

› CAN SDN HELP IN REALIZING ICN MIGRATION SCENARIOS?

Piotr Zuraniewski, Borgert van der Kluit, Ray van Brandenburg

TNO innovation
for life

FOCUS: SDN AS ENABLER

- › This talk is focused on: Software Defined Networking (SDN) as enabler for services not available before (or at least: difficult to realize before)
- › This talk is **not** focused on: is Information Centric Networking good or bad, will ICN succeed, can't we just use DNS instead etc.
- › Small live demo at the end

FUNDAMENTALLY, THE INTERNET IS STILL A TELEPHONY NETWORK



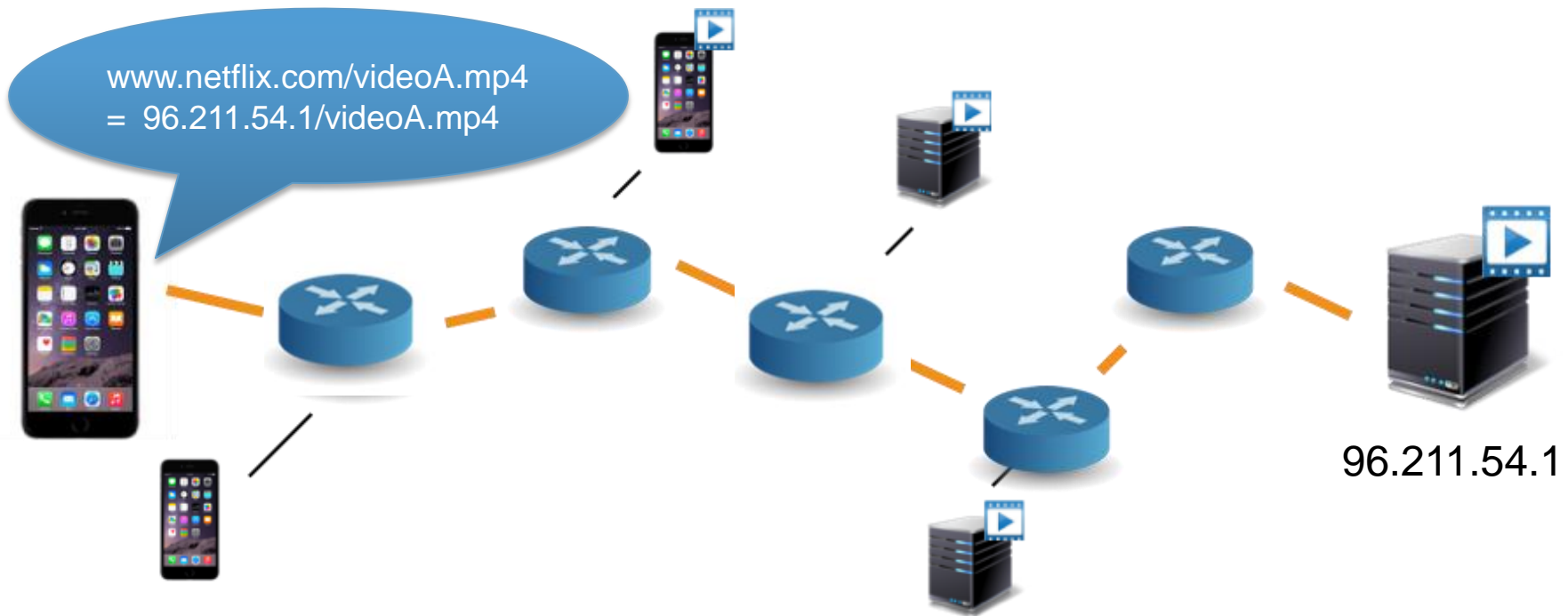
You want to call someone:
you find his telephone number



You want to download a file:
you find its ~~telephone number~~ IP address

HOST-ORIENTED

HOST-ORIENTATION PROBLEM FROM DISTRIBUTION CONTENT PERSPECTIVE



From a “distribution content” point of view:

- › not efficient
- › not flexible
- › not secure (connection-oriented)

ALTERNATIVE TO HOST-ORIENTATION: INFORMATION-CENTRIC NETWORKING

Named data instead of the servers on which it is stored

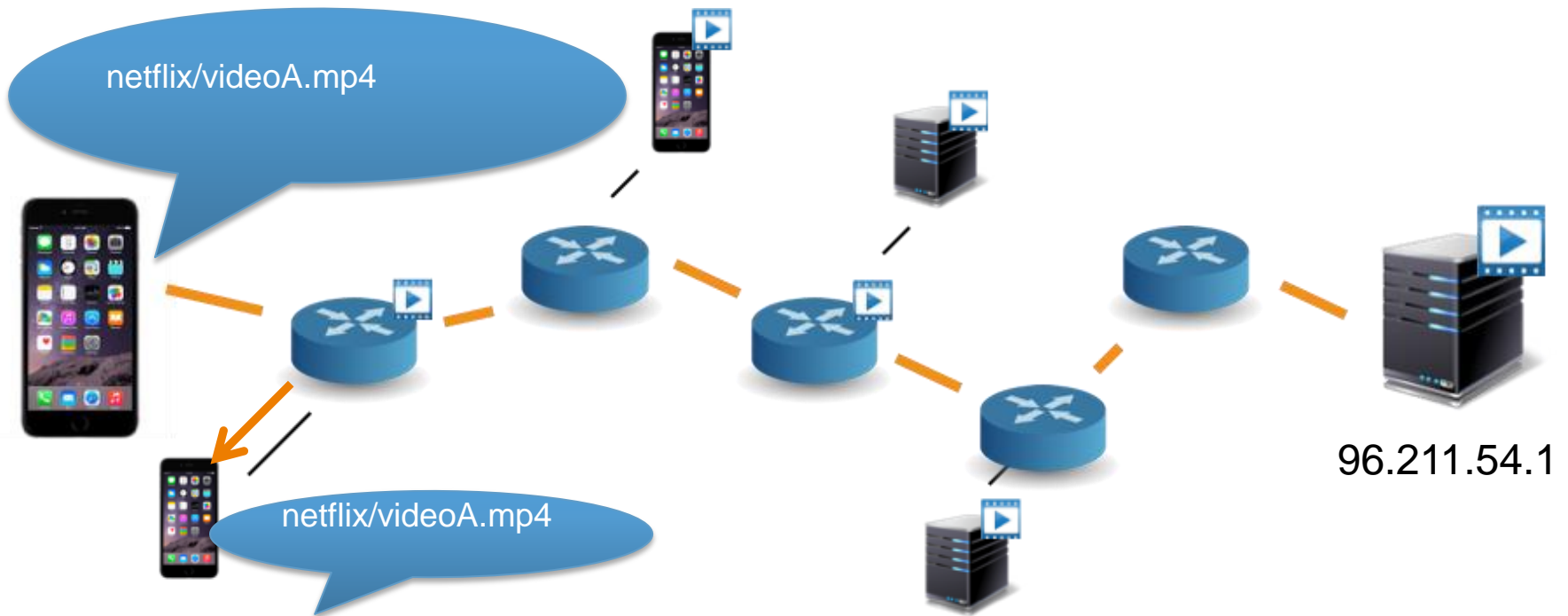
User: Hey, Network, I am interested in this information:

[netflix/videoA.mp4](#)

Network: OK, User. I will find out where it is stored and

deliver data chunks you need

ICN: CONTENT CHUNKS INSTEAD OF HOSTS



From a “distribution content” point of view:

- › efficient: caching possible
- › secure (data chunks signed)
- › network-agnostic

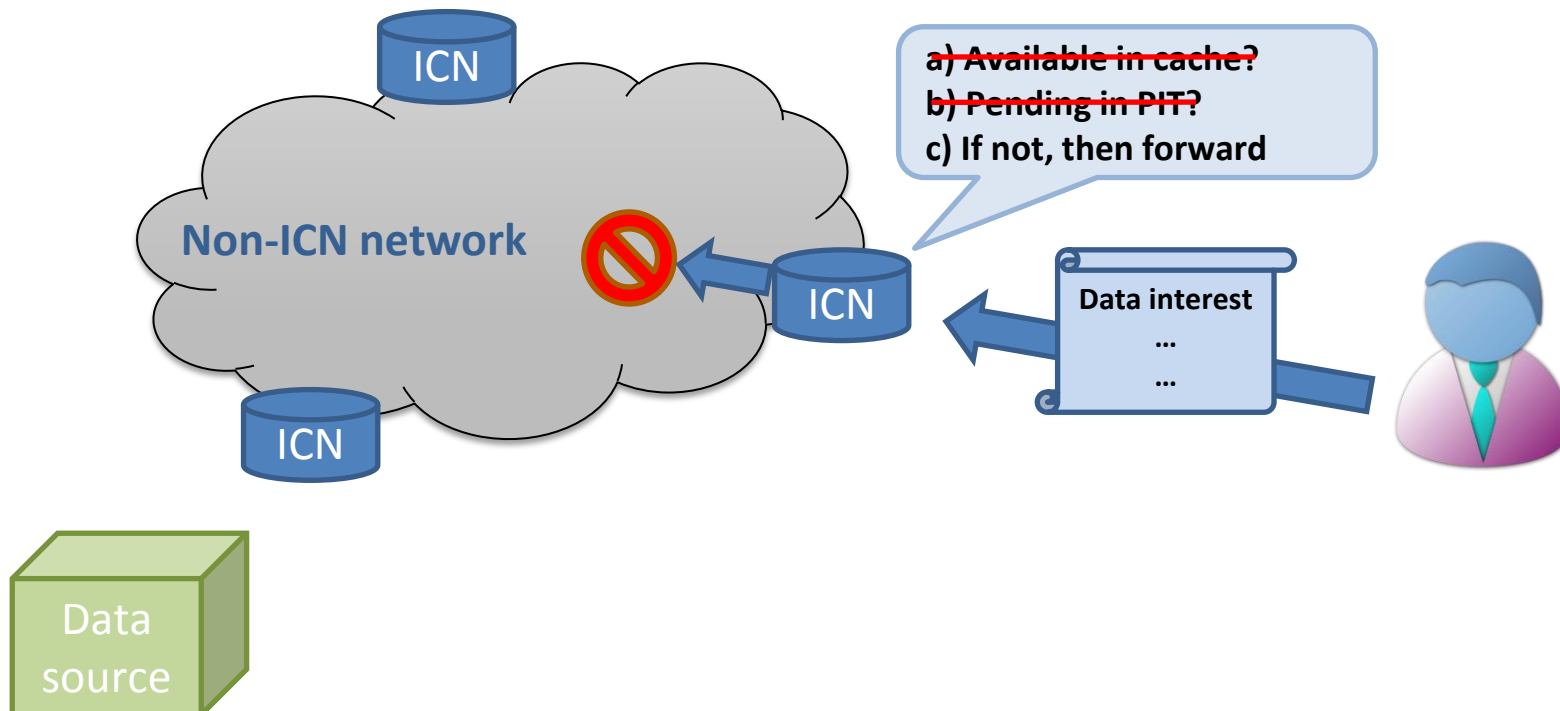
WILL ICN SUCCEED?

- › We are not sure 😊
 - › If ICN succeeds, it will be a fundamental change to the Internet since the introduction of TCP/IP
 - › Many big players are involved, 5G-PPP to embrace it
- › We are, however, sure that transition (if ever happens) will not be overnight (see ipv6 casus)
 - › Good migration scenarios needed
 - › **Can SDN help in realizing migration scenarios?**

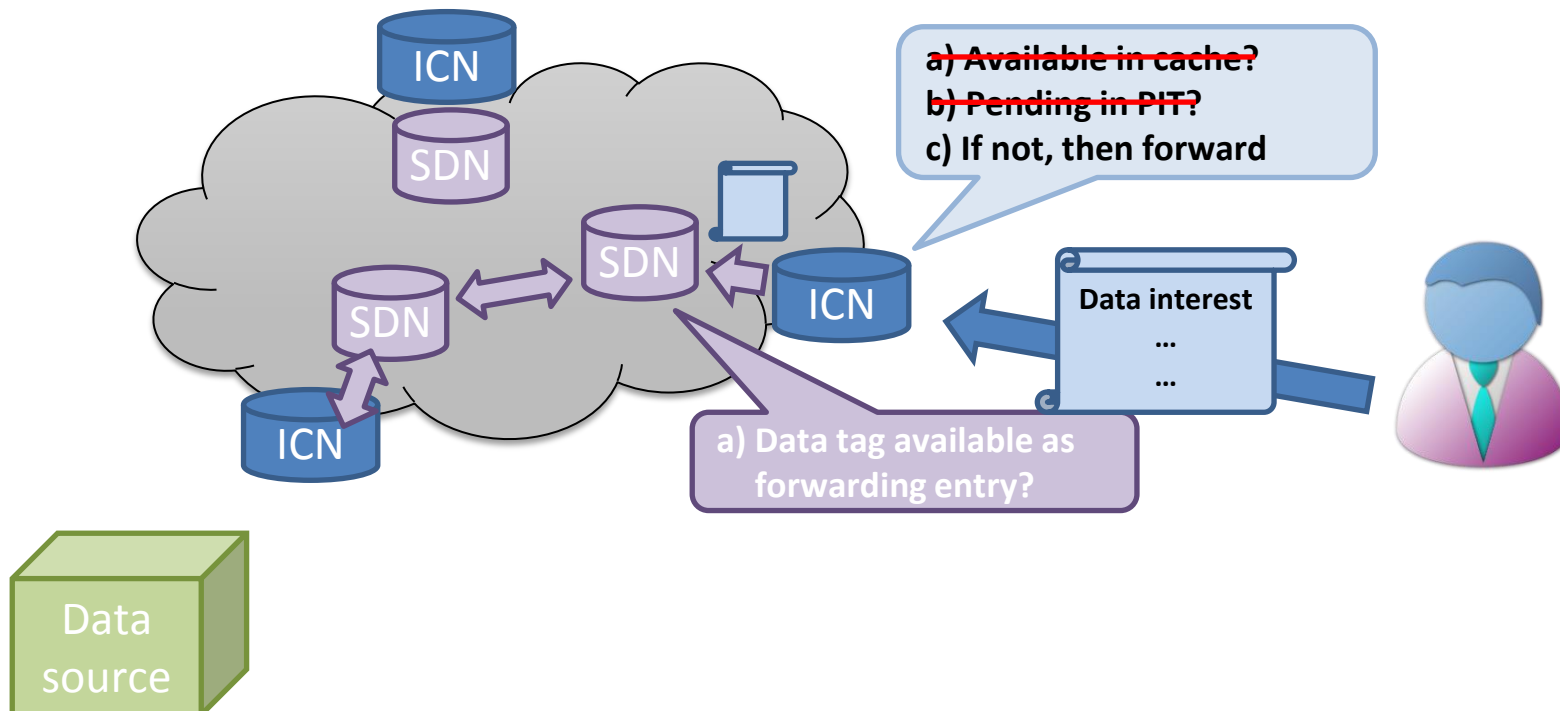


CONNECTING ICN „ISLANDS”

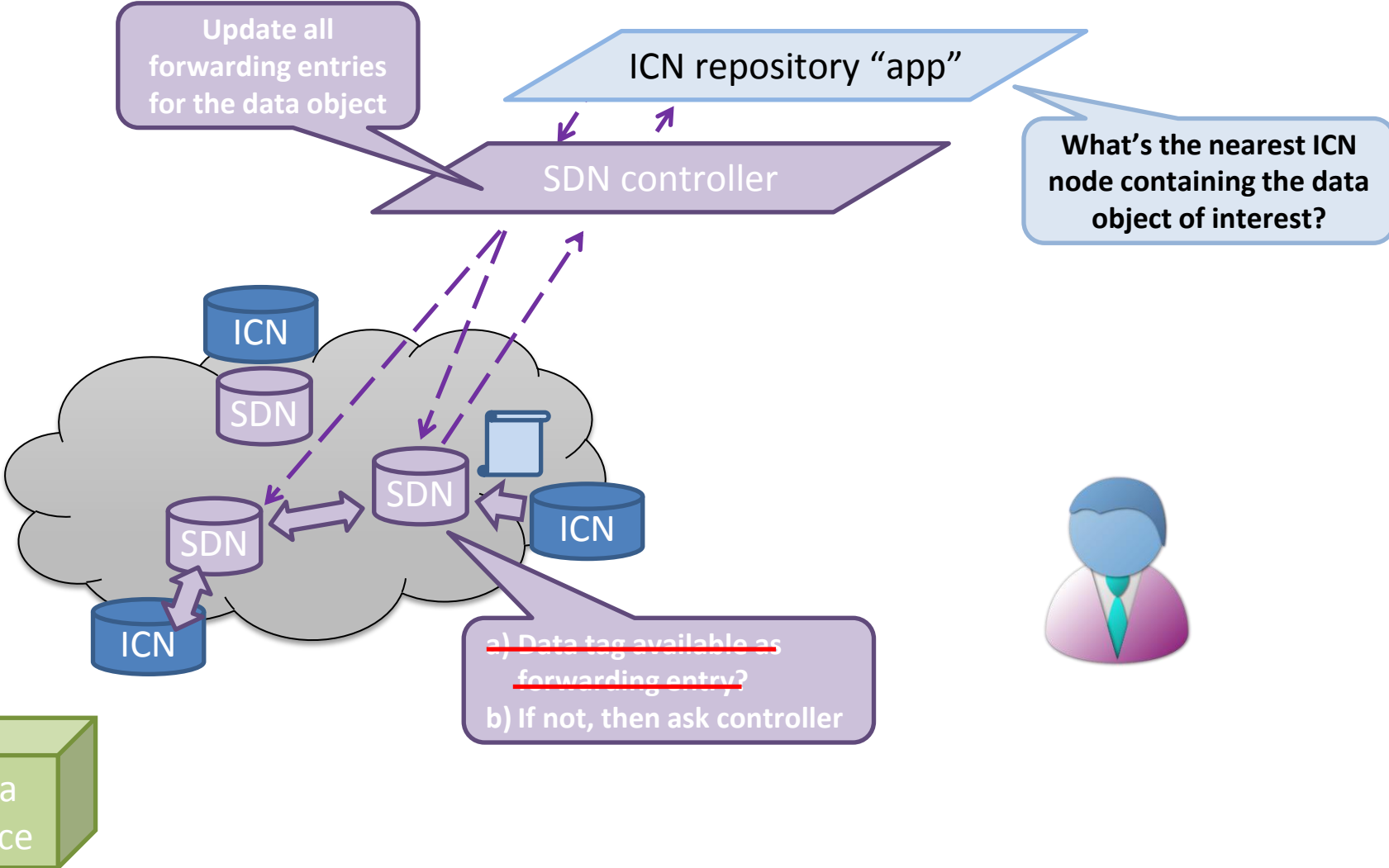
- › ICN „islands” separated by traditional network
- › Manual tunnels set-up to enable communication: tedious, error-prone etc.



SDN ENHANCED ICN FORWARDING



SDN ENHANCED ICN FORWARDING



ASKING CONTROLLER IS DIFFICULT WITH ICN DUE TO TLV

- › Current OpenFlow switches match on fixed fields (e.g., MAC, VLAN, IP)
- › However, by its very nature, ICN packets have unpredictable length
 - › /tno/piotr – 10B interesting
 - › /tno/borgert – 12B interesting
- › Thus it is organized in (Type, Length, Value) format
 - › ...matching on TLV is not handled by default by OF switches

- › ...but here we come 😊

FLEXIBLE OPENFLOW MATCH

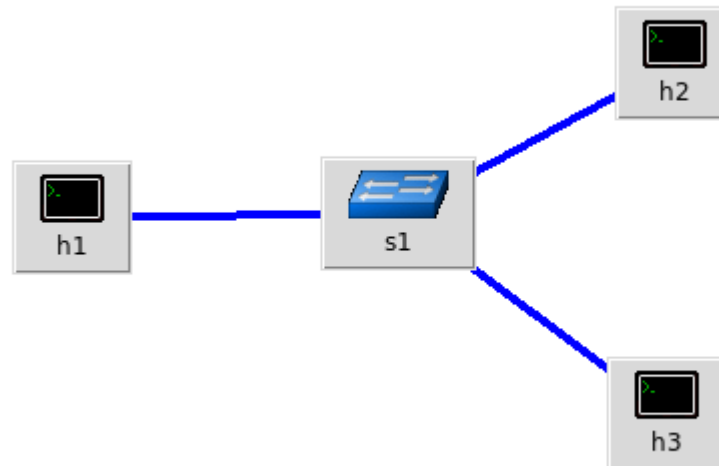
- › ACK: Jouet, S., Cziva, R., Pezaros, D.; Arbitrary Packet Matching in OpenFlow
 - › Uses Standard OpenFlow Extensible Match (OXM)
 - › Sends Berkeley Packet Filter (BPF) program to the controller
 - › This program is executed on a switch to perform match action
 - › However, length of such program is limited to 240B – insufficient for ICN
 - › Not only lengthy name must be carried but also the whole ICN packet parsing program itself

```
ldh [16] ; Load Ethernet type field
jeq #0x0800, Keep, Drop ; Check ethertype against 0x0800
Keep: ret #0x0000ffff ; Return full packet
Drop: ret #0 ; Discard packet
```

TNO CONTRIBUTION – OPENFLOW MATCH ON (ALMOST) ANYTHING

- › Our approach
 - › Also use BPF to match on required part of datagram carrying ICN packet
 - › ...but upload BPF program with OpenFlow Experimental Messages which allow for much longer BPF programs (up to 64kB)
 - › OXM points to the previously uploaded BPF program
 - › Long and complex arbitrary matches are possible (incl. ICN) on network demand
 - › Solution is OpenFlow standard compliant

DEMO



PLANS

- › We modified (as J-C-P) SoftSwitch to execute our BPF program
 - › We plan to modify OpenVSwitch (*de facto* VS standard) to do so
- › We used „plain” BPF
 - › We plan to verify new possibilities eBPF (extended BPF) gives
- › Performance & scalability tests to be done to embed it in BigData context
 - › Scalable retrieval of large amounts of popular (possibly ‘mutated’) and sensitive data
 - › Network decides in autonomous way about which paths to provision and what parameters (e.g., QoS) they should have