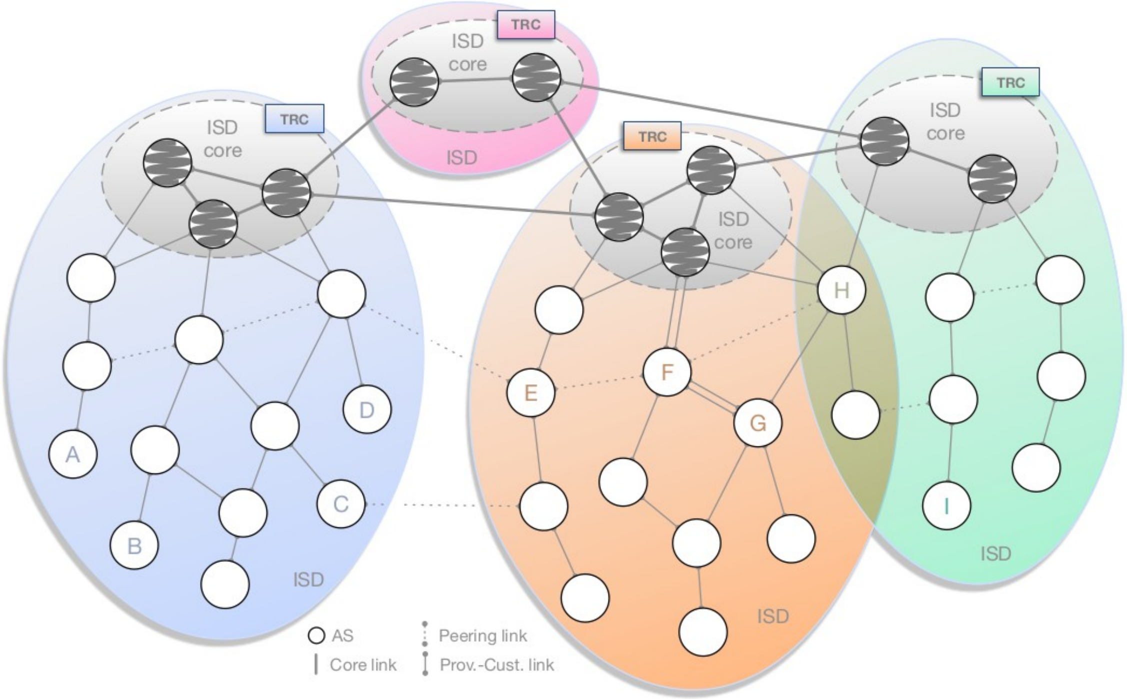
- Scalability, Control, and Isolation On Next-generation Networks
- New internet architecture
- Network Security Group, ETH Zurich
- Goal: improve security of inter-domain routing and isolation of compromise
- Scalability and security through Isolation Domains (ISDs)
  - Group of autonomous systems
  - E.g., per country or jurisdiction

SCION

- Security by design
  - Routes authenticated both in control and data plane
- Path-aware networking
  - Sender selects path
  - Enables, for example, geofencing
- Multi-path communication
  - Can be used, for example, for redundancy
- Existing application can still be used

Isolation domains

- Group of autonomous systems
  - E.g., per country or jurisdiction
- ISD core: ASes managing the ISD
- Core AS: AS part of the ISD core
- PKI organised per ISD
- Hierarchical control plane
  - Inter-ISD control plane
  - Intra-ISD control plane



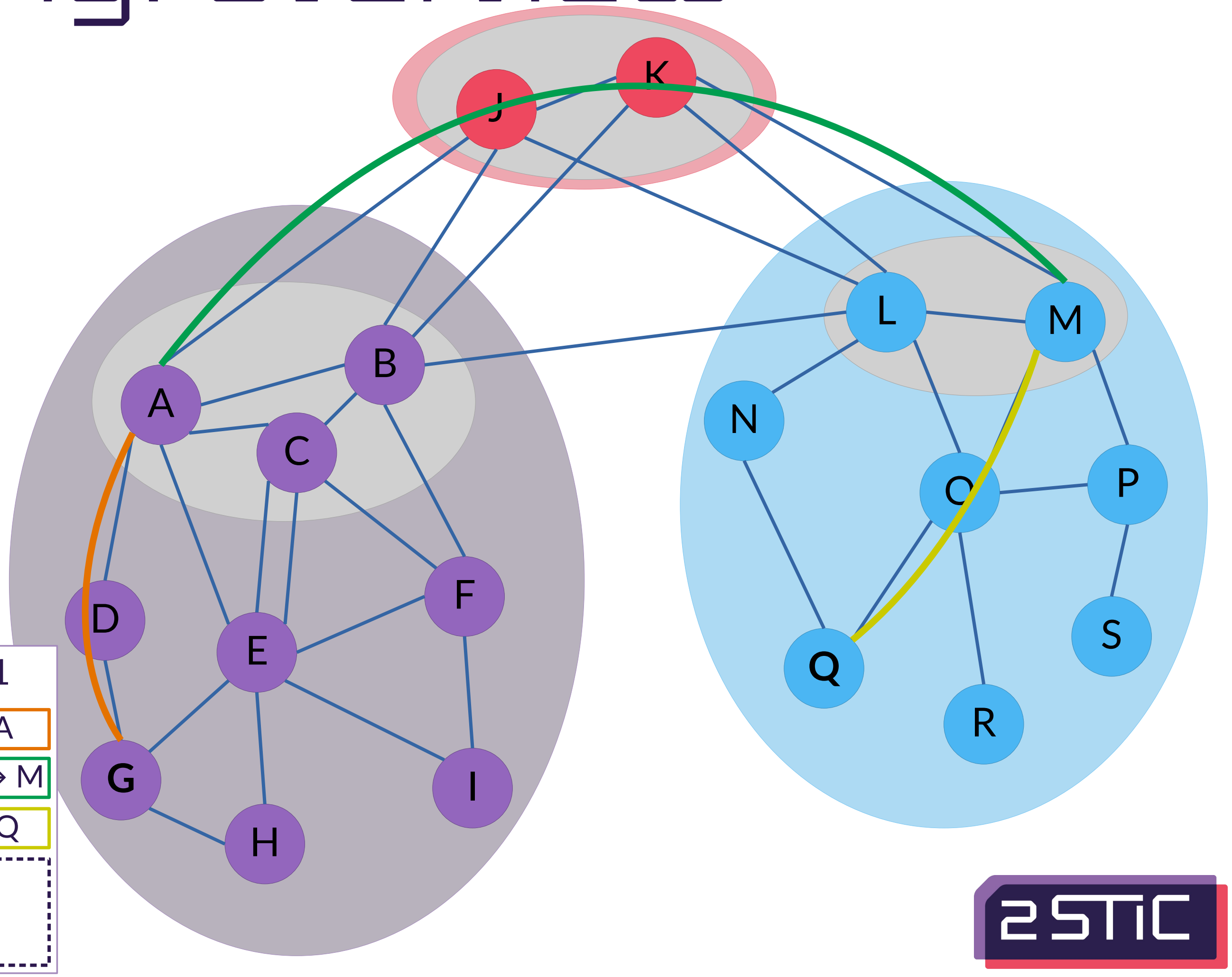
Source: The SCION Internet Architecture: An Internet Architecture for the 21st Century, Barrera et al., 2017

Routing: overview

- Control plane – finding end-to-end paths
  - Path exploration & registration
- Data plane – sending packets
  - Path lookup & combination
- Every AS runs a path server to provide path registration and lookup

Routing: overview

- Control plane
  - Construct and disseminate path segments
- Data plane
  - Combine path segments to path
  - Packets contain path
  - Routers forward packets based on path (stateless)



Packet P1

G → D → A

A → J → K → M

M → O → Q

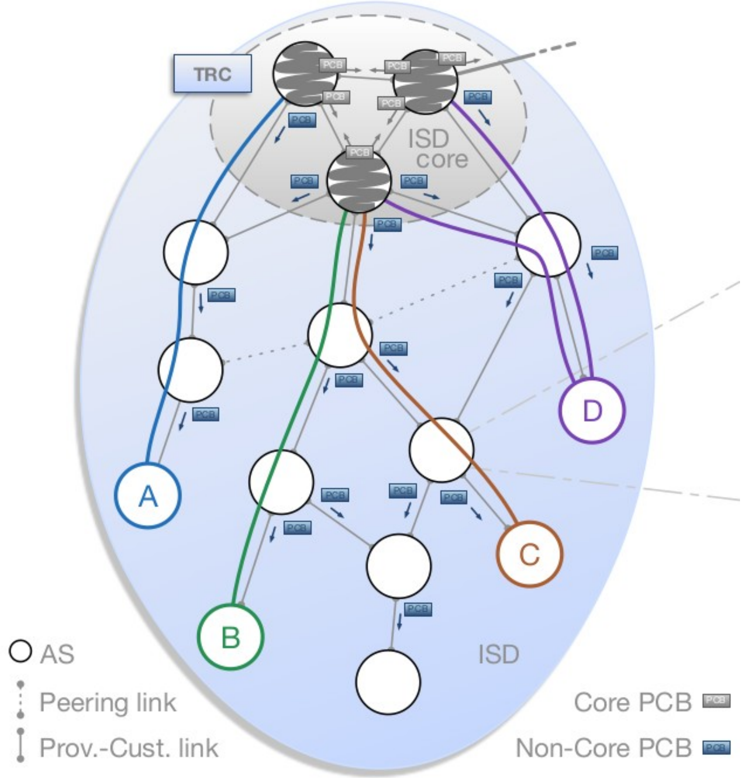
Payload

Control plane: path exploration

- Inter-ISD
  - Performed by core ASes
  - Flooding similar as with BGP
  - Less ASes involved (only core)
- Intra-ISD
  - Downstream multi-path flooding

Intra-ISD path exploration

- Path Construction Beacons (PCBs) sent downstream using multi-path flooding
  - Initialised by core ASes
  - Extended and forwarded by receiving ASes
  - Add incoming and outgoing interface and optional peerings
- Eventually all nodes know how ISD core can be reached
- Path registration
  - Preferred down-segments (path from core to AS) with path server in the core
  - Preferred up-segments registered with local path server in AS



Source: The SCION Internet Architecture: An Internet Architecture for the 21st Century, Barrera et al., 2017

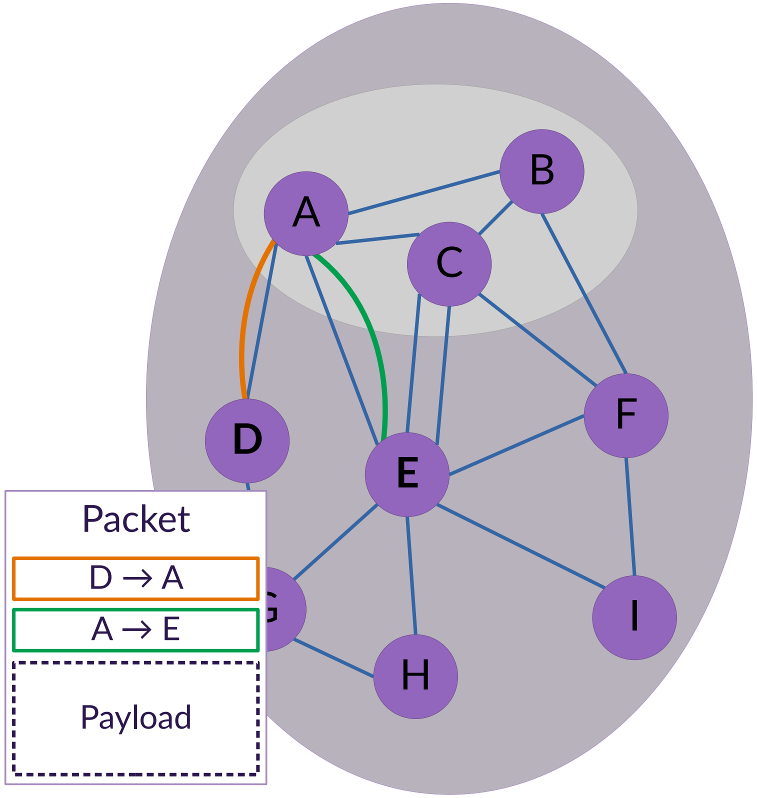
Path Construction Beacons

- Path Construction Beacons are signed by every AS along the path
  - Authenticated path
- Hop fields included that can be used to later select paths
  - Contain forwarding information
  - Contain MAC computed using hop field key
  - Only processed locally

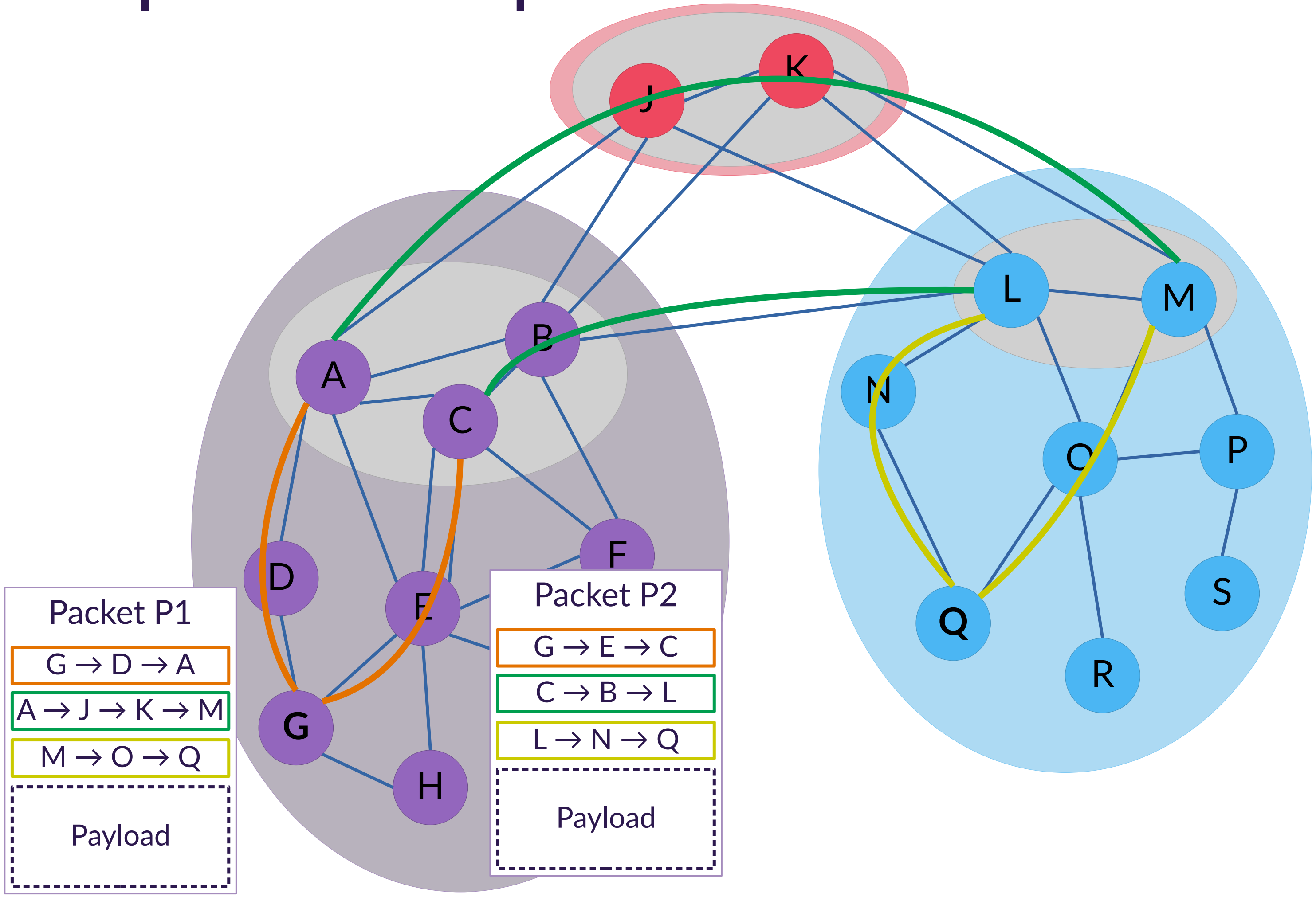
Data plane: path lookup

- Path construction performed by end hosts
- Request route to (ISD, AS) from local path server
- Local path server replies with
  - Up-path segments to local ISD core
  - Down-path segments in remote ISD from core to destination AS
  - Core-path segments needed to connect up-path and down-path segments
- End hosts pick and combine segments to determine path

Data plane: path combination



Data plane: path combination



Packet P1

G → D → A

A → J → K → M

M → O → Q

Payload

Packet P2

G → E → C

C → B → L

L → N → Q

Payload

Data plane: path combination

- Possible paths determined by
  - Up-stream AS, by deciding which PCBs to forward to where
  - Core AS, by offering path segments to path server in local AS
  - Local AS, by registering down-path segments with ISD core
  - Local AS, by offering path segments to clients
  - Clients, by combining path segments offered by local path server

Routing summary

- Path information included in packet headers
  - Corresponding hop fields included
  - No forwarding information necessary at routers
  - Packet-carried forwarding state (PCFS)
- Sender selects the path
  - Possible to use multiple paths
  - Fast failover
- Recipient address no longer used to route between autonomous systems
  - Only used by the destination AS
  - Local delivery is responsibility of destination AS

Security

- Path information authenticated in control plane and data plane
- Control plane
  - Beacons authenticated using digital signatures
  - No route hijacks
- Data plane
  - User selects path
  - Hop fields ensure only authorised paths possible

Security

- Address spoofing no longer possible on AS-level
  - Protects against reflection attacks
  - Reduces impact of DDoS attacks
- Hidden path
  - Path information not published
  - Can only be used by parties that know the relevant hop fields

Reliability and QoS

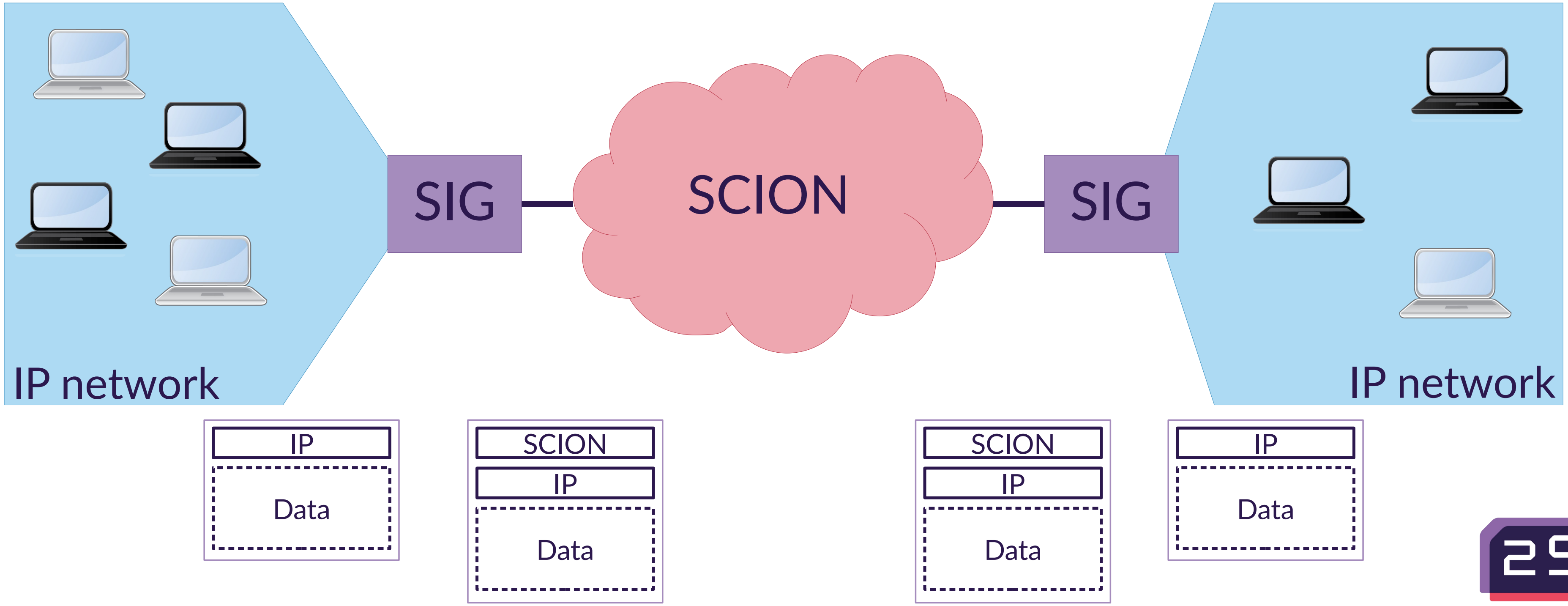
- Redundancy through use of multi-path communication
- Fast failover in case of link failure
  - No waiting for convergence
- Possible to add latency information to beacons
  - Path selection based on latency
- COLIBRI extension
  - Minimum bandwidth reservation

Deployment

- Open source implementation available
  - https://github.com/scionproto/scion
- International testbed SCIONLab
  - https://www.scionlab.org/
- Production network managed by spin-off Anapaya
- In use at banks, government and hospitals

Deployment

- Can be combined with existing applications using SCION-IP Gateway



SCION recap

- Security by design
  - Routes authenticated both in control and data plane
  - For example, no address spoofing
- Path-aware networking
  - Control over path that network traffic takes
- Improved reliability and QoS
  - Multi-path communication
  - Bandwidth reservation
- Existing application can still be used
  - SCION-IP gateway

RINA

Recursive InterNetwork Architecture

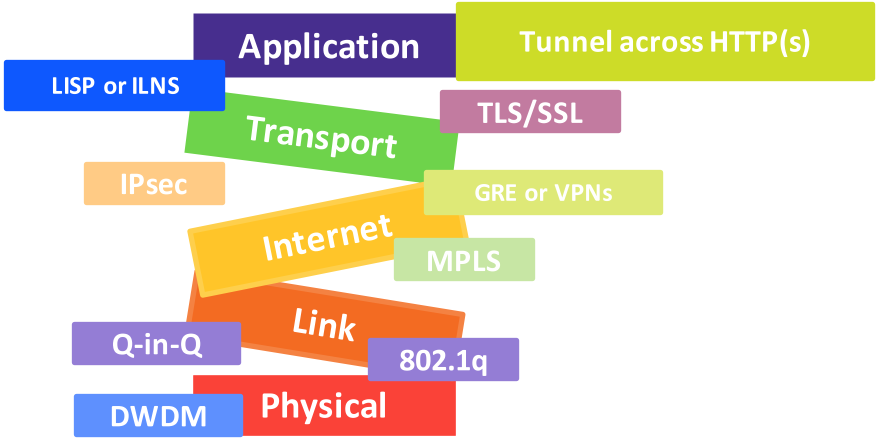
What are the **main flaws**
of today’s network architecture?

1. Structure: layers mess

TCP/IP RM - **Theory**

Application	network part of each application
Transport	data transfer services
Internet	logical communication
Link	adjacent communication
Physical	medium abstraction

TCP/IP RM - **Practice**



- **Fixed number of layers**, sometimes more needed between transport and application. -> Need concepts like “overlay”, “VPN”, “virtual networks”…
- Although the need for scope is clear (link, network, Internet, VPN …), **layers are organised as units of modularity**, with each layer providing a different function to each other.

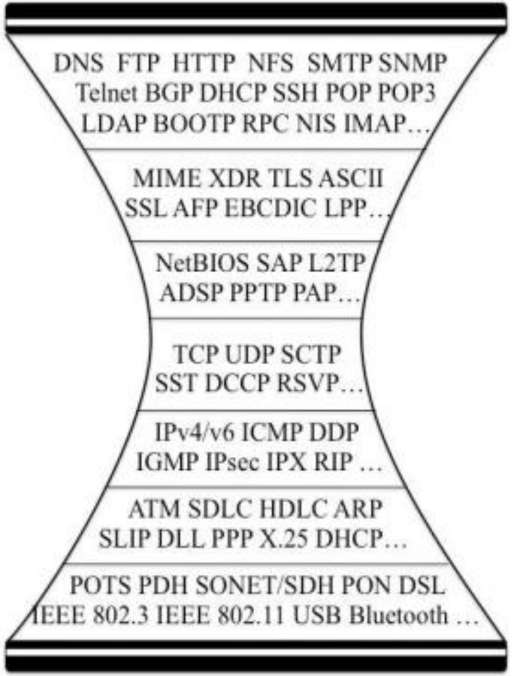
2. Protocols mess

- **Multiple protocols per layer**, even if each layer performs a function.

- Almost **each new use case** requires a **new protocol**.

- Flaws in the architecture (e.g. multi-homing, mobility) require **special protocols**.

Results in **protocol proliferation**!!!



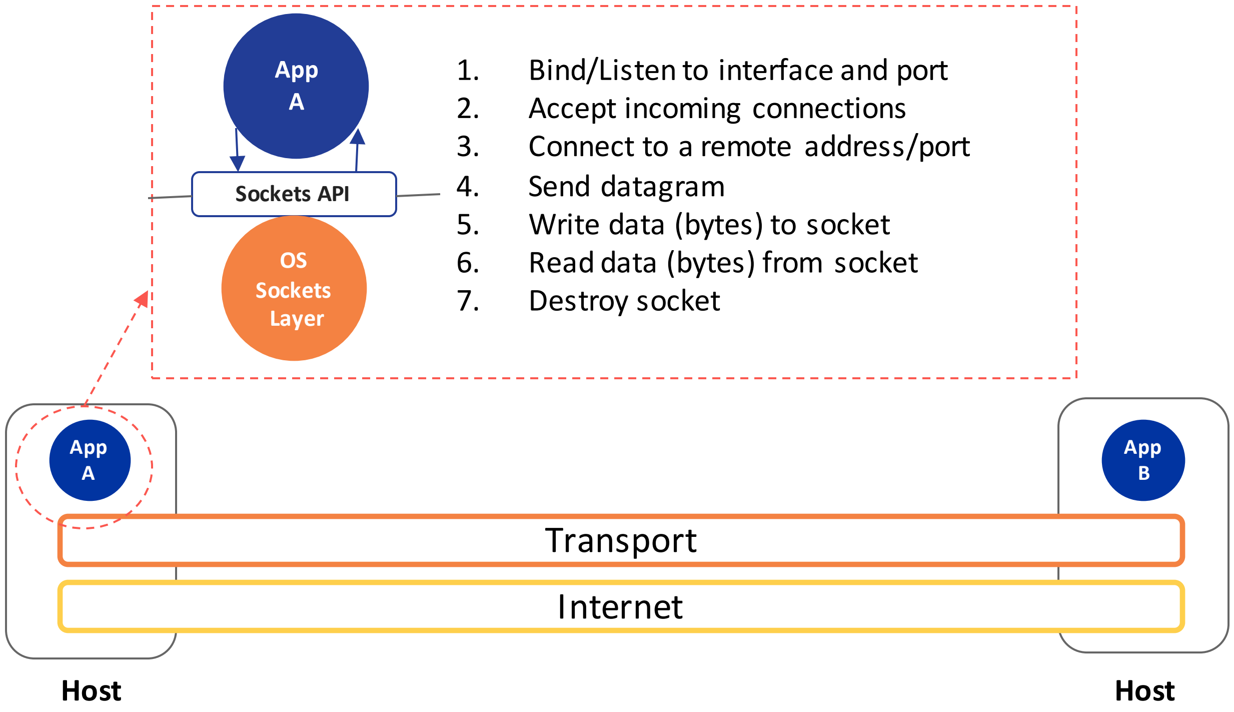
© "RINA Introduction"
Eduard Grasa. Arcfire 2017

2. Protocols mess

- Protocols are usually **independently designed from each other** (little commonality) in different standards developing organizations.

IEEE
Advancing Technology for Humanity
ICANN
W3C
Word Wide Web Consortium
iana
Internet Assigned Numbers Authority
ITU
IRTF
IAB
Internet Society
ISO
IETF
3GPP
A GLOBAL INITIATIVE
TIA
Telecommunications Industry Association

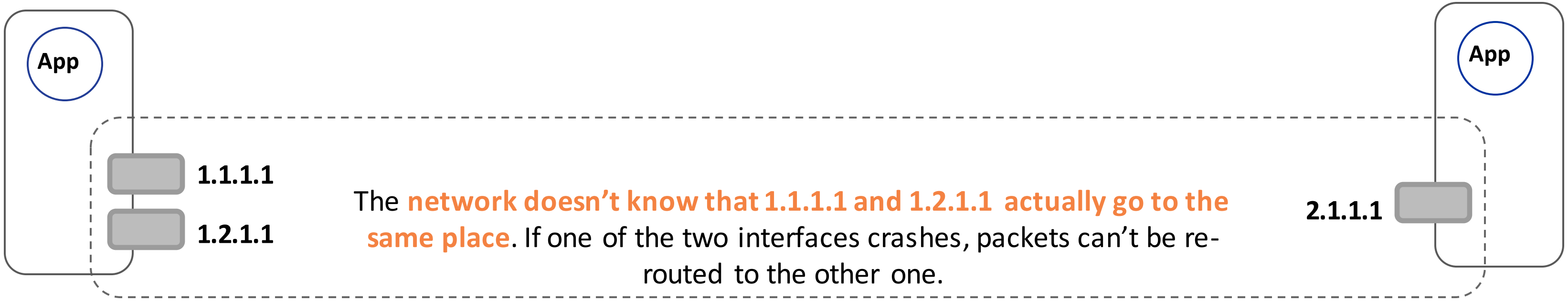
3. Application API



1. Bind/Listen to interface and port
2. Accept incoming connections
3. Connect to a remote address/port
4. Send datagram
5. Write data (bytes) to socket
6. Read data (bytes) from socket
7. Destroy socket

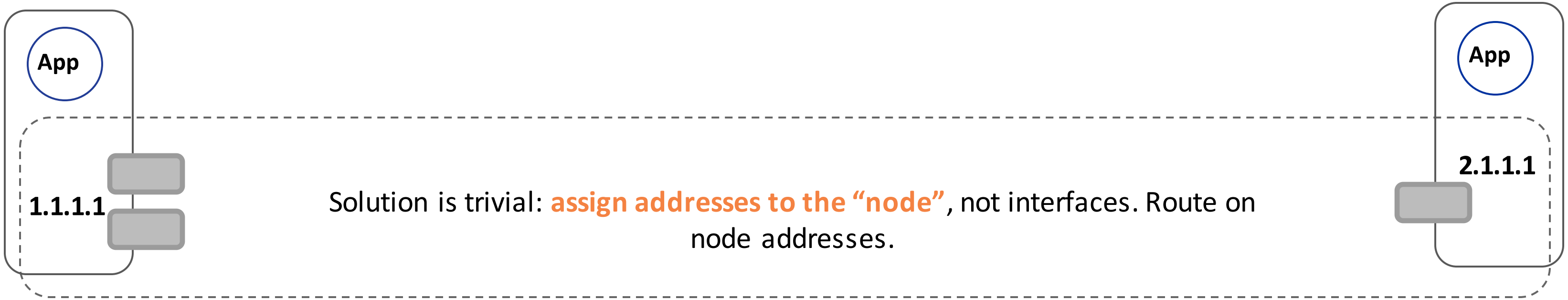
- Application must know about Transport Protocol and choose it
- Addresses are exposed to applications (security problem)
- No way to request QoS parameters (e.g. loss, delay…)

5. Multi-homing issue



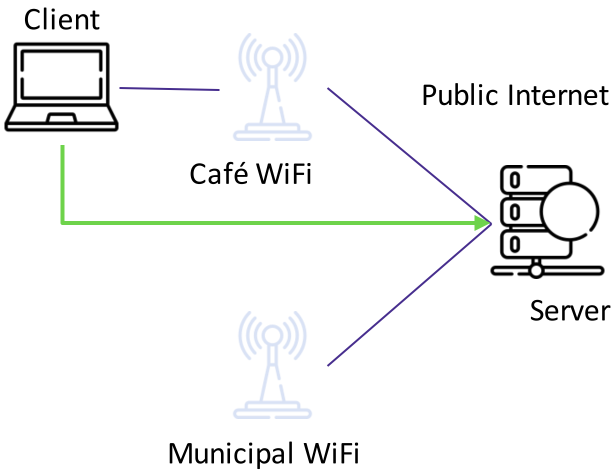
The **network doesn't know that 1.1.1.1 and 1.2.1.1 actually go to the same place**. If one of the two interfaces crashes, packets can't be re-routed to the other one.

A number of special protocols designed to partially deal with it: SHIM6, Multipath TCP, BGP (multi-homing at the AS level), SCTP.

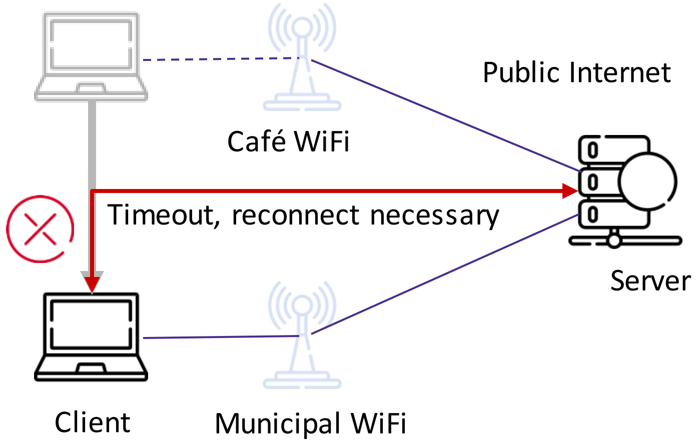


Solution is trivial: **assign addresses to the "node"**, not interfaces. Route on node addresses.

6. Mobility issue



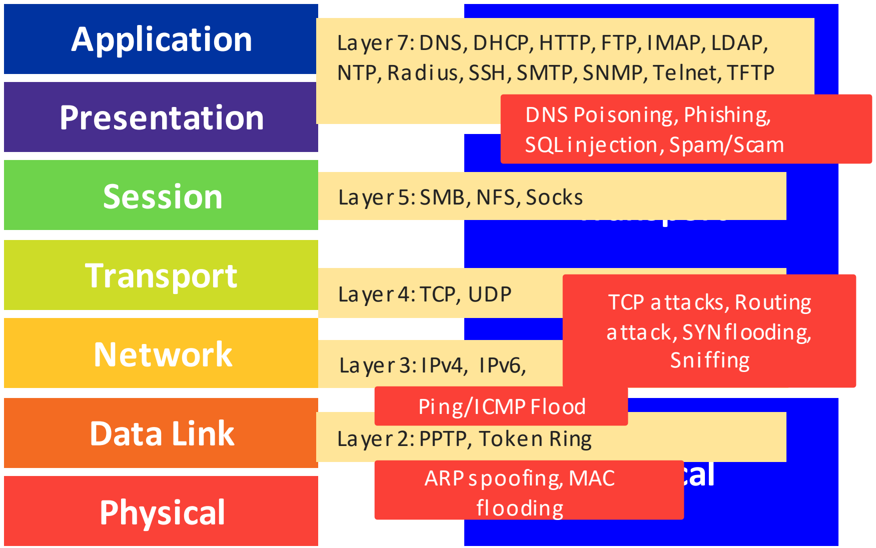
When I walk out of the café…



With TCP/IP: my connection is broken.

7. Security issue

Attacks on Different Layers



Application

Presentation

Session

Transport

Network

Data Link

Physical

Layer 7: DNS, DHCP, HTTP, FTP, IMAP, LDAP, NTP, Radius, SSH, SMTP, SNMP, Telnet, TFTP

DNS Poisoning, Phishing, SQL injection, Spam/Scam

Layer 5: SMB, NFS, Socks

Layer 4: TCP, UDP

TCP attacks, Routing attack, SYN flooding, Sniffing

Layer 3: IPv4, IPv6,

Ping/ICMP Flood

Layer 2: PPTP, Token Ring

ARP spoofing, MAC flooding

OSI Reference Model

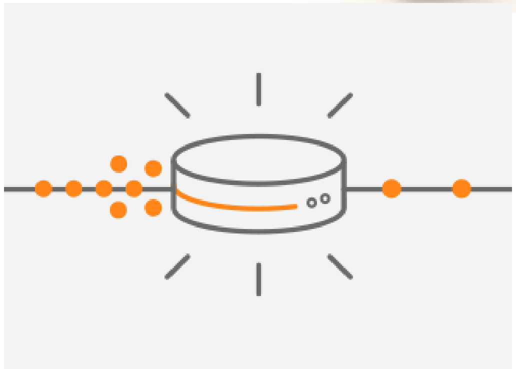
TCP/IP Model

- Security function for every protocol
- Use of well-known ports
- Network address exposed to applications

8. Quality of Service (QoS) issue

- QoS: guarantees w.r.t. loss rate, delay, for example
- Lack of a consistent QoS model across layers

9. Congestion Control issue



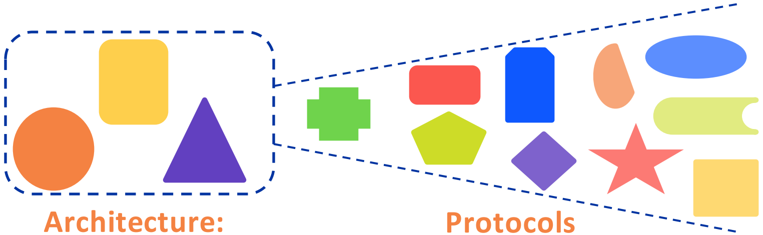
- Congestion control: avoid overloading network links/networks
- Only end-to-end congestion control loops
- Predatory (implicit) congestion control
- Homogeneous congestion control policies for heterogeneous networks

2 Introduction to RINA

Recursive InterNetwork Architecture

What do we **want** for a **better architecture**?

As much invariants as possible in the architecture, so that we can **minimize the number of protocols** and **maximize their commonality.**

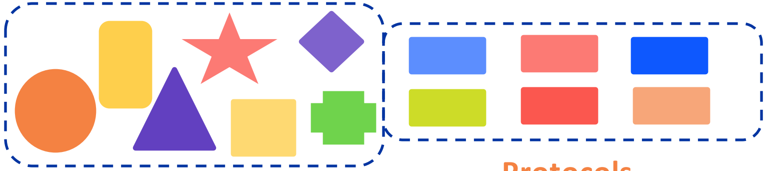


Architecture:
patterns, invariants,
building blocks, methods

Protocols

Today:

- Architecture has too little patterns/commonality, and they are a bit broken
- Too many protocols, too little commonality



Architecture:
patterns, invariants,
building blocks, methods

Protocols

Target:

- Architecture provides as much invariants as much invariants as possible
- Few protocols, sharing lots of commonality

Going back to the basics...

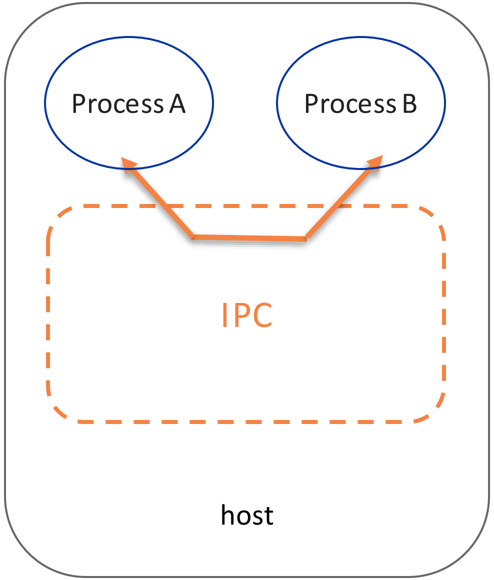
“Computer Networking is InterProcess Communication (IPC)”

– Robert Metcalfe, inventor of Ethernet, 1972

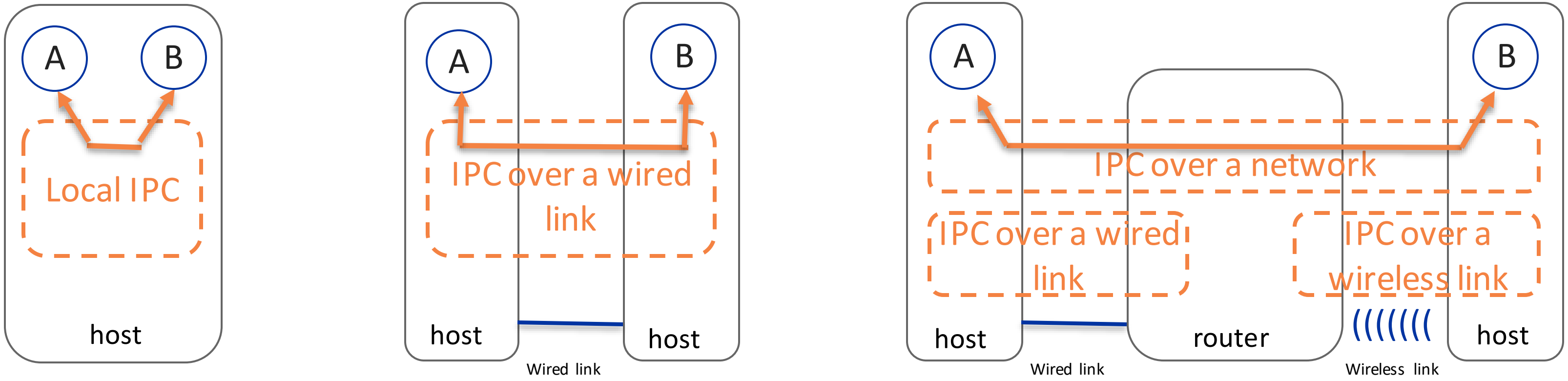
What is **Inter-Process Communication (IPC)**?

Some processes, executing at the same time in the operating system, may need to **cooperate** with each other: they will **communicate data**.

For example so that all the tasks can run smoothly without clashing with each other.

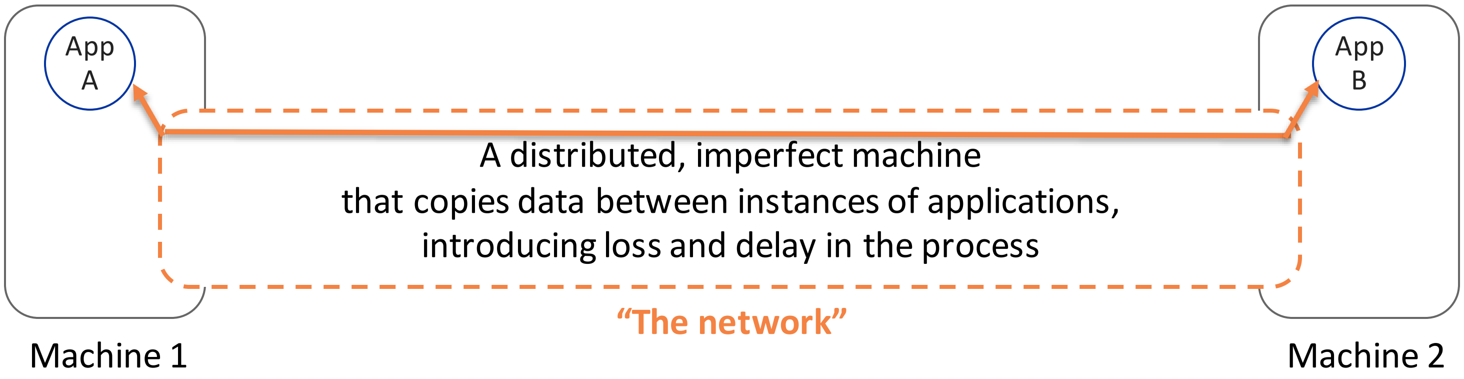


Example of communication between 2 application processes



Is there a difference when the processes are in several systems?

What is **the network**?



A distributed, imperfect machine
that copies data between instances of applications,
introducing loss and delay in the process

"The network"

Computer networking is Inter Process Communication (IPC)

© "RINA Introduction"
Eduard Grasa. Arcfire 2017

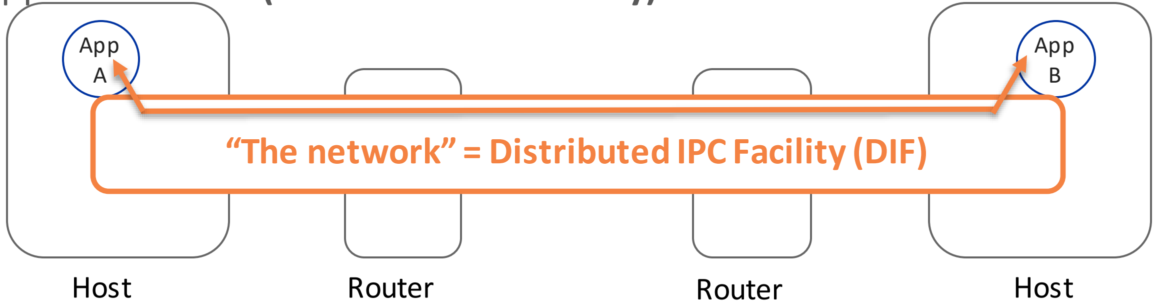
What is **the network**?

Provides IPC services, but what is it? Some hints:

- Executes in computers running operating systems (PCs, laptops, routers, sensors, smartphones, tablets, switches, etc.)
- Has instances distributed through many machines, exchanging information and collaborating
- Just like... the web, Skype, mail, etc.

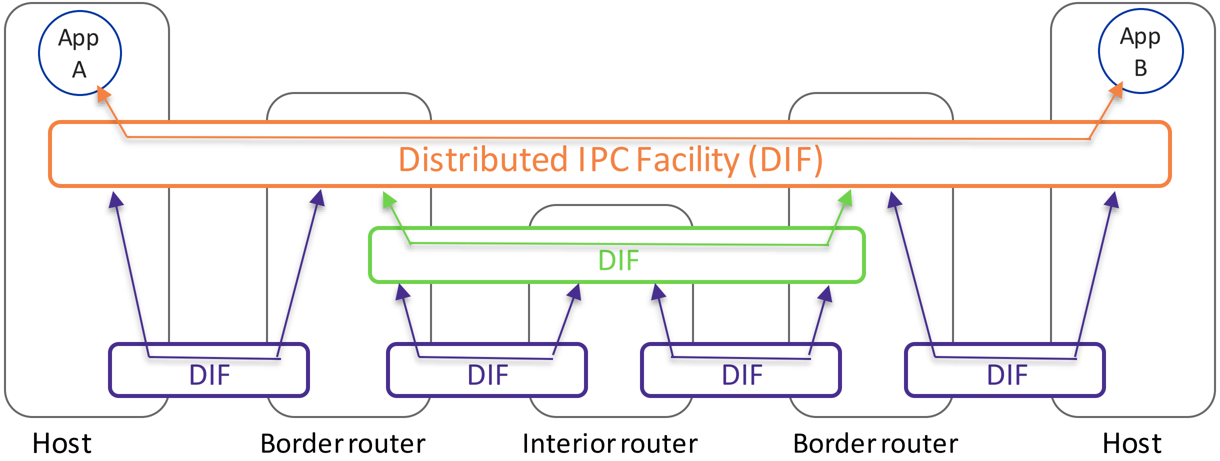
Thus **the network is just a distributed application specialised to provide IPC**.

We'll call this application a **DIF (Distributed IPC Facility)**.



1. Structure: layering

- But a single DIF for all applications and all machines in the world/universe is not very scalable…
  - ○ We need to isolate **scopes** (link, network, Internet, VPN, etc.)
- Solution: have multiple DIFs, providing IPC services to each other!
  - ○ After all a DIF is just a distributed application, right?

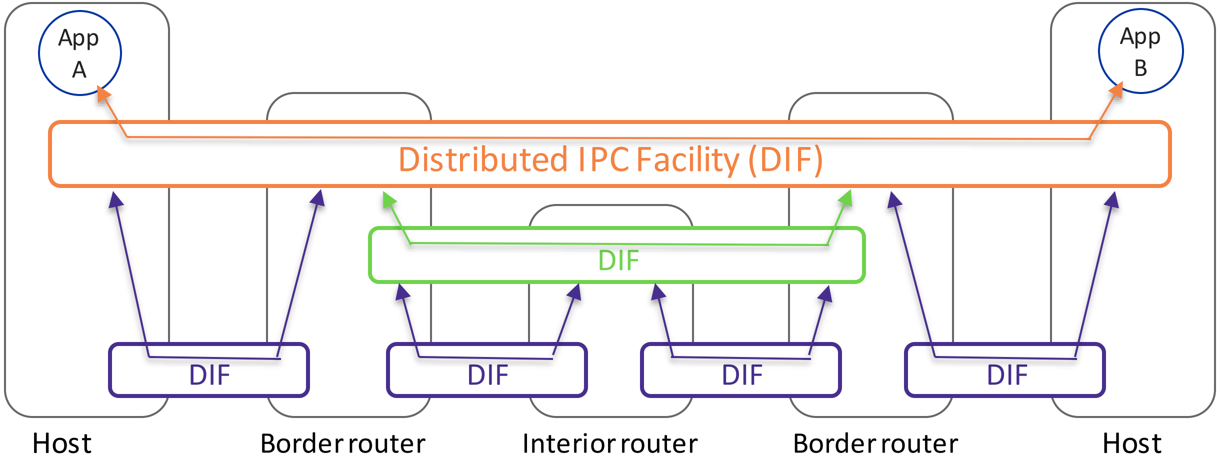


© "RINA Introduction" Eduard Grasa. Arcfire 2017

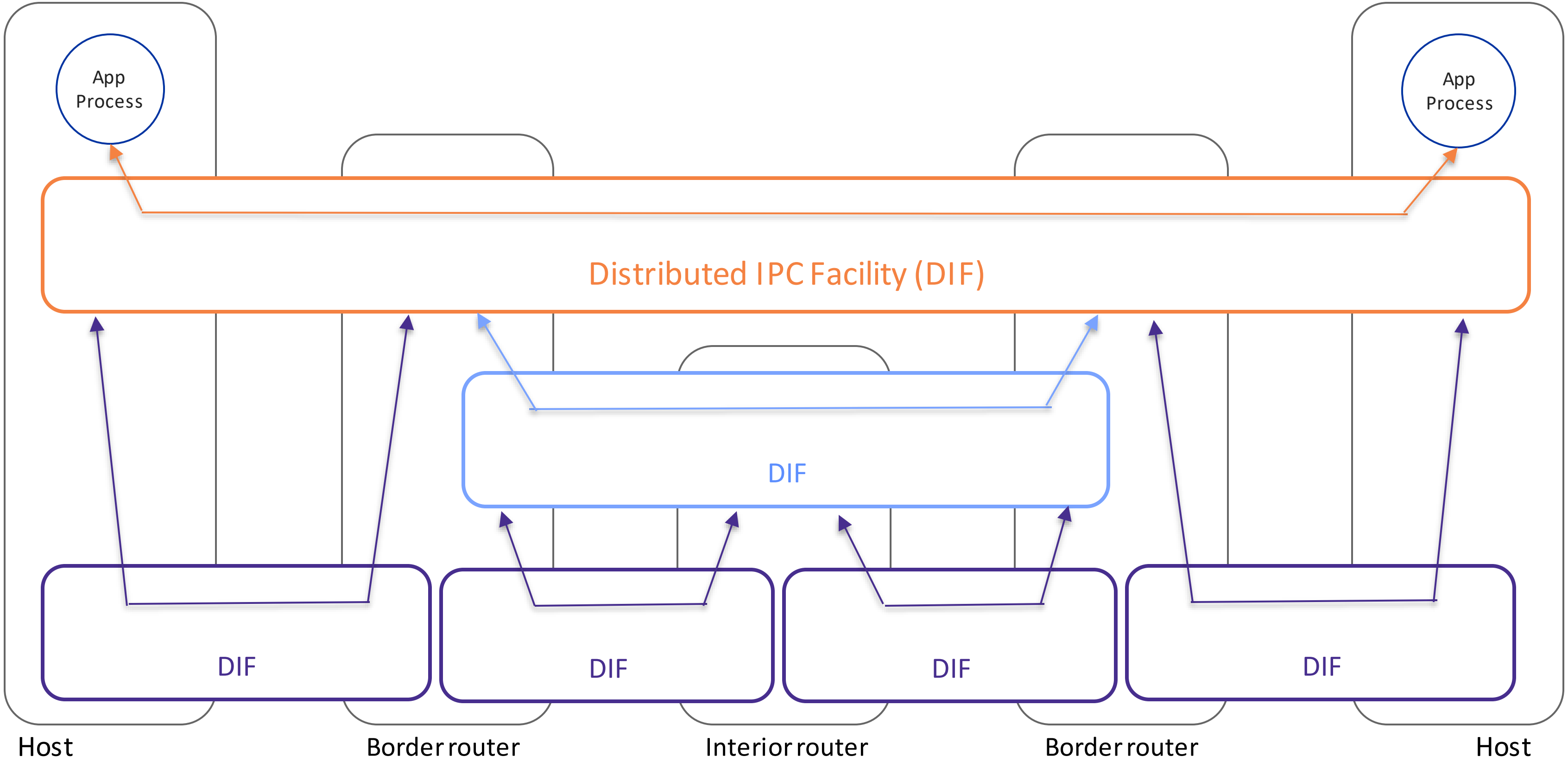
1. Structure: layering, a better pattern

Single type of layer, providing an IPC service that repeats as many times as needed by the network designer.

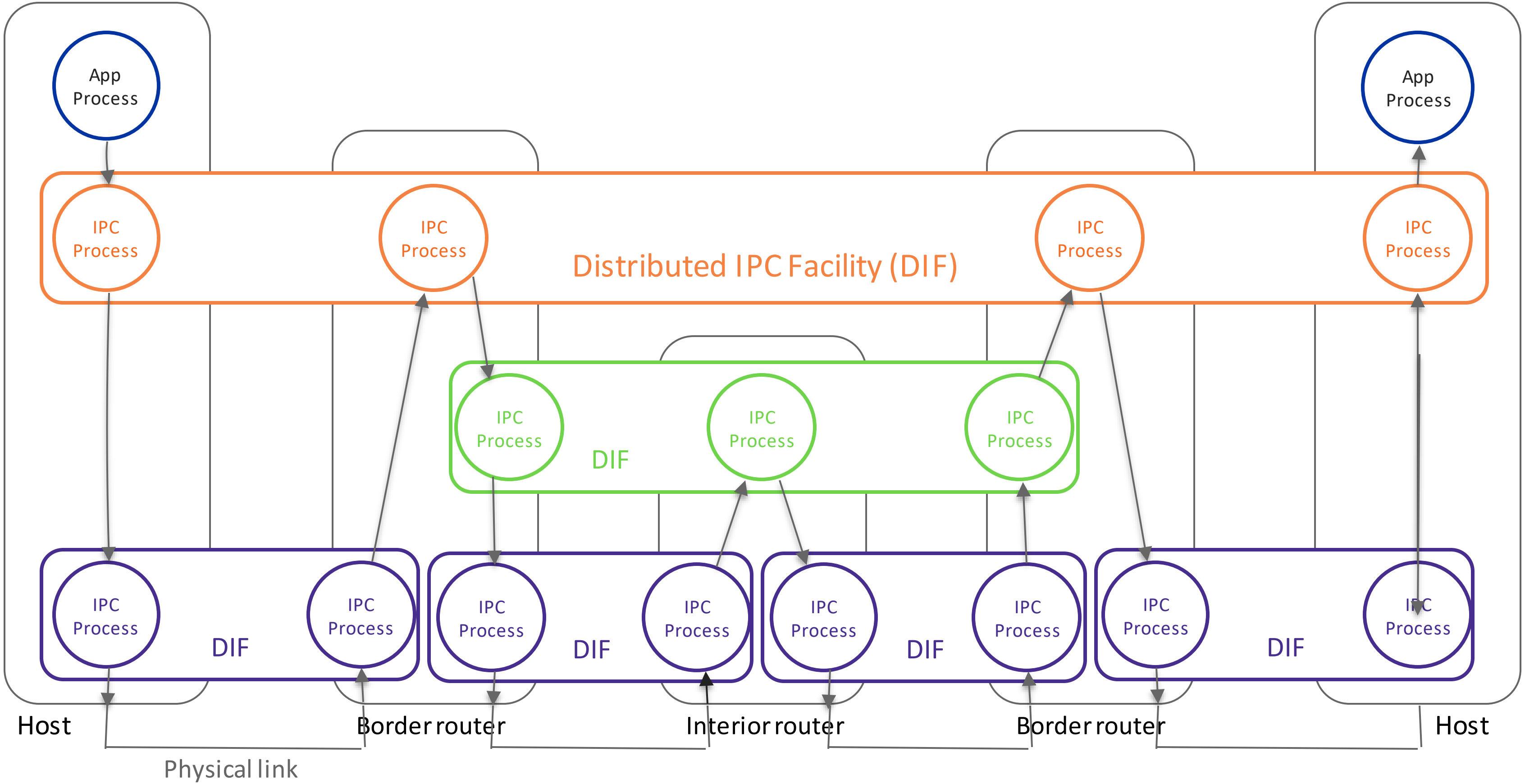
A layer is a **resource allocator** that **provides and manages the IPC service over a given scope** (link, network, Internet, VPN, etc.). A layer allocates resources (memory in buffers, scheduling capacity, bandwidth) to competing flows.



The DIF being a distributed application…

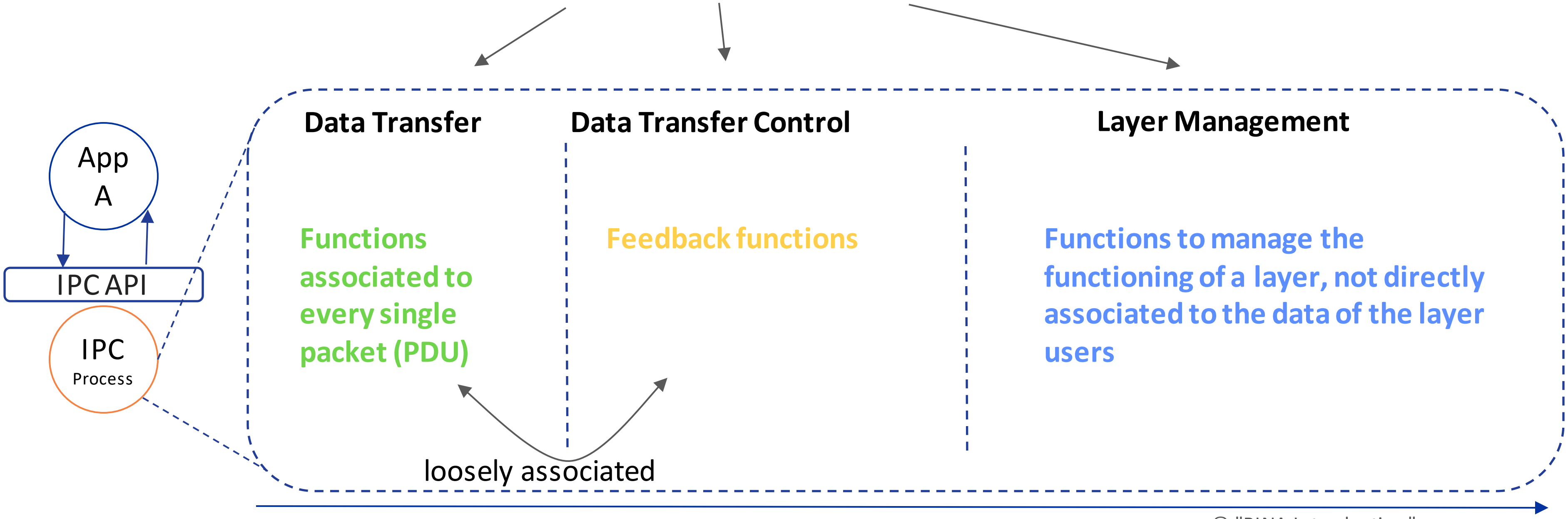


The processes of the DIF are IPC Processes



Organization of functions inside a DIF

Each DIF performs a number of distributed functions coordinated via network protocols, which can be categorised:



Data Transfer

Functions associated to every single packet (PDU)

Data Transfer Control

Feedback functions

Layer Management

Functions to manage the functioning of a layer, not directly associated to the data of the layer users

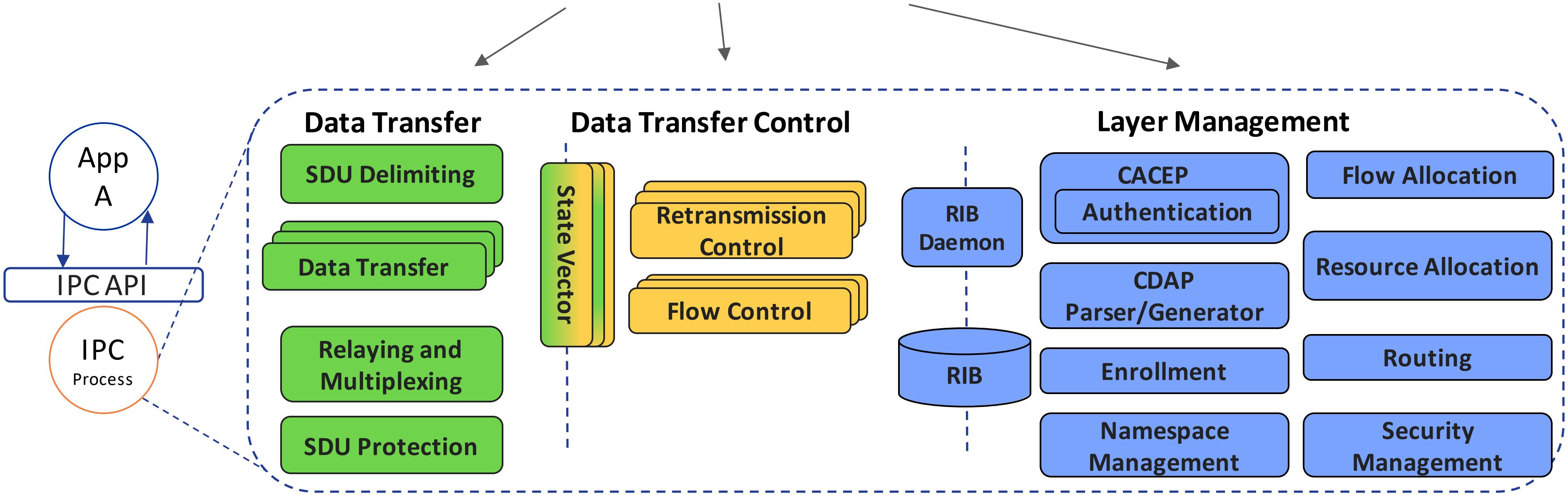
loosely associated

Increasing timescale (functions performed less often) and complexity

© "RINA Introduction"
Eduard Grasa. Arcfire 2017

Organization of functions inside a DIF

Each DIF performs a number of distributed functions coordinated via network protocols, which can be categorised:



© "RINA Introduction"
Eduard Grasa. Arcfire 2017

2. What protocols inside a DIF?

To limit the variability in protocols to the minimum, we apply **separation of mechanism and policy**:

- **Mechanism** = part in a protocol that is **fixed**
  (e.g. an acknowledgement ACK)

- **Policy** = part of the protocol that **can change**
  (e.g. when to send an ACK)

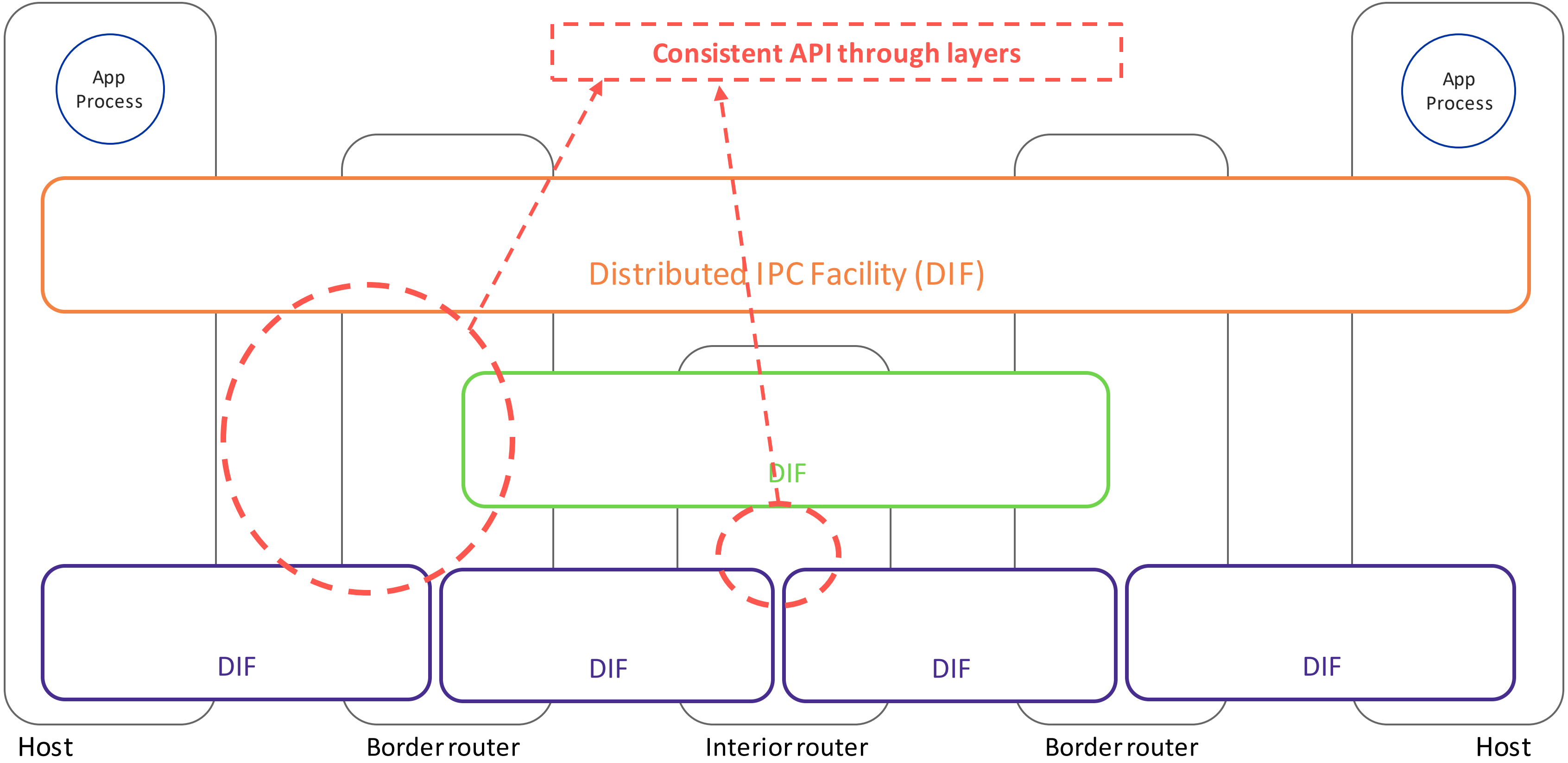
2. What protocols inside a DIF?

Each DIF has different requirements, so we cannot have the same protocols in all of them, but can we **abstract invariances** so that we end up with:

one protocol (framework) for data transfer
(EFCP - Error and Flow Control Protocol)

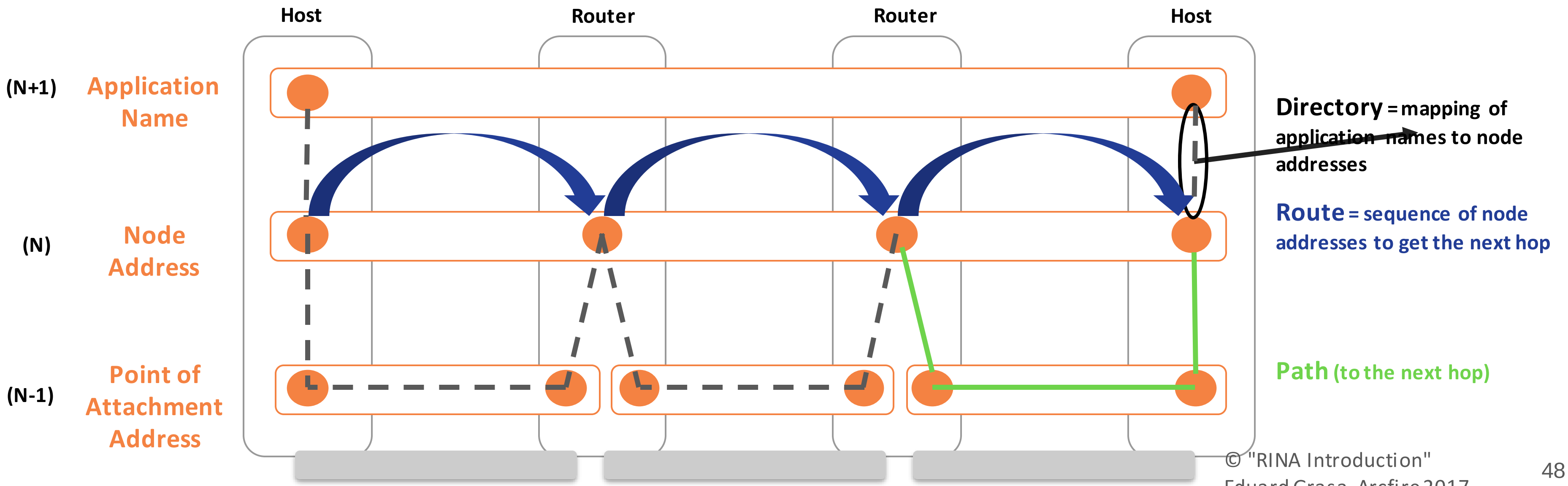
one protocol (framework) for layer management
(CDAP - Common Distributed Application Protocol)

3. API


Consistent API through layers
App Process
App Process
Distributed IPC Facility (DIF)
DIF
DIF
DIF
DIF
DIF
Host
Border router
Interior router
Border router
Host

4. Naming and Addressing

- Application names are location-independent to allow an application to move around
- Node addresses are location-dependent but route-independent
- PoA addresses are by nature route-dependent
- Mobility and multihoming are inherent. No need for special protocols.



Directory = mapping of application names to node addresses

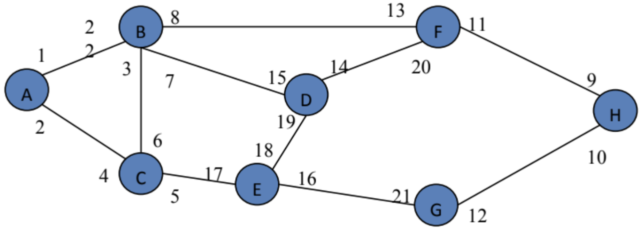
Route = sequence of node addresses to get the next hop

Path (to the next hop)

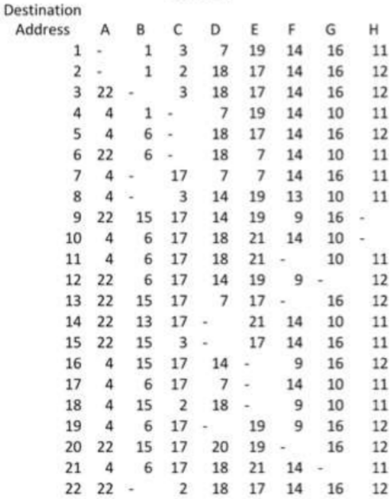
© "RINA Introduction"
Eduard Grasa. Arcfire 2017

5. Implications for multi-homing

Addresses assigned to interfaces (like in IP)

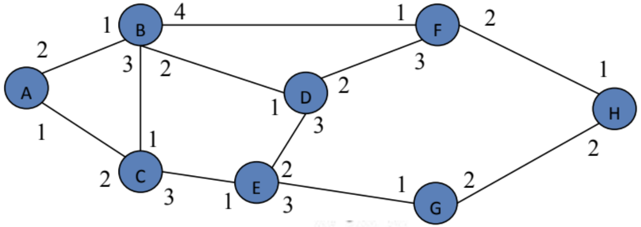


Next Hop

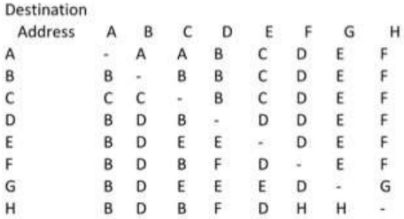


Destination Address	A	B	C	D	E	F	G	H
1	-	1	3	7	19	14	16	11
2	-	1	2	18	17	14	16	12
3	22	-	3	18	17	14	16	12
4	4	1	-	7	19	14	10	11
5	4	6	-	18	17	14	16	12
6	22	6	-	18	7	14	10	11
7	4	-	17	7	7	14	16	11
8	4	-	3	14	19	13	10	11
9	22	15	17	14	19	9	16	-
10	4	6	17	18	21	14	10	-
11	4	6	17	18	21	-	10	11
12	22	6	17	14	19	9	-	12
13	22	15	17	7	17	-	16	12
14	22	13	17	-	21	14	10	11
15	22	15	3	-	17	14	16	11
16	4	15	17	14	-	9	16	12
17	4	6	17	7	-	14	10	11
18	4	15	2	18	-	9	10	11
19	4	6	17	-	19	9	16	12
20	22	15	17	20	19	-	16	12
21	4	6	17	18	21	14	-	11
22	22	-	2	18	17	14	16	12

Addresses assigned to nodes (like in RINA)



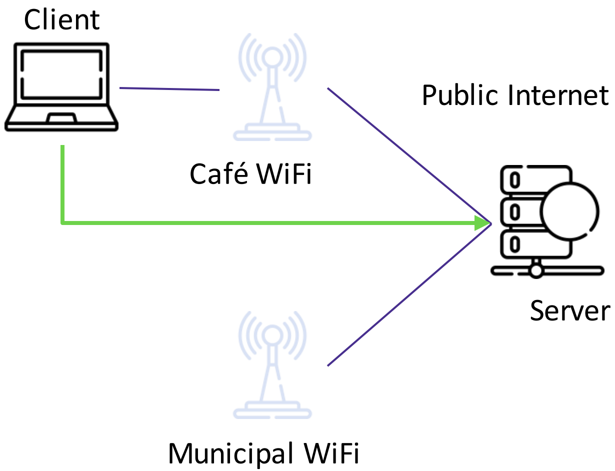
Using Next Hop



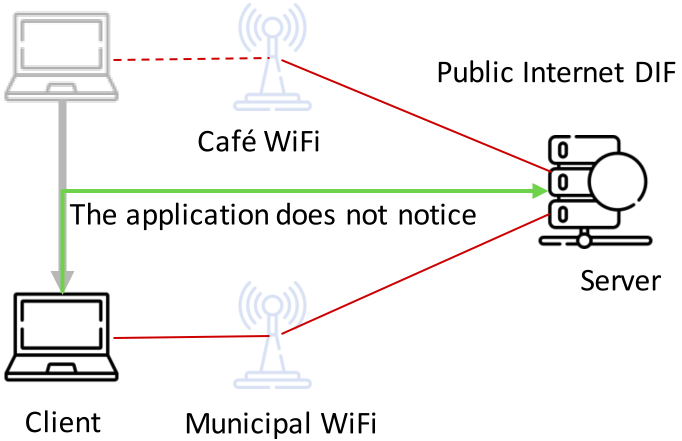
Destination Address	A	B	C	D	E	F	G	H
A	-	A	A	B	C	D	E	F
B	B	-	B	B	C	D	E	F
C	C	C	-	B	C	D	E	F
D	B	D	B	-	D	D	E	F
E	B	D	E	E	-	D	E	F
F	B	D	B	F	D	-	E	F
G	B	D	E	E	E	D	-	G
H	B	D	B	F	D	H	H	-

- Addressing the node instead of the interface: ***3-4x time routing/forwarding table reduction***!
- ***No*** need for ***special protocols to support multi-homing***

6. Mobility



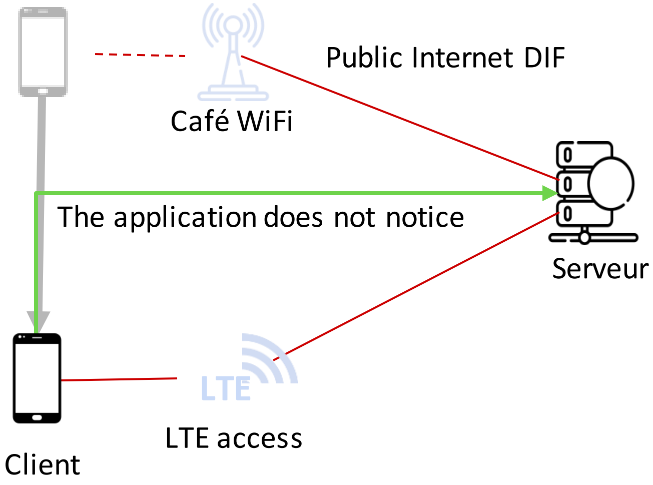
When I walk out of the café...



With RINA: seamless handover.

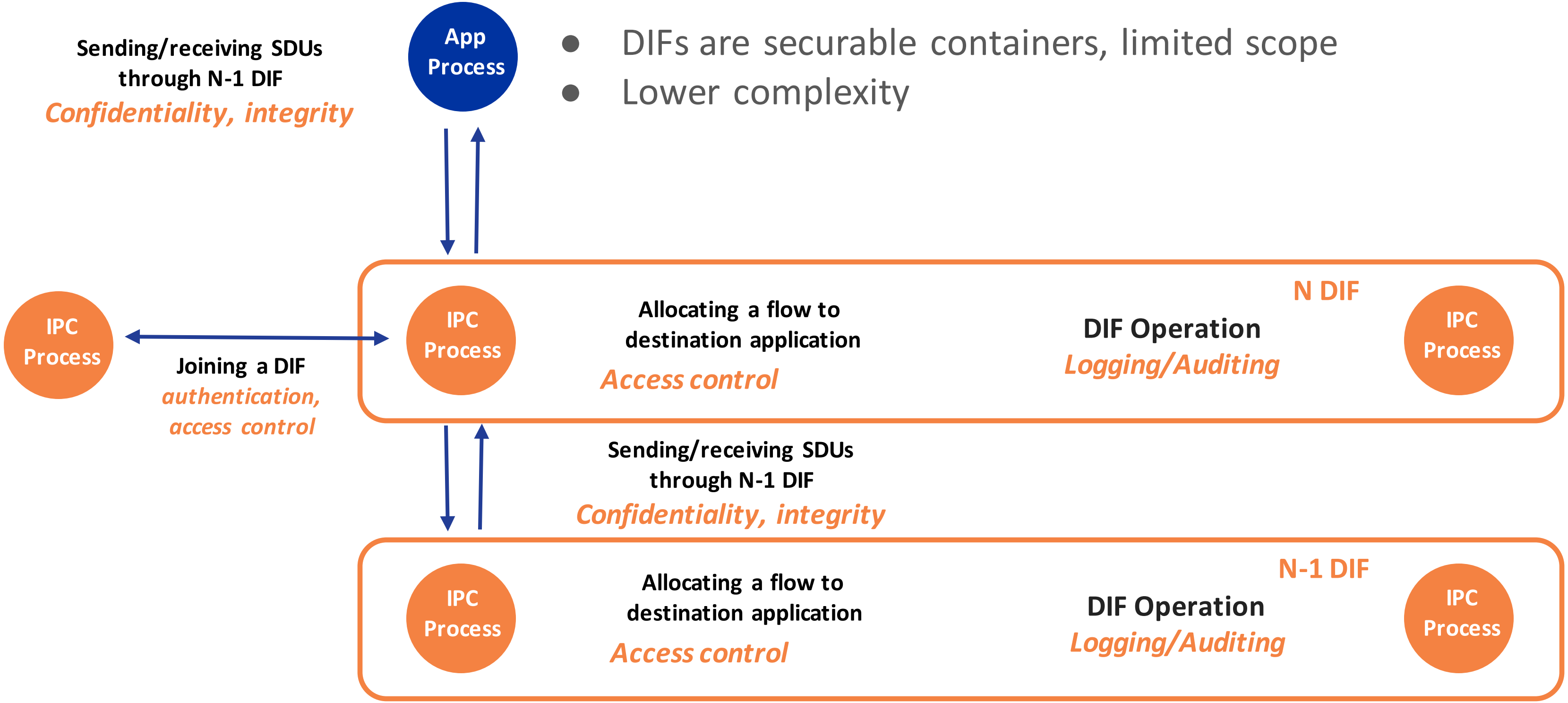
6. Mobility

Also cross technology mobility



7. Security: DIFs are securable containers

- DIFs are securable containers, limited scope
- Lower complexity


Sending/receiving SDUs through N-1 DIF
Confidentiality, integrity
App Process
IPC Process
Joining a DIF
authentication, access control
IPC Process
Allocating a flow to destination application
Access control
N DIF
DIF Operation
Logging/Auditing
IPC Process
Sending/receiving SDUs through N-1 DIF
Confidentiality, integrity
IPC Process
Allocating a flow to destination application
Access control
N-1 DIF
DIF Operation
Logging/Auditing
IPC Process

8. Quality of Service

QoS classes with different restrictions on several parameters such as:

- bandwidth
- delay
- loss rate
- ordered or not ordered delivery
- jitter

9. Congestion control

- Congestion control loops using ECN **close** to where congestion occurs

- Per-DIF **customized** congestion control **policies** for heterogeneous networks.

Deployment

- No need for clean slate deployment or big bang

- RINA can be deployed incrementally where it has the right incentives, and interoperate with current technologies (IP, Ethernet, MPLS, etc.)
  - Over IP (just like any overlay - VXLAN, NVGRE, GTP-U, etc.)
  - Below IP (just like any underlay MPLS or MAC-in-MAC)
  - Next to IP (gateways/protocol translation - like IPv6)
- There are 2 main prototype implementations of RINA: IRATI and rlite.

Summary (RINA is not a protocol!)

① Network architecture resulting from a fundamental theory of computer networking.

② Networking is InterProcess Communication (IPC) and only IPC. Unified networking and distributed computing: the network is a distributed application that provides IPC.

③ There is a single type of layer with programmable functions, that repeats as many times as needed by the network designers (DIF!).

④ All layers provide the same service: instances or communication (flows) to two or more application instances, with certain characteristics (delay, loss, in-order-delivery, etc).

⑤ There are only 3 types of systems: hosts, interior and border routers. No middleboxes (firewalls, NATs, etc) are needed.

⑥ Deploy it over, under and next to current networking technologies.

The RINA slides were adapted from a presentation from

e-Hayt Research Foundation

RINARMENIA

Reading materials

SCION

- https://www.scion-architecture.net/pdf/2017-SCION-CACM.pdf
- https://www.scion-architecture.net/pages/publications/

RINA

- http://rina.tssg.org/docs/ITP_vol5_p3_42-50.pdf
- https://www.etsi.org/deliver/etsi_gr/NGP/001_099/009/01.01.01_60/gr_NGP009v010101p.pdf

2STIC

Thanks for your attention!

caspar.schutijser@sidn.nl
joeri.deruiter@sidn.nl

www.2stic.nl
www.sidnlabs.nl