Towards a new generation of information-oriented Internetworking architectures

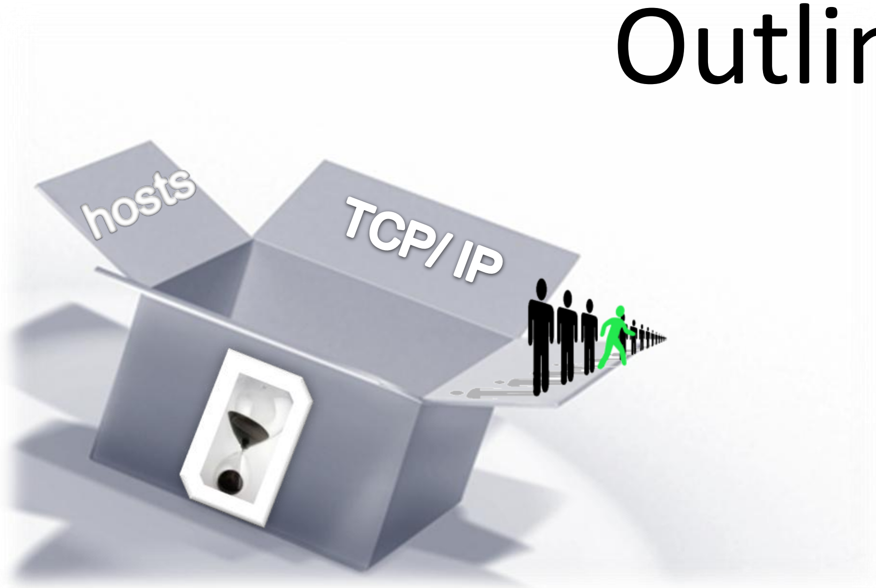
C. Esteve Rothenberg, Fábio L. Verdi and Maurício F. Magalhães
University of Campinas, Brazil
{chesteve, verdi, mauricio}@dca.fee.unicamp.br

ACM CoNEXT 2008
December 9-12, 2008
University Carlos III & IMDEA Networks
Madrid, SPAIN

Sponsored by ACM SIGCOMM
Association for Computing Machinery
Advancing Computing as a Science & Profession

4th Annual ACM CoNEXT Conference 2008

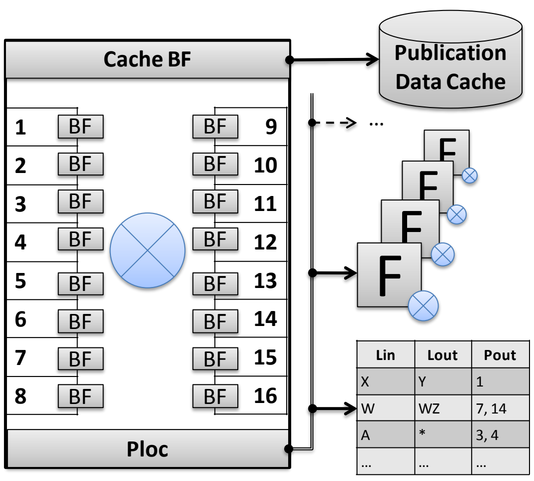
Outline



"out-of-the-TCP/IP-box" thinking

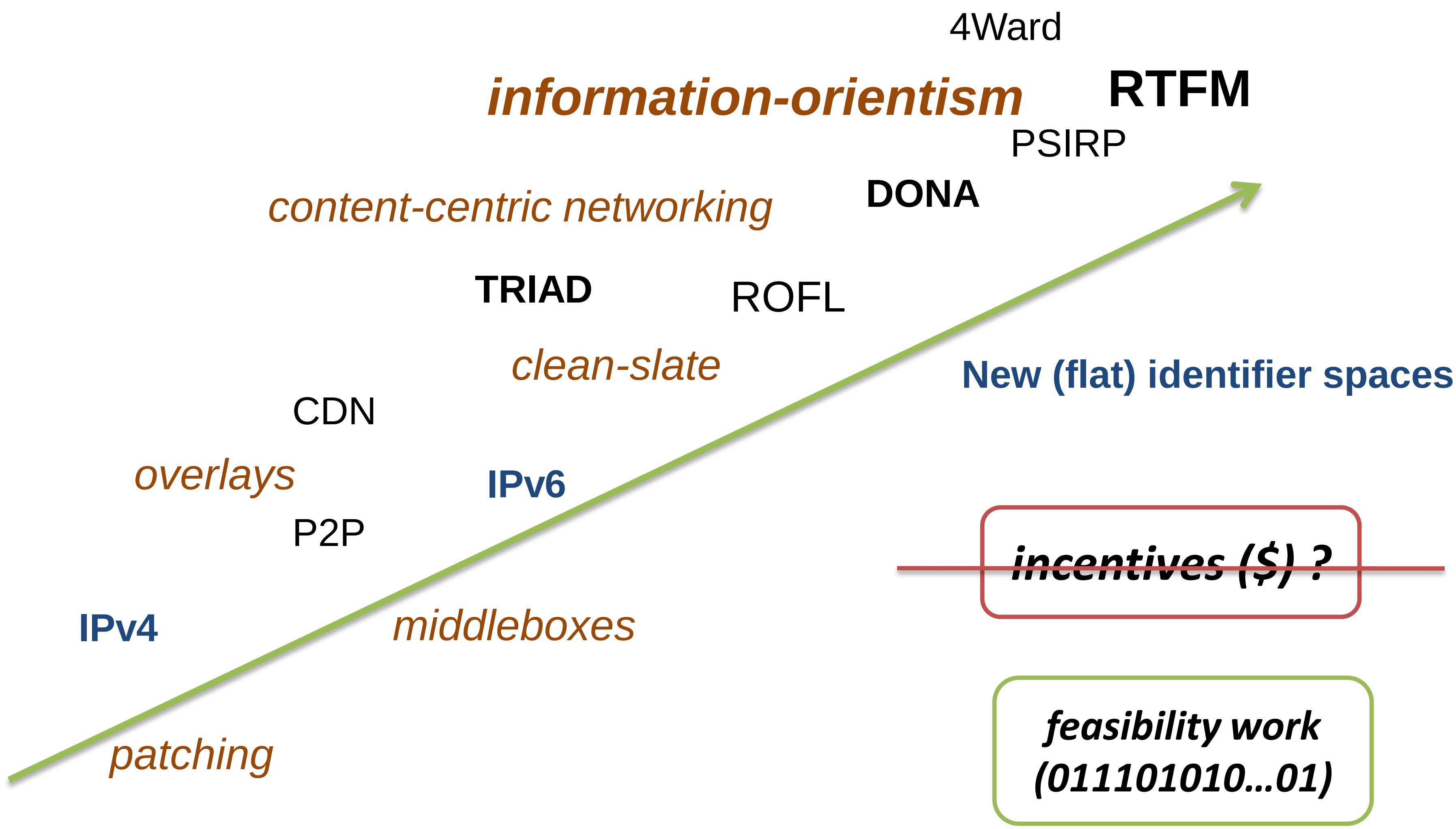
data / content / information networking

large flat labels



SPSwitch: fast scalable forwarding

Re-Architecturing efforts



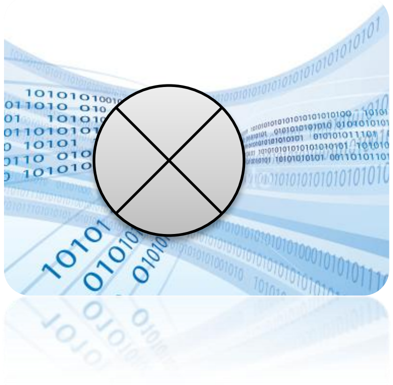
Rethinking concepts

Table 1: Concepts of information-oriented networking versus the original Internet design. Rethinking fundamentals.

Original Internet	Information-Oriented / Content-Centric Internetworking
Sender	Content producer (publisher)
Receiver	Content consumer (subscriber)
Sender-based control	Receiver-based control
Client/Server communications	Publish/Subscribe Sender and Receiver uncoupled
Host-to-host	Service access / Information retrieval
Topology / Domain	Information scope
Unicast	Unified uni-, multi- and anycast
Explicit destination	Implicit destination
End-to-End (E2E)	End-to-Data (E2D)
Host name (look-up oriented)	Data/Content name ("search" activity)
Secure channels, host authentication	Integrity and trust derived from the data

Challenges & Paradigm

- Common **challenge** in data-oriented paradigms:
  - Take switching decisions
    - at *wire speed* (Gbps)
    - on a *large* universe (e.g., 256-bit hash values)
    - of *flat* (non-aggregatable) identifiers



- Let’s take **advantage** of the data-oriented paradigm:
  - Pub/sub inherently tolerates *false positives*
  - Opportunistic *caching*

Longest IP prefix vs. flat label matching

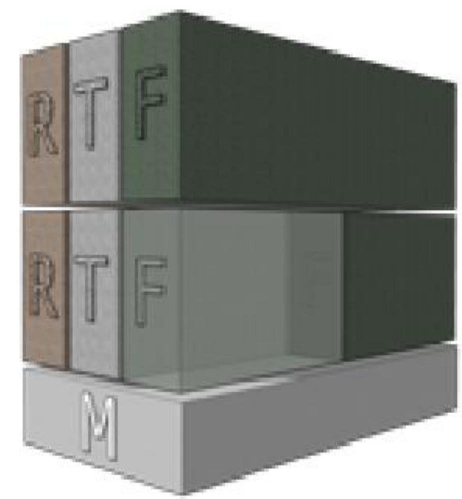
IPv4	Dec.	81.216.171.106	*human*
	Bin.	01010001 11011000 10101011 01101010	*machine*
IPv6	Hex.	ca12:b9fa:655a:0000:0000:ac2f:ccef:f0ab	
	Bin.	11001010 00010010 10111001 11111010 11001010 01011010 00000000 00000000 10101100 00000000 00000000 00101111 11001100 11101111 11110000 10101011	
256-bit flat label	Hex.	08090A0B0D0E0F101213141517181 91A1C1D1E1F2122232426272829 2B2C2D2E	
	Bin.	11111010 11001010 01011010 00000000 00000000 10101100 00000000 00000000 10101100 00000000 00000000 00101111 11001100 11101111 11110000 10101011 11001010 00010010 10111001 11111010 01011010 00000000 00000000 10101100 11101111 11110000 10101011 11001010 00010010 10111001 11111010 11001010	

Aggr. (IPv4, IPv6)

256-bit flat ID matching is *expensive* @ wire speed

RTFM Architecture*

- **Rendezvous**: Matches subscriptions to publications.
- **Topology**: Creates and maintains delivery trees used for forwarding traffic.
- **Forwarding**: Actual data delivery operations. (label switching and fast forwarding tables)
- **Mediation**: Node-to-node link data transfer & **More** (opportunistic caching, collaborative and network coding)

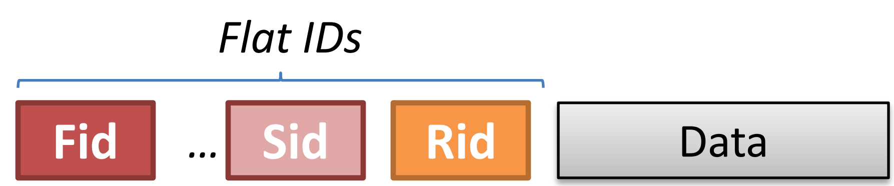


- Metadata and hash-based identifiers

*[Särelä et al. 2008]

Identifiers space (approx.)

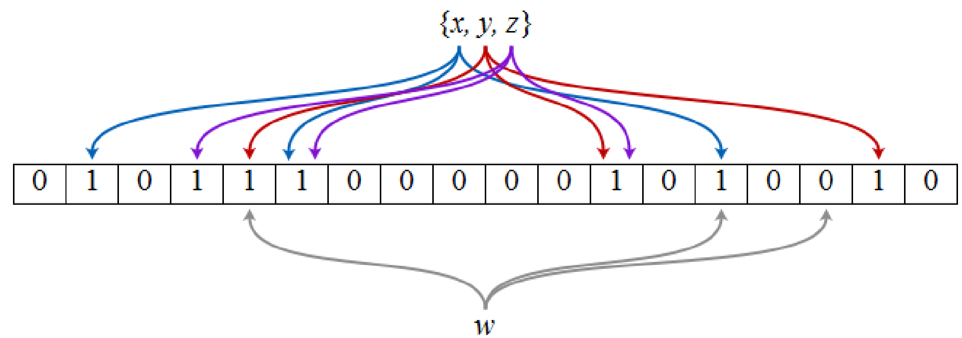
- 10^{15} rendezvous identifiers (256-bit RiD)
- 10^{10} scope identifiers (256-bit SiD)
- Forwarding identifies (256-bit FiD)



The role of Bloom and family

- Well known Bloom filters
  - Efficient *data aggregators*
  - Performance:

    f (memory / # elements)

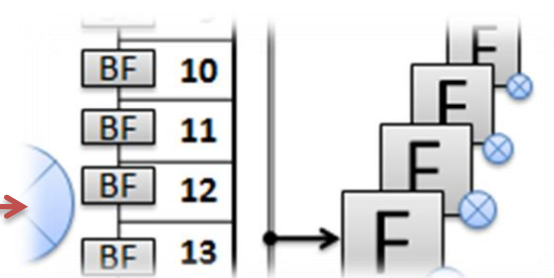


- High-speed router requirements
  - Low (bounded) packet processing *time* (constant time to hash)
  - High-speed *memory* limitations
- Our first *naïve* p-bank Bloom-filter-based switching approach:
  - **Bloom filter *membership-problem***

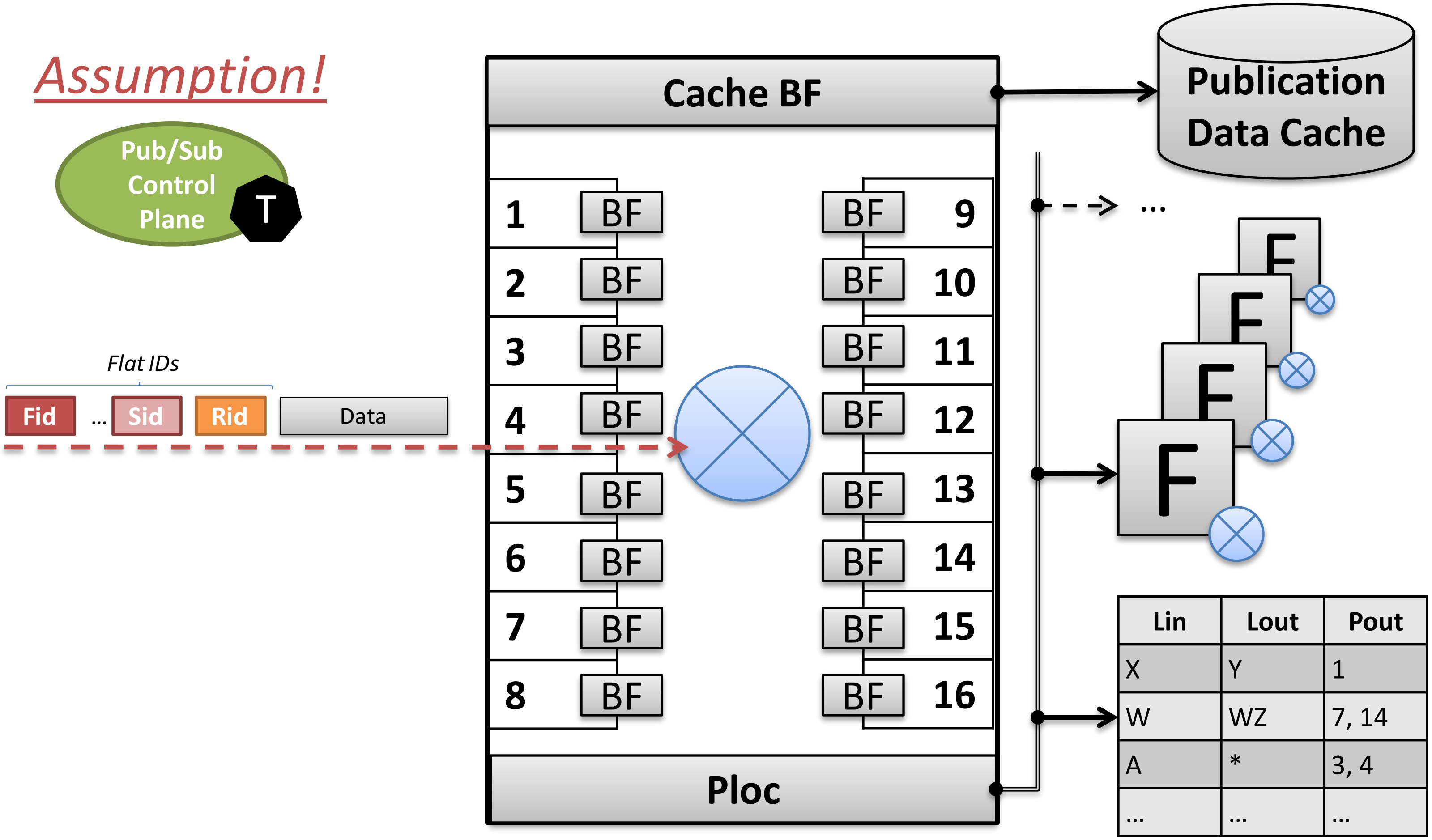
Is $label_x$ in outport P_y?

110010100110010 … 011111001010

?



SPSwitch– Abstract Switching Element



Lin	Lout	Pout
X	Y	1
W	WZ	7, 14
A	*	3, 4
…	…	…

Given an incoming packet identified by a flat ID, which is the output port/interface?

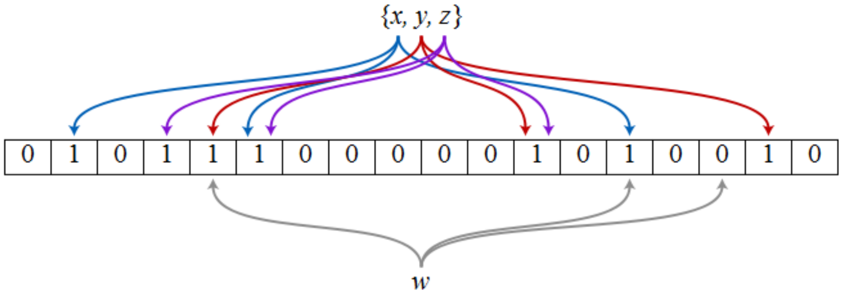
Limitations of standard BFs

a) lack of associated *values*

b) expensive *deletion*

c) no notion of *time*

d) *unbalanced* usage of memory per outport

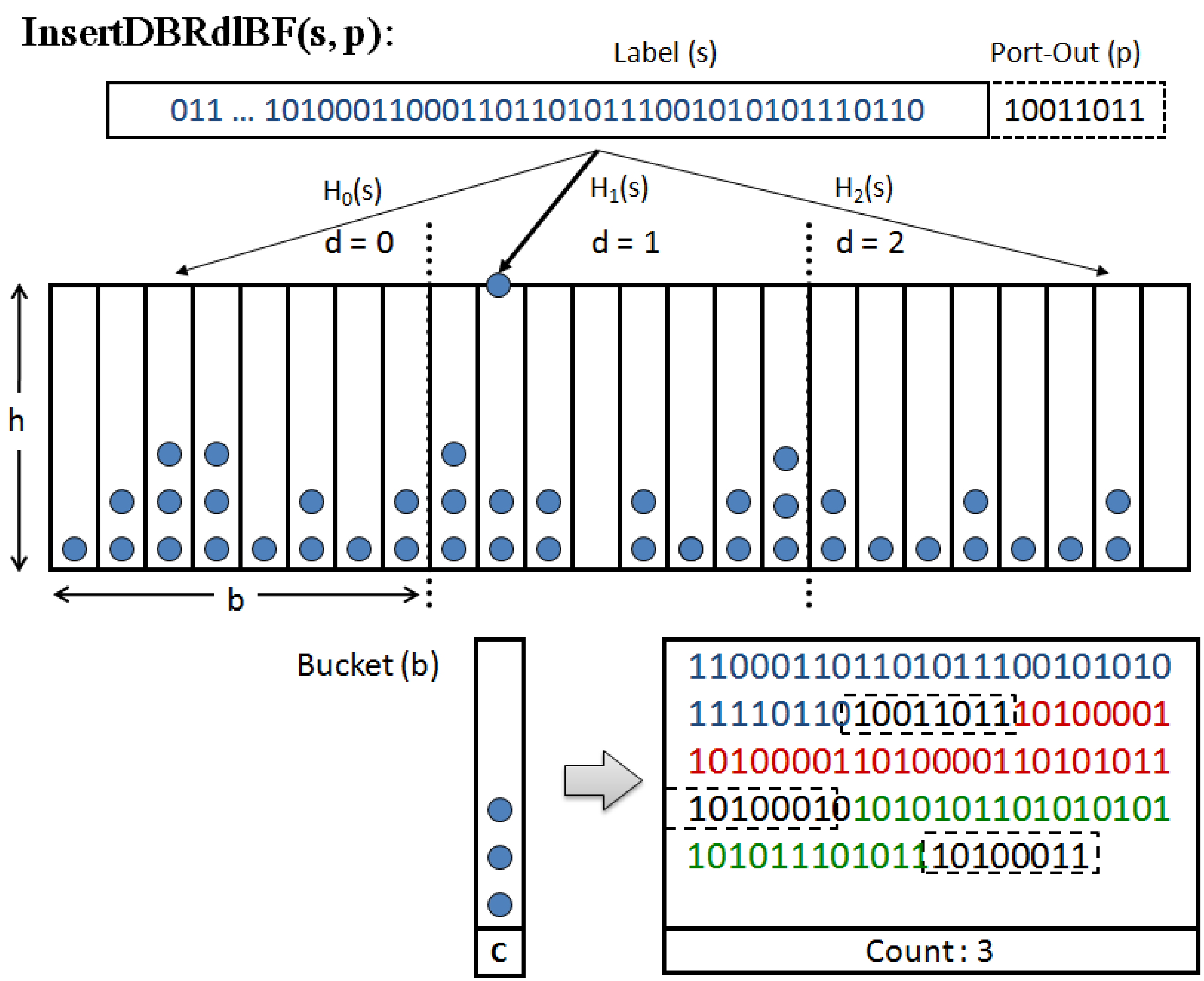


*We need a more **flexible** (probabilistic) data structure!*

→ d-left Fingerprint Compressed Filter (FCF)*

*recent results by Bonomi et al (2006)

Statefull d-left FCF approach



Experimental results

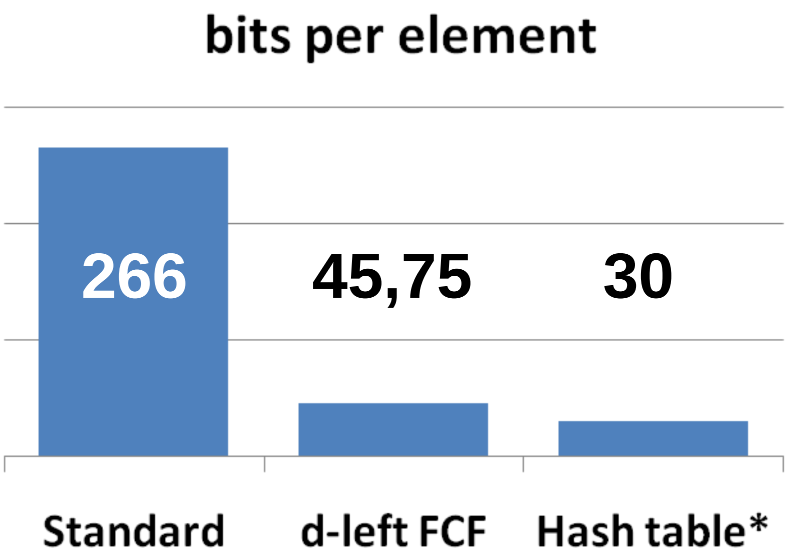
Table 2: Analytical and experimental comparison of different data structures for the switching procedures.

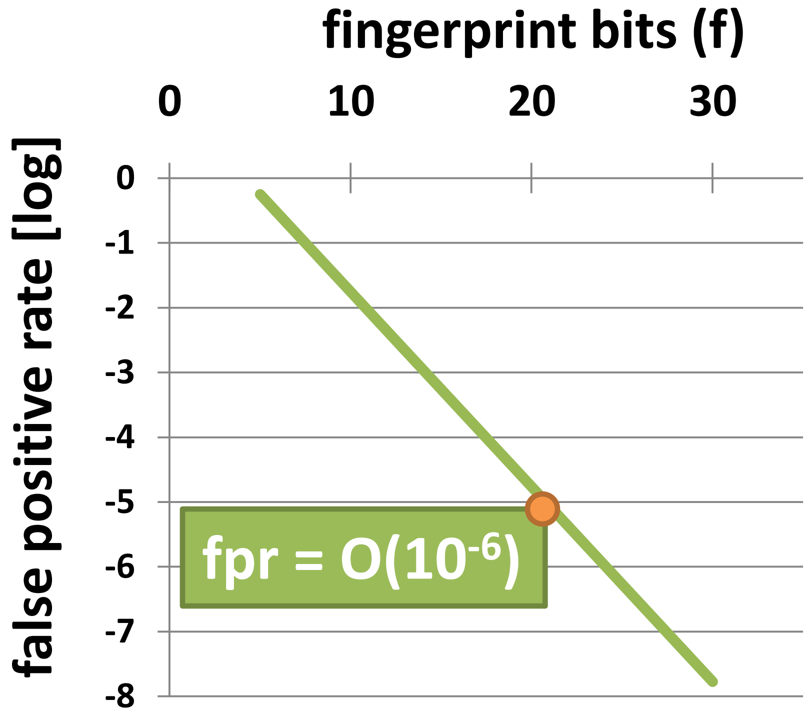
	Mem. access	Mem. size M	(Mbits)**	(bpe)	False positive	(predicted)**	(actual)**
Standard Table	$O(n) — O(1)$*	$n * (s + p)$	253.68	266.0	0	0	-
Fingerpr. Table	$O(n) — O(1)$*	$n * (f + p)$	28.61	30.00	2^{-f}	$9.54 * 10^{-7}$	-
p-bank BF	$O(1)$	$2^P * m$ ***	43.63	45.75	$\approx 2^P * 0.62^{M/n}$	$2.91 * 10^{-7}$	$4.33 * 10^{-3}$
d-left FCF	$O(1)$	$d * b * h * (f + p)$	42.92	45.00	$< d * h * 2^{-f}$	$1.72 * 10^{-5}$	$1.51 * 10^{-5}$
d-left FCF DBR	$O(1)$	$d * b * (h * (f + p) + c)$	43.63	45.75	$< d * h * 2^{-f'}$	$3.57 * 10^{-6}$	$3.46 * 10^{-6}$

* Assumes a perfect hash function. ** Parameters: $n = 1.000.008; d = 3; b = 83.334; f = 20; p = 10; h = 6; c = 3; s = 256$.

*** Total memory of the p-bank Bloom filters equal to the value M of the d-left FCF DBR. $m = M/2^P; k_{opt} = 31$.

20-bit fingerprint + 10-bit port





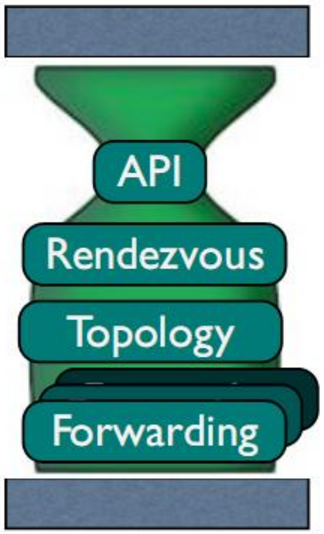
Future work

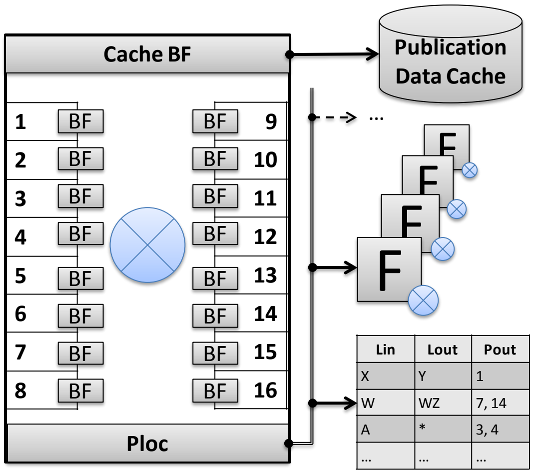
NetFPGA

OneLab 2

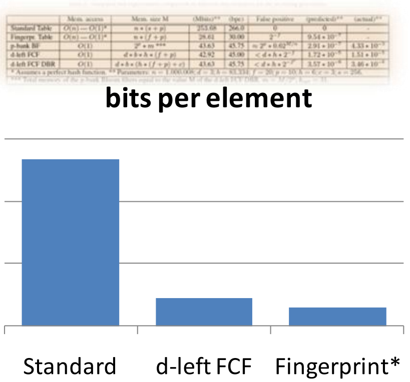
PSIRP
PUBLISH-SUBSCRIBE
INTERNET ROUTING
PARADIGM

Conclusions





SPSwitch



Q&A procedure