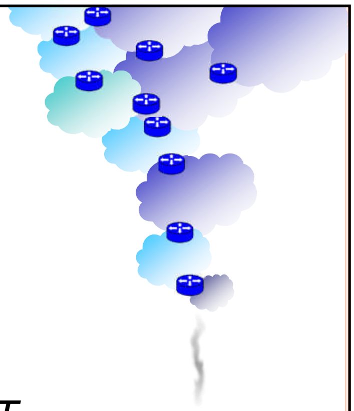


The Edge of the IoT

INFORMATION CENTRIC NETWORK

Julien Fasson – Maître de conférences INPT/ENSEEIHT
julien.fasson@enseeiht.fr

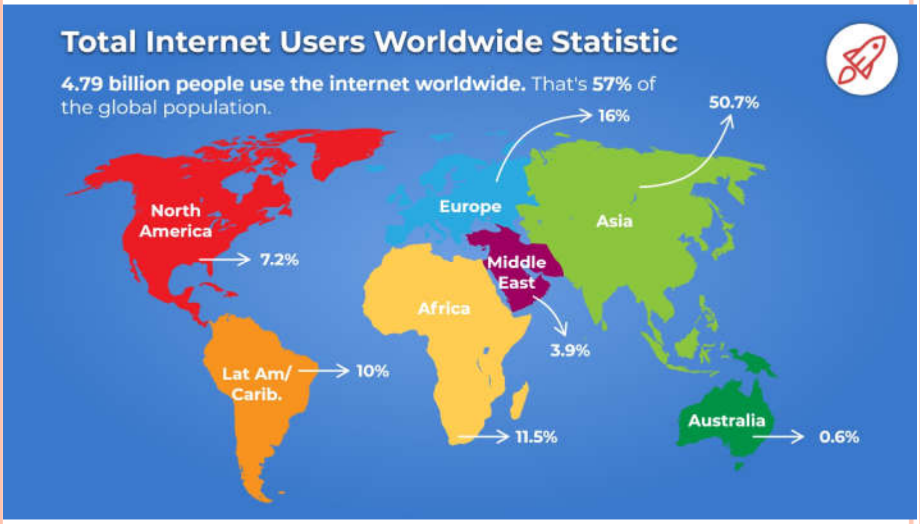
MY SPECIAL THANKS TO

Adrien THIBAUD
PhD Student on CC in NDN

Alexandre FOURES and François HECTOR
Ingénieurs N7 Informatique et Réseaux

2

The Internet Today



*https://www.broadbandsearch.net/blog/internet-statistics

Summary

I – From Host to Content

- From a centralized design to a decentralized adventure
- Decentralized data structure

II – Information Centric Networks

- Keys
- Timeline

III – Name Data Networks

- Principle
- Issue

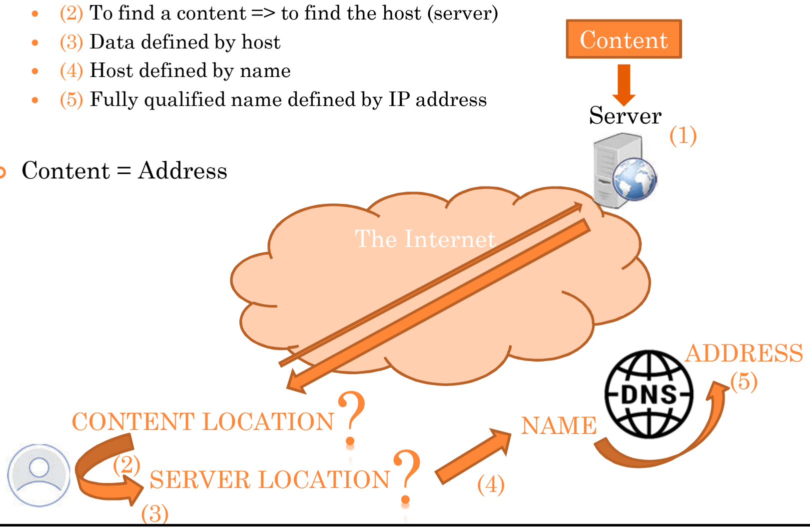
From Host to Content
Content

- From a centralized design to a decentralized adventure
  - Client/Server
  - Mirrors
  - CDN
  - Cloud, Fog and Edge
- Decentralized data structure
  - Needs
  - Hash
  - Hash tree

5

The First Commercial Model in The Internet
Centralised Content – The Server

- Centralized Content
  - (1) One content => one host
  - (2) To find a content => to find the host (server)
  - (3) Data defined by host
  - (4) Host defined by name
  - (5) Fully qualified name defined by IP address
- Content = Address



6

The First Commercial Model in The Internet
Uniform Ressource Locator

- Uniform Resource Identifier
  - URN: Uniform Resource Name
  - URL: Uniform Resource Locator
- A string used to uniquely identify a resource on the WWW
- Syntax of a URL:

  protocol://user:pass@domain:port/path?query#fragment

http://www.enseeiht.fr/fr/formation/masteres-specialises/securite-informatique.html
https://en.wikipedia.org/w/index.php?title=IP_over_Avian_Carriers&action=edit&section=2

7

The First Commercial Model in The Internet
Host Centric Design

- The Internet = an host centric solution
  - Based on the location of host
- Pro
  - ____________________
  - ____________________
  - ____________________
  - ____________________
- Cons
  - ____________________
  - ____________________
  - ____________________
  - ____________________

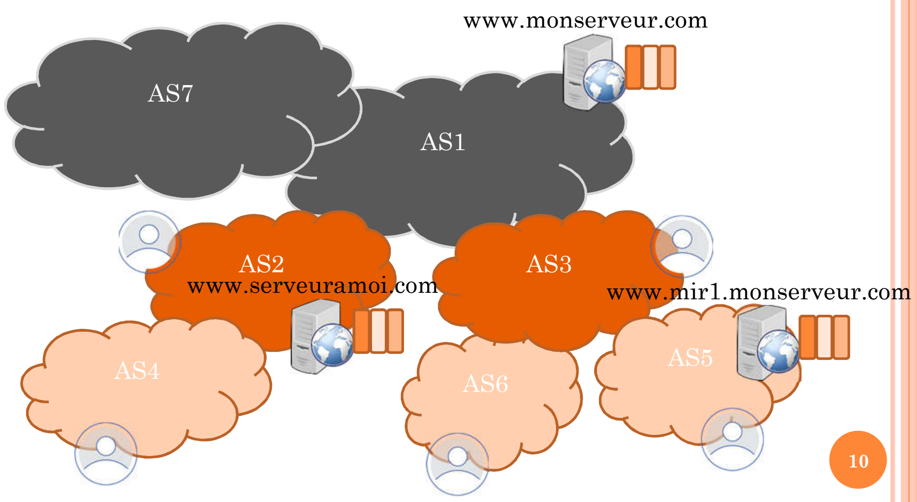
8

The First Commercial Model in The Internet
Mirrors

- A mirror site is
  - a replica of all or part of (web)site hosted on another server.
  - Using generally another URL than the primary site,
    - a subdomain
    - a completely different domain.
- Goal :
  - to distribute the traffic load
  - to reduce latency by locating servers closer users
- Services: ***www*** and ***ftp***
- Drawbacks :
  - ____
  - ____
- Example
  - https://www.debian.org/mirror/list
    - France ftp.fr.debian.org/debian/
    - Canada ftp.ca.debian.org/debian/

9

The First Content Distribution
Mirrors Illustration



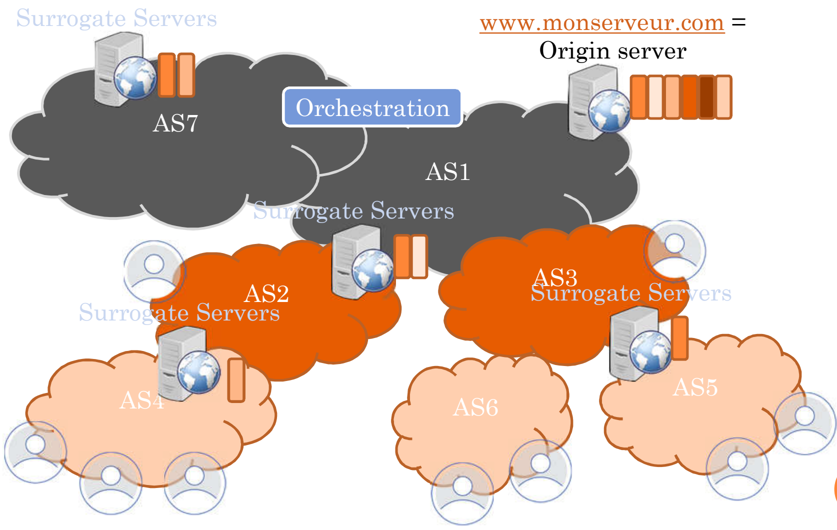
10

Content Delivery Network
With Infrastructure

- “A **content delivery network**, or **content distribution network** (**CDN**), is a geographically distributed network of proxy servers and their data centers. The goal is to provide high availability and performance by distributing the service spatially relative to end users.” Wikipedia
- Appears in the end of 90’s
  - to manage/replace mirrors
  - because the Internet is becoming a mission-critical medium of communication
- Goals
  - To alleviate the issue due to bottlenecks of the Internet.
  - To enhance user QoE
- How
  - Build a overlay network
  - Use of caches, called surrogate servers
    - Cache part of content
    - Proactive or Opportunist
  - Use of redirections
    - More common redirections is made through DNS

11

Content Delivery Networks
Illustration



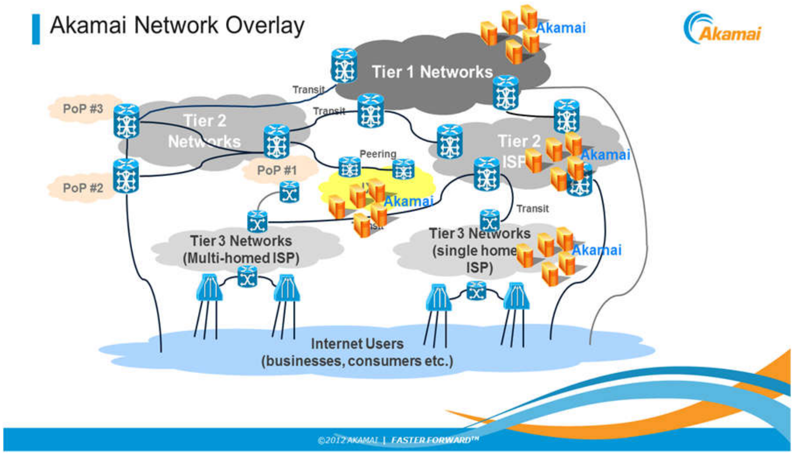
12

Content Delivery Networks
Actors

- Main Actors
  - ____________________
  - ____________________
  - ____________________
  - ____________________
- Pro
  - ____________________
  - ____________________
  - ____________________
  - ____________________
- Cons
  - ____________________
  - ____________________
  - ____________________
  - ____________________

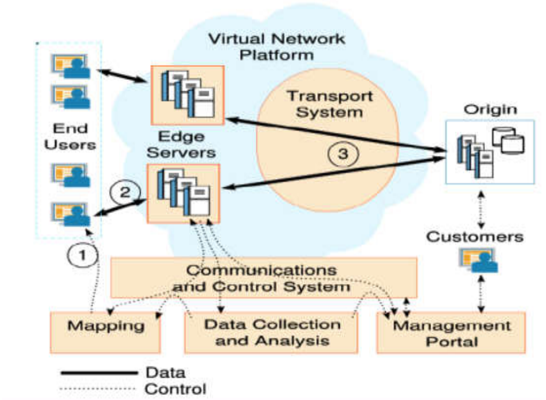
13

Content Delivery Network
Akamaï Example



14

Content Delivery Network
Akamaï Example

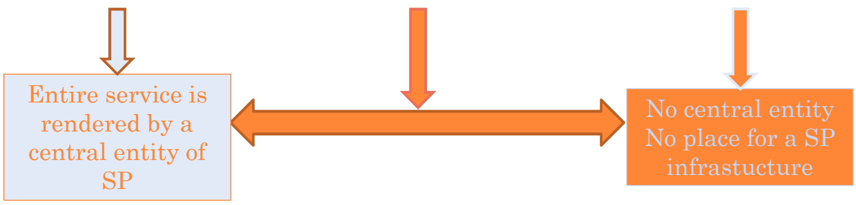


Study (1) *Impact des overlay dans Internet*
Overlay Networks: An Akamai Perspective - 2014 wiley

15

Content Delivery Network
Without Infrastructure – P2P

- "**Peer-to-peer** (**P2P**) computing or networking is a distributed application architecture that partitions tasks or workloads between peers. Peers are equally privileged, equipotent participants in the application. They are said to form a peer-to-peer network of nodes." Wikipedia



16

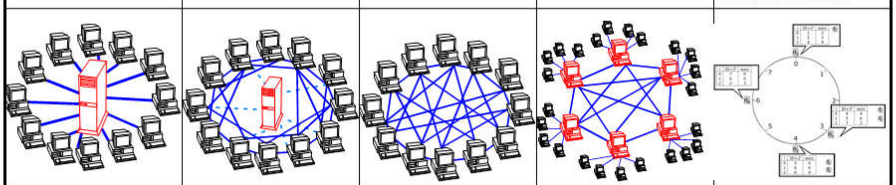
Content Delivery Network
Without Infrastructure – P2P

- Appears in the end of 90's – 1999 with NAPSTER
  - 1999 – Napster – a centralized P2P – Shawn Faning
  - 2000 – Gnutella 0.4 – "Pure" P2P – Nullsoft (AOL)
  - 2001 - BitTorent – Bram Cohen
- Goals
  - To alleviate the issue due to bottlenecks of the Internet
  - To enhance user QoE
  - Independence from Service Providers
- How
  - Build a overlay network
  - Use of End Users resources
    - End User are both client/servers (Servent)
    - User Friendly?
  - A lot of applications

17

Content Delivery Network
Without Infrastructure – P2P

Client-Server	***Peer-to-Peer***			
1. Server is the central entity and only provider of service and content. → Network managed by the Server 2. Server as the higher performance system. 3. Clients as the lower performance system Example: WWW	1. Resources are shared between the peers 2. Resources can be accessed directly from other peers 3. Peer is provider and requestor (Servent concept)			
	Unstructured P2P			***Structured P2P***
	Centralized P2P	*Pure P2P*	*Hybrid P2P*	*DHT-Based*
	1. All features of Peer-to-Peer included 2. Central entity is necessary to provide the service 3. Central entity is some kind of index/group database Example: Napster	1. All features of Peer-to-Peer included 2. Any terminal entity can be removed without loss of functionality 3. → No central entities Examples: Gnutella 0.4, Freenet	1. All features of Peer-to-Peer included 2. Any terminal entity can be removed without loss of functionality 3. → dynamic central entities Example: Gnutella 0.6, JXTA	1. All features of Peer-to-Peer included 2. Any terminal entity can be removed without loss of functionality 3. → No central entities 4. Connections in the overlay are "fixed" Examples: Chord, CAN



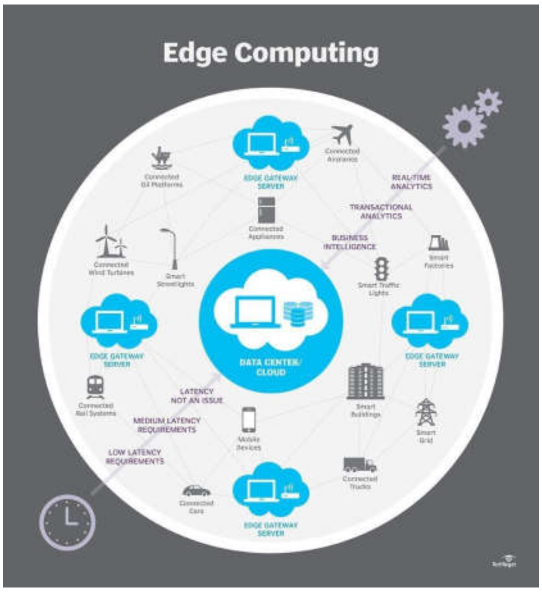
1st Gen. 2nd Gen.

Cloud , Fog and Edge

- Cloud Definition
  - First use is more like online storage
  - “Centralised infrastructure in data centres that host a lot of compute and storage resources.” BCS – History of the Cloud
- Timeline
  - 2002 - Amazon Web Services (AWS) launching its public cloud in 2002.
  - 2006 - Database services available on the cloud (Dropbox = cloud storage as a service)
  - 2006 - Amazon Elastic Compute Cloud (EC2) = rent virtual server to deploy your own web service
  - 2017 at now – to reduce latency
    - **Compute and store next to user**
- Services
  - Software as a Service (SaaS)
  - Platform as a Service (PaaS)
  - Infrastructure as a Service (IaaS)
- Key elements of cloud
  - Virtualization
  - Network
  - Distribution

19

Cloud, Fog and Edge



20

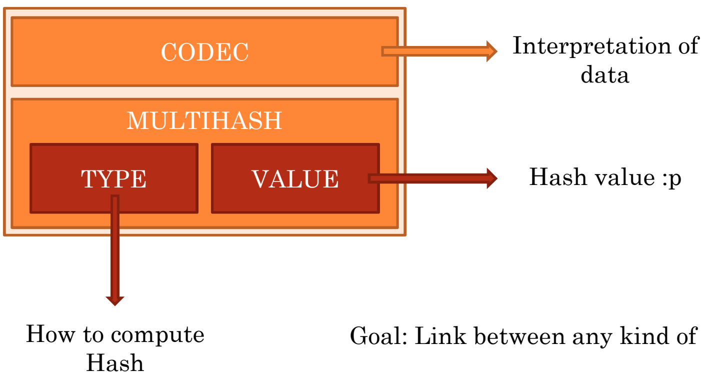
Decentralized Data Structures
Hash

- Distribution through the network and virtualization implies
  - Decentralized Data Structures
- Needs :
  - To trust the data structure => verification function
  - To find the data => link function
- Tool = Cryptographic hashing
  - Hash represents a content, a data
  - It is unique
  - Use of fixed size hash
- Issues
  - Not human readable
  - Must have the complete data to compute its hash
  - How to mix different types of content? Of hash functions?

21

Decentralized Data Structures
CID

- **C**ontent **ID**entifiers
  - First developed by **I**nter **P**lanetary **F**ile **S**ystem (https://ipfs.io) 2013
  - Codec + Multihash



Goal: Link between any kind of data

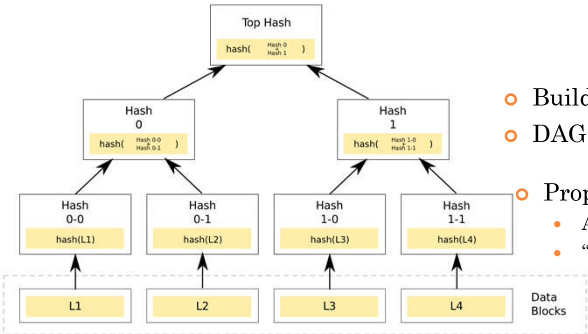
22

Decentralized Data Structures

Hash Tree

- More Known as Merkle tree

  "In cryptography and computer science, a **hash tree** or **Merkle tree** is a tree in which every leaf node is labelled with the cryptographic hash of a data block, and every non-leaf node is labelled with the cryptographic hash of the labels of its child nodes. Hash trees allow efficient and secure verification of the contents of large data structures. Hash trees are a generalization of hash lists and hash chains"



- Build from leafs up to the root
- DAG is a specific hash tree
- Properties:
  - Authentication chunk + whole
  - "Distributivity"

23

Summary & Questions

- ICN most successful architecture

  => NDN

- NDN principles
  - ⇒ Receiver Driven (pull)
  - ⇒ Naming
  - ⇒ Caching
  - ⇒ Forwarding
- Issues/questions
  - ⇒ Caching management
  - ⇒ Routing
  - ⇒ Congestion Control
  - ⇒ Deployment
  - ⇒ Mobility
  - ⇒ ...

24

II Information Centric Network
Big Picture

- Paradox
  - IP = communication network with packet with endpoints name only
    - ⇒ End to End communication
  - The Internet = distribution network
    - ⇒ End to End communication is not relevant
- To place content at the centre of the network
  - Location of the hosts should not be the key anymore
- Challenges
  - Data Name
  - Data routing
  - Data distribution
  - … and to replace IP! (and all its patches!)

25

II Information Centric Network
Big Picture

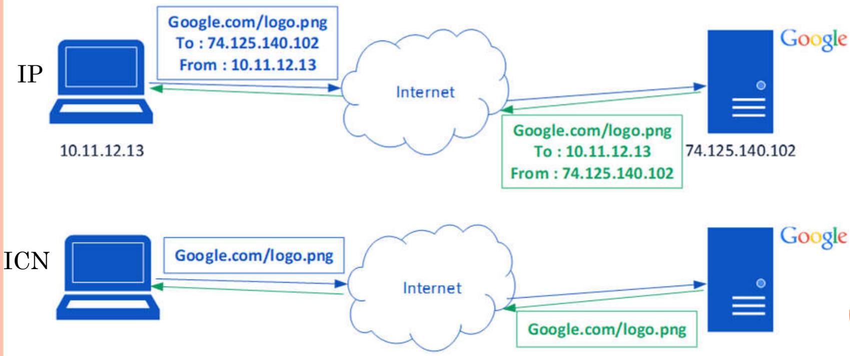
- Opportunities
  - The multi-opportunities
    - Multicast
    - Multi-homing
    - Multi-producers
    - Choose your own multi-thing
  - The storage through the network
    - ICN nodes caching ability
    - And their quite important questions:
      - Where? (Edge vs core)
      - How to decide?
      - What? (Chunk = just part)
  - A more secured world
    - Data includes native security
  - Mobility?

26

II Information Centric Network

Big Picture (bis)

- Tomorrow's Internet architecture
  - Information-Centric communication model
    - Focus on data directly instead of its location
    - Data is retrieved using a name
  - Solve IP's major issues while keeping its strengths

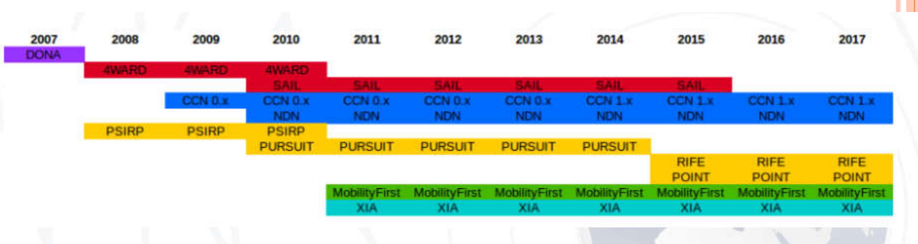


27

*From N7 projet long – François Hector et Alexandre Foures

II Information Centric Network

Timeline



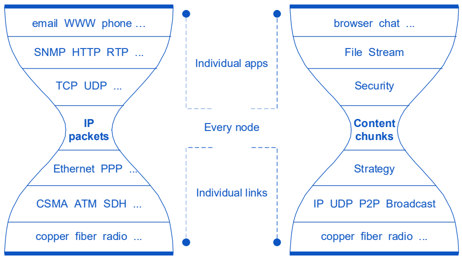
Study (2) *Un autre ICN*

Network of information (netinf) - an information-centric networking architecture

28

III Name Data Network
Hourglass Model

- Internet and NDN hourglass architectures
  - IP: Minimal functionality for global interconnectivity



- NDN
  - Retains this hourglass shape
  - Focus on data directly instead of its location
    - Data is retrieved using a name

29

III Name Data Network
Naming

- Hierarchy in naming
  - Can aggregate paths/routes
  - Nodes will use *Longest Prefix Match* for routing
  - Ressemble the file system
- A consummer must be able to build itself the name of a data…
- Different parts
  - Routing
  - Content/Application
  - Version and **Chunck**

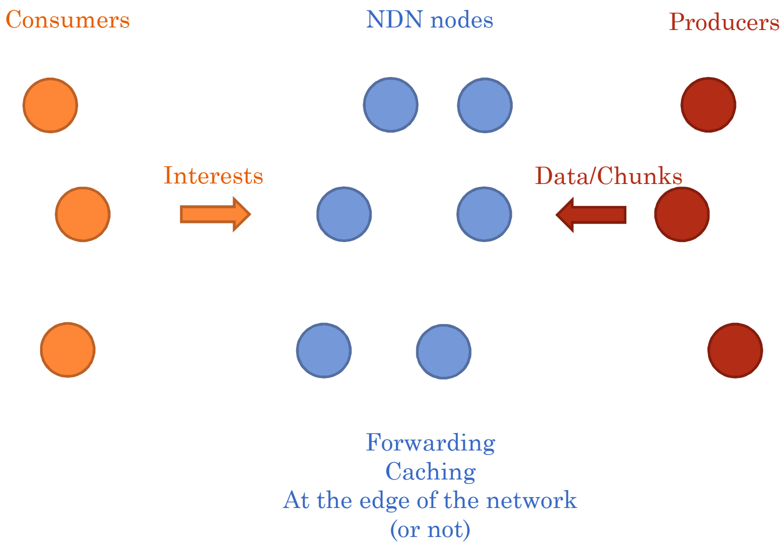
/paris/plan/metro/ligne1/v=2/c=1

/paris/plan/bus/ligne1/v=1/c=5

/toulouse/plan/metro/lignea/v=2/c=3

30

III NAME DATA NETWORK
THE MAIN COMPONENT OF NDN ARCHITECTURE



III NAME DATA NETWORK
INTERESTS

- Interest Packet
  - Name of the requested data (it is generally not the whole content, but a piece of it called chunk)
  - Randomly generated ID by the consumer (sender)
    - Prevent Interest Looping in the network
      - Allow use of multipath to reach a Data

32

Interest Packet
Content Name
Selector
Nonce

III Name Data Network
Data

- Data Packet
  - Answer of an Insterest
    - Name of the encapsulated data
    - Transferred using the Interest's path (in reverse)
  - Signature of the producer
    - For packet authentication
  - Data itself
    - Can be encrypted

Data Packet
Content Name
Signature + Signed Info
Data

33

III Name Data Network
NDN Node Structure - PIT

PIT (Pending Interest Table)			
name	<list> nonce	<list> incoming interface	<list> (outgoing interface, send-time)

⋮

- Create an entry for each new requested name
- List the interfaces that requested the name
- List the interfaces used to propagate the Interest
- Send-Time is used to calculate local RTT when Data is received
  - Round-Trip Time: here is the local delay for the node to receive the data packet, once an interest has been received.
  - Not the RTT of the consumer
- Allow to multicast the Data to requesting interfaces
- Question: Timers, retransmissions, deletions?

34

III Name Data Network
NDN Node Structure - FIB

FIB (Forwarding Information Base)							
Forwarding Ranking →	1	2	3	4	5	6	…
Name prefix	...	...		...	...	...	
⋮							

Interface ID, routing preference, RTT, status

35

- Name prefix entries created by the routing
- Interfaces are ranked following
  - Routing preference
  - RTT (measured with the PIT)
  - Status (green, yellow, red)
- Question : as in IP, how to fill the FIB?

35

III Name Data Network
NDN Node Structure - CS

CS (Content Store)	
name	Data ...
⋮	

- On-Path Caching
  - Opportunistic caching
  - Data can be cached when passing through the router
    - Allow fast retransmission of cached content
- Off-Path Caching
  - Proactive CDN
- Use data from cache
  - Save bandwidth
  - Save server time
- Questions:
  - How to select the content to store?
  - Is opportunistic caching relevant for edges?

36

III Name Data Network
NDN Forwarding Principle

- Chooses interfaces to send an incoming packet
- Stateful plane, decision of the outgoing interface(s) based on
  - Performance/Ranking of the Interfaces
  - Forwarding policies
    - The sooner the better, performance stability
  - How to use multipath
    - Transmit on one, two, all interfaces?

- Key roles
  - Hop-By-Hop "congestion control" = offloading
    - Automatic Loadbalancing
  - Handle detection and recovery of network failure
- Question : Is offloading sufficient for CC?

37

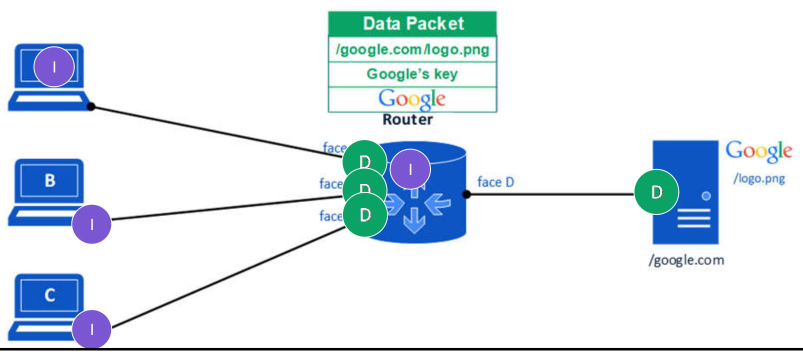
III Name Data Network
NDN Forwarding Illustration

CS (Content Store)	
/google.com/logo.png	Google

FIB (Forwarding Information Base)	
Ranking →	1
/google.com	Interface=D, status=green

PIT (Pending Interest Table)			
name=/google.com/logo.png	nonce=123,456	incoming=A,B	outgoing=D

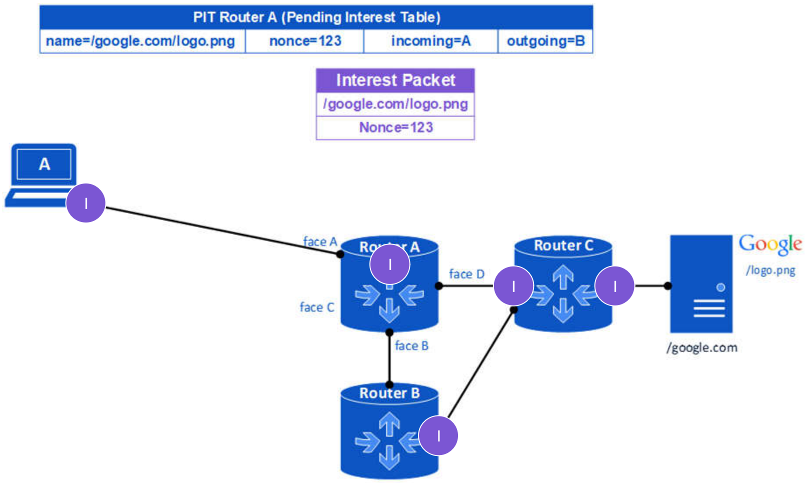
38



38

III Name Data Network

NDN Forwarding – Loop Prevention -Illustration



39

III Name Data Network

Routing in NDN - NLSR

- **<u>Goal</u>**: Fills the routing tables
- NLSR : Named-data Link State Routing Protocol
  - One solution among others
  - Link state routing
    - Each router know the whole network topology and costs
      - Stocked in its LSDB (Link State Database).
  - LSA : Link State Advertisements
    - LSA Name: update list of handled prefix
    - LSA Adjacency: update list of neighbours and cost
  - Use NDN functionalities (naming and security => signature)

Entity	Name
Root key	/<network>/key
Site key	/<network>/<site>/key
Operator key	/<network>/<site>/<operator>/key
Router key	/<network>/<site>/<router>/key
NLSR key	/<network>/<site>/<router>/NLSR/key
Data	/<network>/NLSR/LSA/<site>/<router>/<type>/<ver>

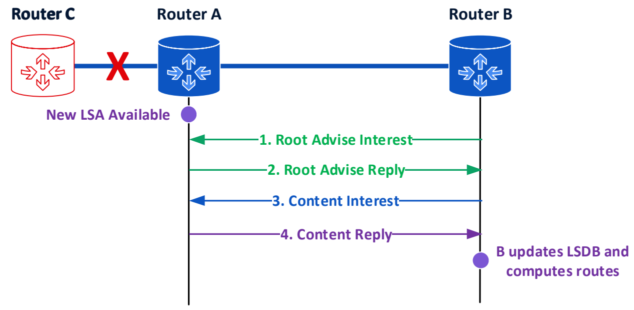
Sign Verify

40

III Name Data Network

Routing in NDN –NLSR Illustration

- Impossible to push data in NDN
  - Interest must be sent to require data
- Example: LSDB Synchronisation (routing update using polling)



- After update, route to all prefixes are computed for each interface
  - Allow multiple interface choice at forwarding level

III Name Data Network

Routing in NDN – Other Solutions

- <u>**OSPFN**</u> Open Shortest Path First for Named-Data
  - The first step before NLSR
  - Use of OLSA – Opaque LSA
    - Exchange of OLSA by routers
    - But OLSA are interpreted by another application
- <u>**Hyperbolic Routing**</u>
  - To use angular/similarity coordinates instead of name
  - Why?
    - Hyperbolicity emerges in very different networks with very different “addressing”
    - Addressing does not really matter
    - Hyperbolicity is linked to popularity and similarity. A rule NDN namespace will certainly follow.
  - Pro: A local view is sufficient for routing
  - Cons: Not the best path or even Deadend

42

III Name Data Network
Congestion Control in NDN

- One Interest => one chunck, no need for CC!
  - Objection!
  - Why?
  - ______________________________
    ____
    ____
- Offloading is sufficient !
  - Objection!
  - Why?
  - ______________________________
    ____
    ____

43

III Name Data Network
Congestion Control in NDN

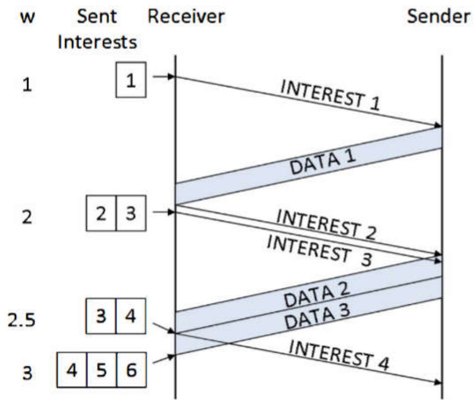
- To change the Internet with…
- … classical solutions of the Internet!

- Type of solutions
  - End-to-End solutions = TCP like
    - TCP like
      - Interests = data segments
      - Data Chunks = ack
      - AIMD algorithms
    - ICP the most common one (Interest Control Protocol)
  - Hop-by-hop solutions
    - AQM like
    - And Forwarding strategy! (path selection = load balancing?)

44

III Name Data Network
Congestion Control in NDN – End-To-End

- ICP
  - A receiver driven TCP
    - Slow Start or Congestion avoidance?
    - ACK or SACK?
  - Issues/Questions
    - Interaction with cache
    - Interaction with NDN node timers
    - Is it really relevant?

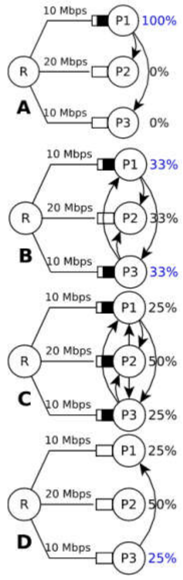


III Name Data Network
Congestion Control in NDN – Hop-by-Hop

- Hop-by-hop solution
  - Use of AQM
    - RED like
    - Shaping
  - Use of Forwarding strategy
- Fast Pipeline Filling (FPF)
  - To fill all the path at their maximum
  - To select the forwarding interface, FPF chooses the one
    - Not at max capa (how to compute well the link capacity?)
    - The lowest RTT

III Name Data Network
Congestion Control in NDN – Hydrid

- Practical Congestion cOntrol (PCON)
  - Use of AQM for congestion detection
    - AQM = CoDel – based on the delay in queue
    - Mark on the data packet
  - NDN Nodes:
    - Forwarding change if possible
  - Consummers
    - At the reception of a marked data
    - Consumer reduces its Interest window
    - ICP like mechanism



47

III Name Data Network
Deployment

- Reality or Utopia?
  - The Internet will not be replace
    - In one day
    - By a single infrastructure
  - Does NDN is lost?
    - Simulation, platform…
    - Devices
    - Use in specific networks
      - Overlays on the Internet
      - Delivery networks
        - Core
        - Edge

48

Lecture, analyse et examen

- (1) Impact des overlay dans Internet
  - **Overlay Networks: An Akamai Perspective - 2014 wiley**
  - Axer l'analyse en conclusion sur les Edges
- (2) NDN et IoT
  - IoT-NDN: An IoT Architecture via Named Data Netwoking (NDN) -2019
  - Axer sur les défis à faire de l'IoT avec NDN et les propositions pour les résoudre

49

THIS IS
THE END