Implementing IPv6 Segment Routing in the Linux Kernel

David Lebrun, Olivier Bonaventure

ICTEAM, UCLouvain

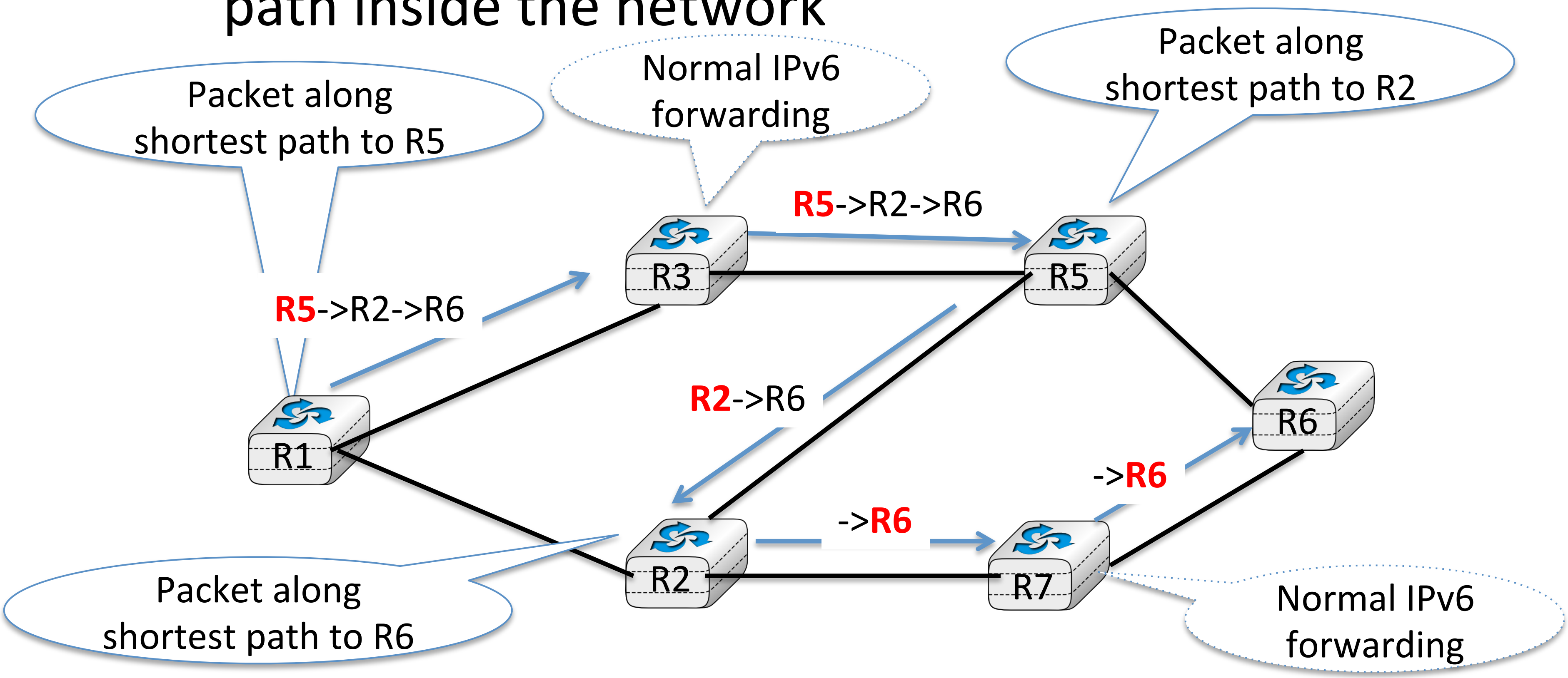
Work supported by ARC grant 12/18-054 (ARC-SDN) and a Cisco grant

Agenda

- IPv6 Segment Routing
- Implementation in the Linux kernel
- Performance evaluation

What is Segment Routing ?

- The return of Source Routing
  - Each packet contains a loose route to encode any path inside the network



IPv6 Segment Routing

- Basic principles
  - IGP distributes IPv6 prefixes and router loopback addresses
  - Loose source route encoded inside IPv6 extension header containing a list of segments
  - Main types of segments
    - Node segment (router's loopback address)
    - Adjacency segment (router outgoing interface)
    - Virtual function (operator defined function)

http://www.segment-routing.net

https://tools.ietf.org/html/draft-ietf-6man-segment-routing-header-06

The IPv6 Segment Routing Header

Remaining segments

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Header   |  Hdr Ext Len  | Routing Type  | Segments Left |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Last Entry   |     Flags     |              Tag              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            Segment List[0] (128 bits IPv6 address)            |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                                                               |
                              ...
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            Segment List[n] (128 bits IPv6 address)            |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                                                             //
//         Optional Type Length Value objects (variable)       //
//                                                             //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

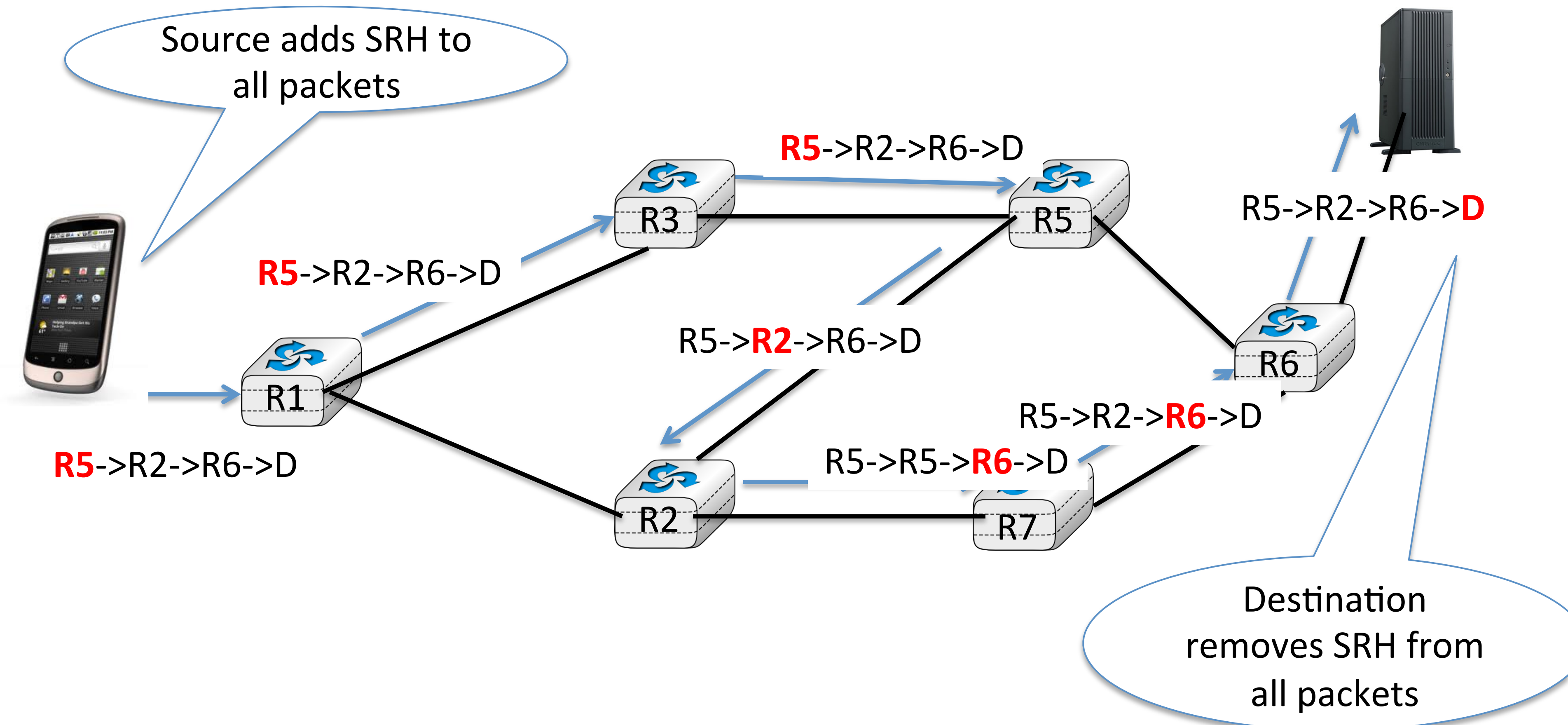
Index of last segment

Each segment is one IPv6 address

Extensibility

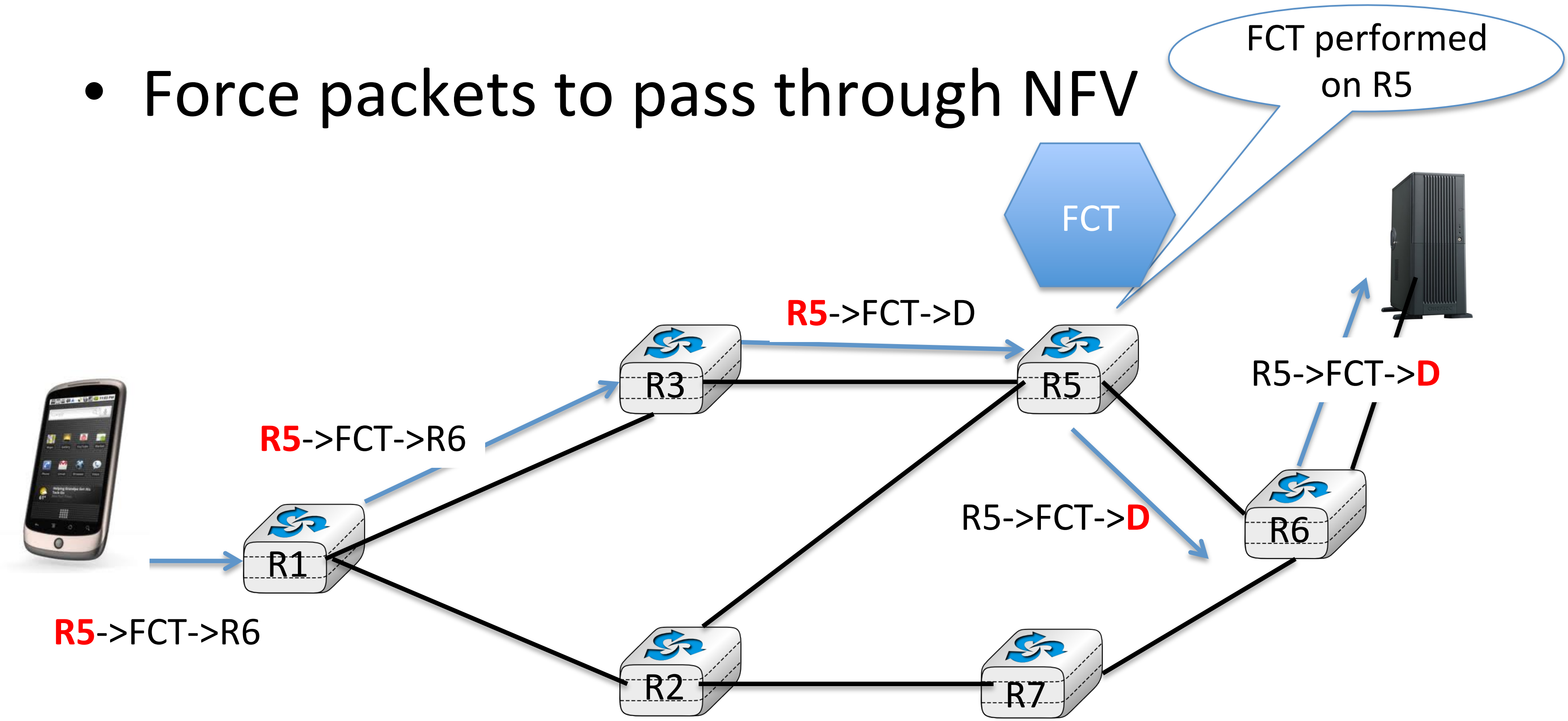
IPv6 Segment Routing use cases

- Paths controlled by the endhosts



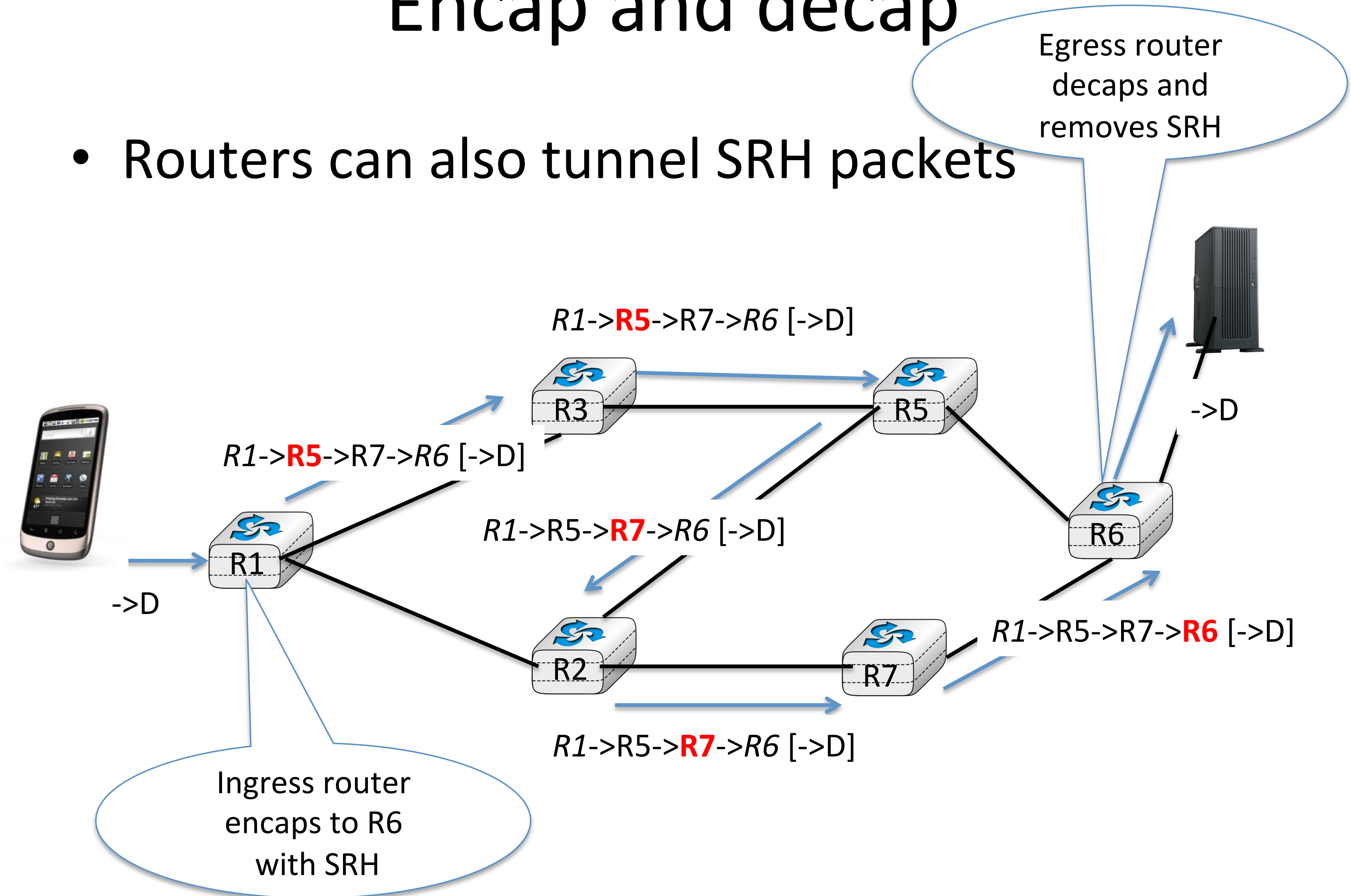
Network Function Virtualisation

- Force packets to pass through NFV



Encap and decap

- Routers can also tunnel SRH packets



Security: Learning from the past

- How to avoid past failures of source routing ?

Security Problems in the TCP/IP Protocol Suite

*S.M. Bellovin**
smb@ulysses.att.com

AT&T Bell Laboratories
Murray Hill, New Jersey 07974

ABSTRACT

The TCP/IP protocol suite, which is very widely used today, was developed under the sponsorship of the Department of Defense. Despite that, there are a number of serious security flaws inherent in the protocols, regardless of the correctness of any implementations. We describe a variety of attacks based on these flaws, including sequence number spoofing, routing attacks, source address spoofing, and authentication attacks. We also present defenses against these attacks, and conclude with a discussion of broad-spectrum defenses such as encryption.

Network Working Group J. Abley
Request for Comments: 5095 Afilias
Updates: 2460, 4294 P. Savola
Category: Standards Track CSC/FUNET
G. Neville-Neil
Neville-Neil Consulting
December 2007

Deprecation of Type 0 Routing Headers in IPv6

Status of This Memo

Abstract

The functionality provided by IPv6's Type 0 Routing Header can be exploited in order to achieve traffic amplification over a remote path for the purposes of generating denial-of-service traffic. This document updates the IPv6 specification to deprecate the use of IPv6 Type 0 Routing Headers, in light of this security concern.

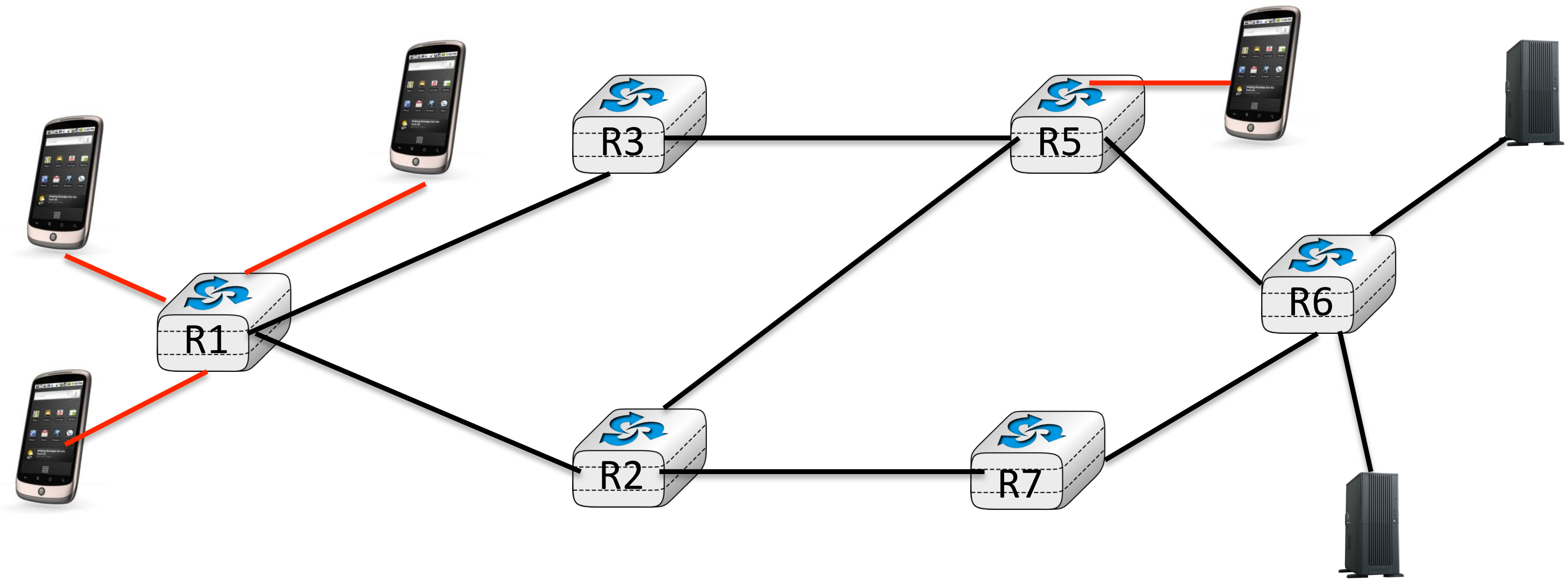
The IPv6 SRH HMAC TLV

Different keys and
different hash
functions can be used

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3     6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-     -+-+-+-+-+-+-+
|      Type     |    Length     |          RESER  D             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-  -+-+-+-+-+-+-+-+
|                  HMAC Key ID (4 octets)                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                              //
|                  HMAC (32 octets)                            //
|                                                              //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Utilisation of the HMAC TLV

- All routers are configured with an HMAC key

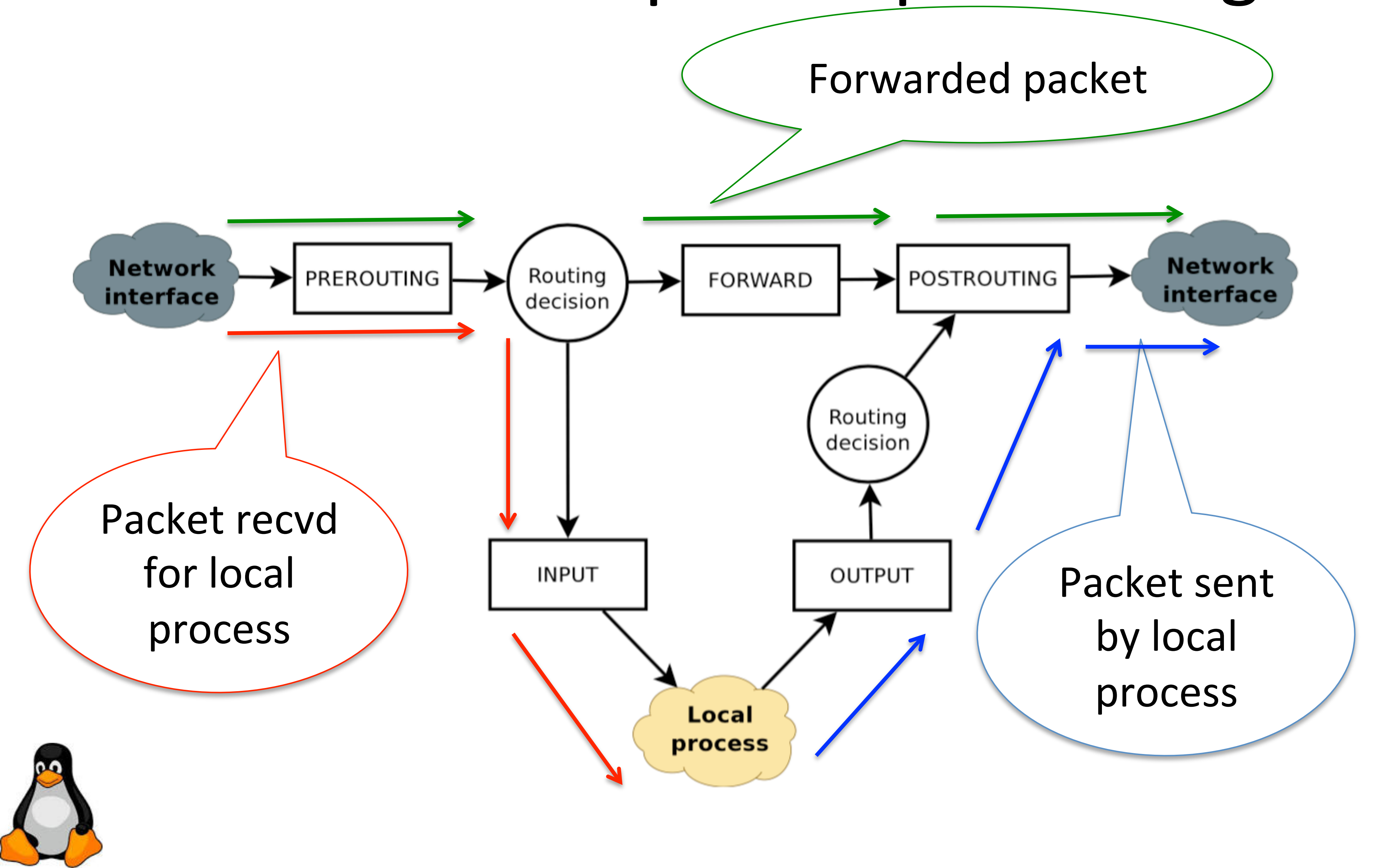


- Clients receive SRH with HMAC key
  - E.g. from SDN controlled
- Trusted servers configured with HMAC key

Agenda

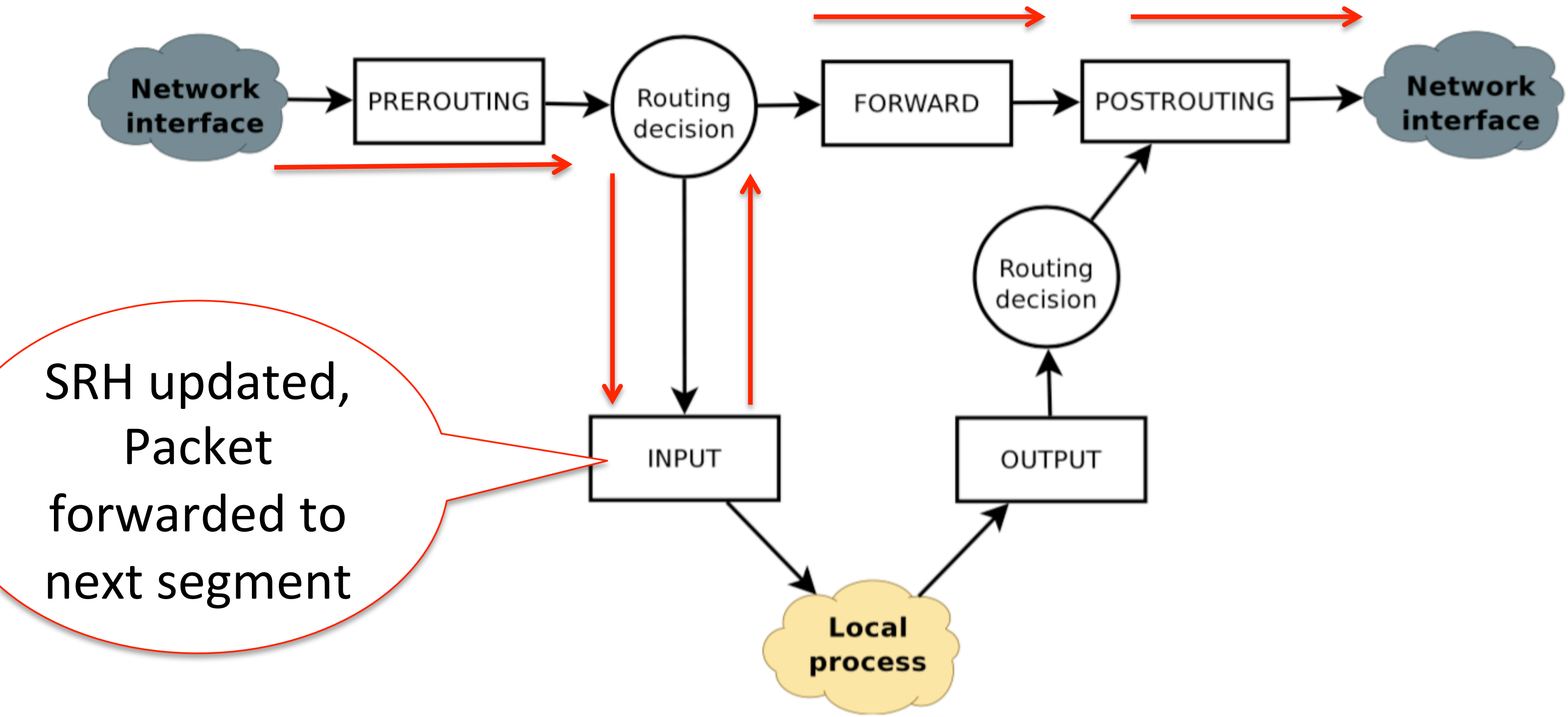
- IPv6 Segment Routing
- Implementation in the Linux kernel
- Performance evaluation

Basics of Linux packet processing



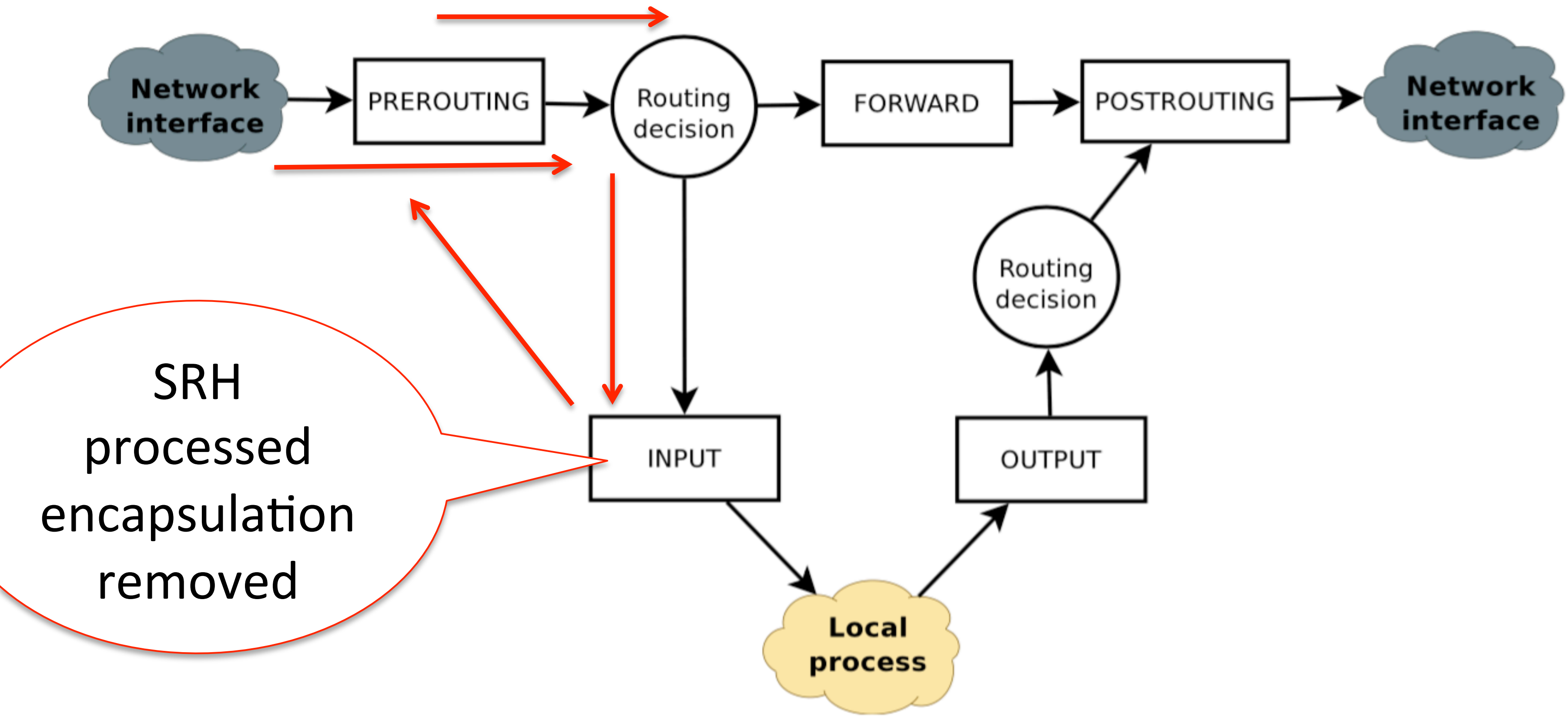
Packet forwarding with IPv6 SR

- Router is one of the segments in the list



Packet forwarding with IPv6 SR

- Egress router receiving encapsulated packet



How to configure IPv6 SR ?

- IPv6 SR implementation extends iproute2
  - Commands passed through rtnetlink
  - Example

Destination match

```
ip -6 route add fc42::/64
       encap seg6 mode encap
       segs fc00::1,2001:db8::1,fc10::7
       dev eth0
```

SRv6 encap

Segments added in the encapsulated packet

SRH usage by applications

- Endhosts can control the SRH on a per flow basis through the socket API

```
struct ipv6_sr_hdr *srh;
int srh_len;

srh_len = build_srh(&srh);
fd = socket(AF_INET6, SOCK_STREAM, IPPROTO_TCP);
setsockopt(fd, IPPROTO_IPV6, IPV6_RTHDR, srh, srh_len);
```

HMAC processing

- Three modes of operations can be configured
  - Ignore
    - All packets are forwarded independently of the HMACs
  - Verify
    - Packets containing an HMAC are processed if HMAC is valid
    - Packets without HMAC are processed
  - Enforce
    - Packets containing an HMAC are processed if HMAC is valid
    - Packets without HMAC are processed

Agenda

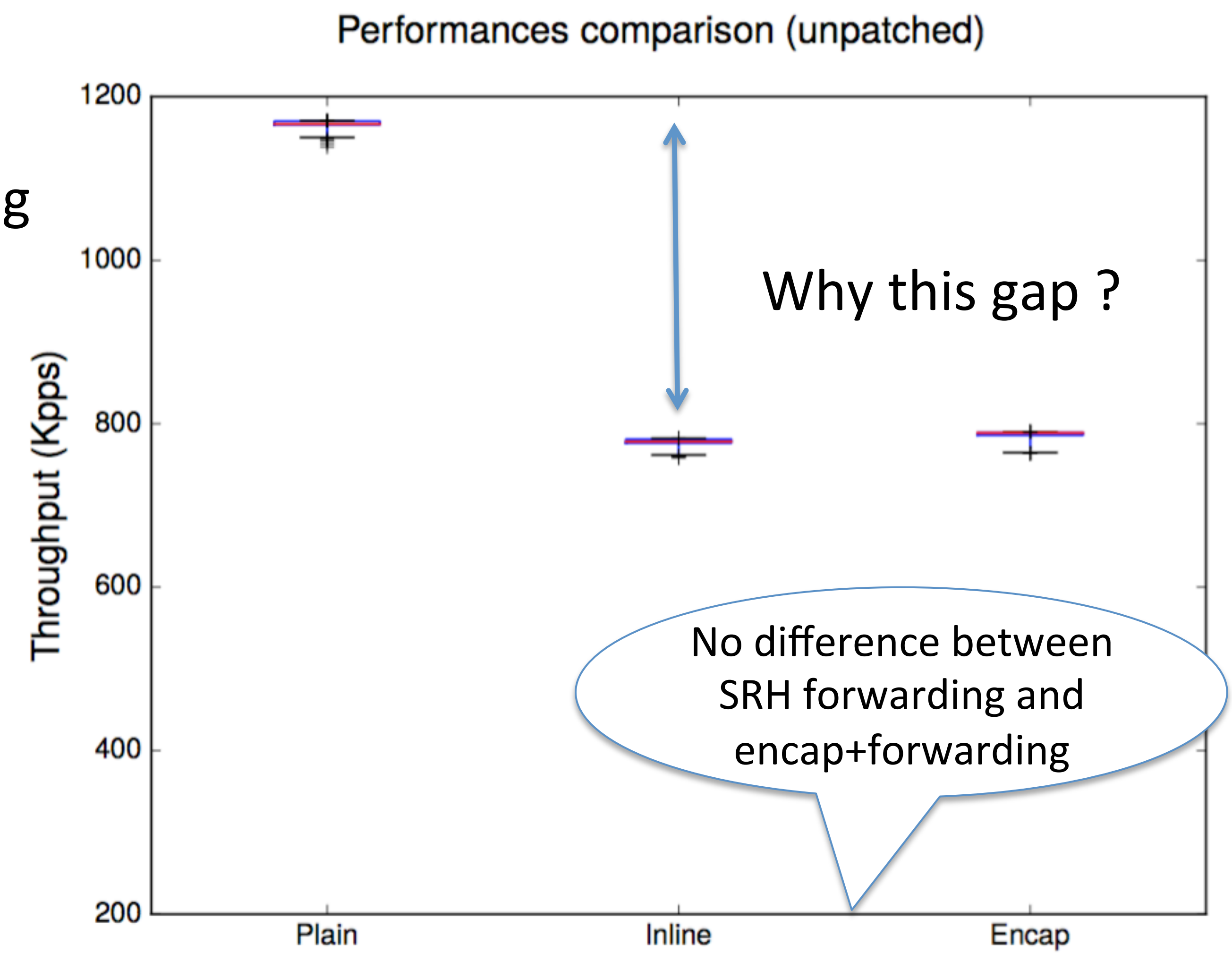
- IPv6 Segment Routing
- Implementation in the Linux kernel
- Performance evaluation

Lab measurements

- Lab setup
  - Intel Xeon X3440 processors (4 cores 8 threads at 2.53 GHz
  - 16 GB of RAM
  - two Intel 82599 10 Gbps Ethernet
    - One queue per CPU, one IRQ per queue
  - Linux kernel 4.11-rc3, TSO and GRO disabled
- Traffic generator
  - Pktgen, in-kernel module sending UDP packets

First measurements with one CPU

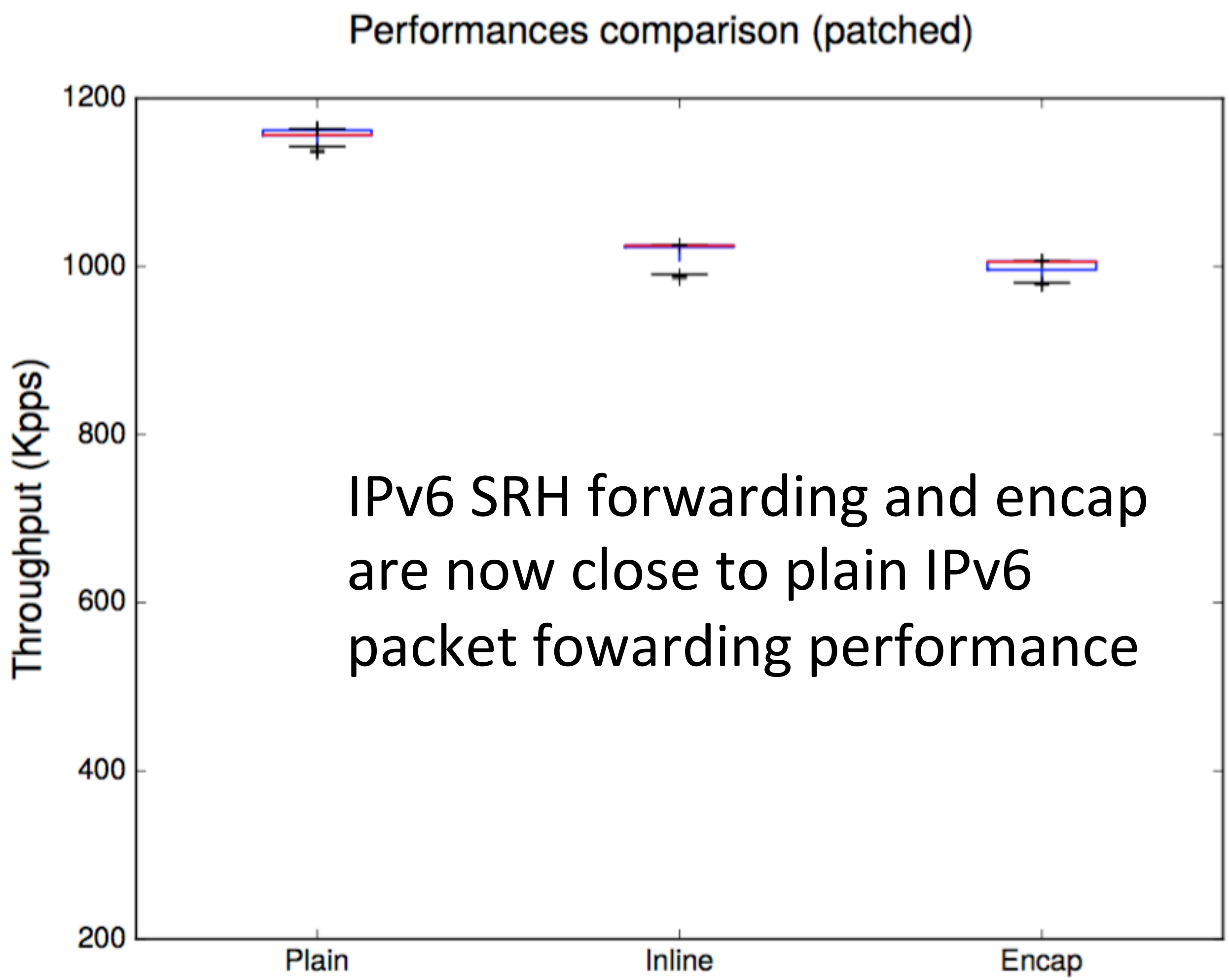
Baseline
Plain IPv6
forwarding



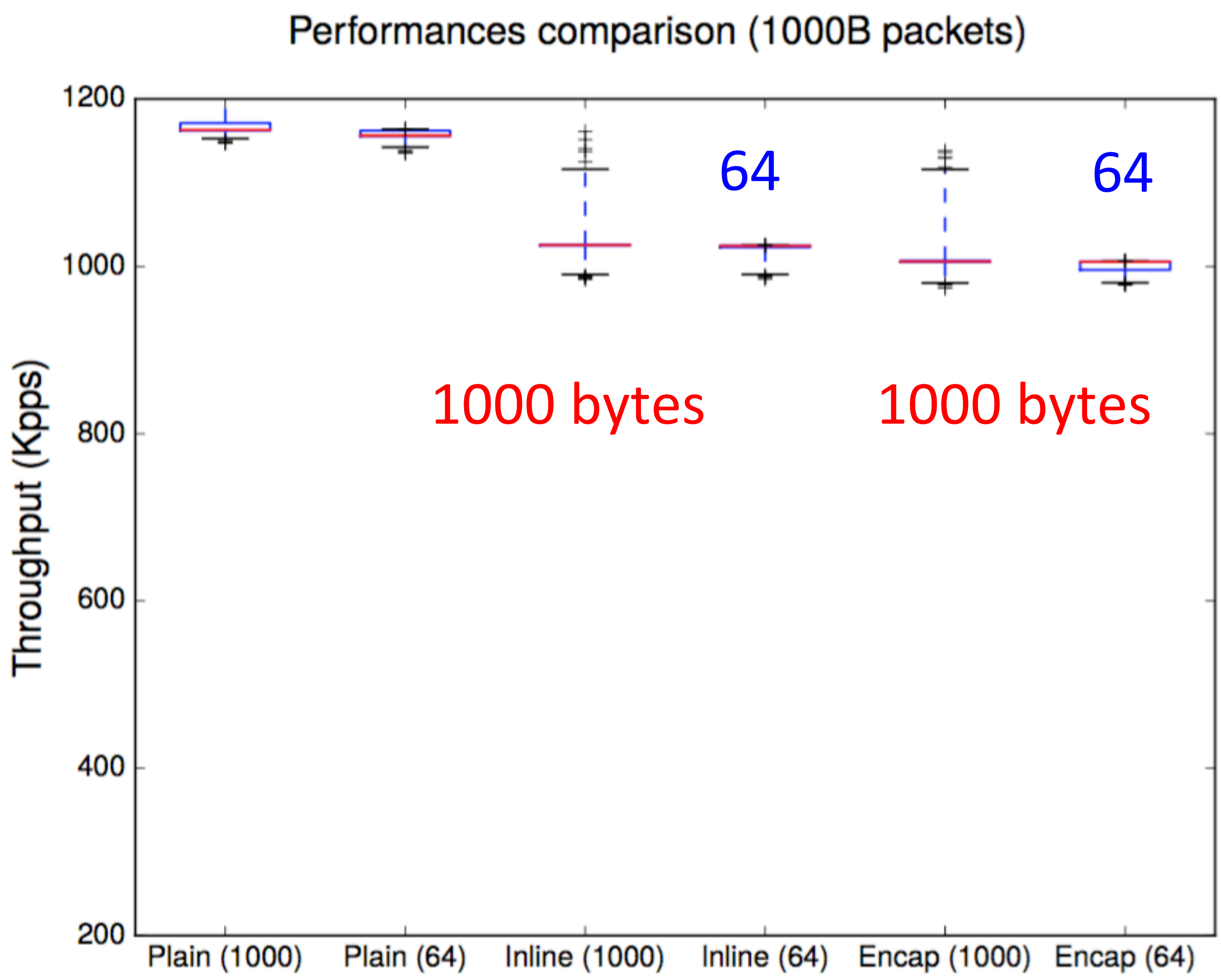
Performance limitations of the first implementation

- Route lookup
  - Destination cache was implemented for locally generated packets but not forwarded ones
    - Fixed with a dest cache
- Issue with memory allocation
  - Forced free to take a slow path involving spinlocks in case packet was processed by different CPU than NIC IRQ
    - Fixed with a better utilisation of the skb

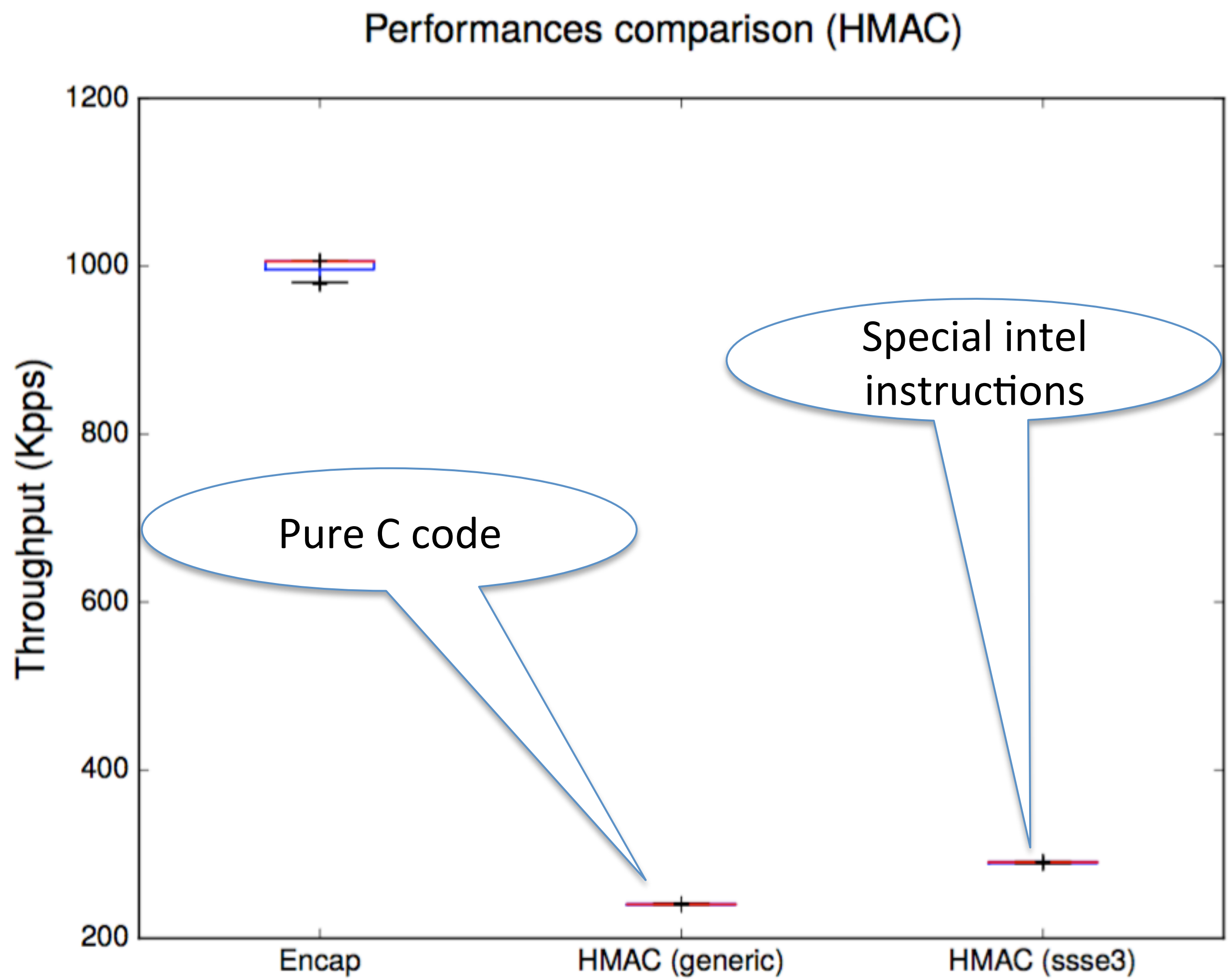
Improved performance on one CPU



Does packet size affect performance ?

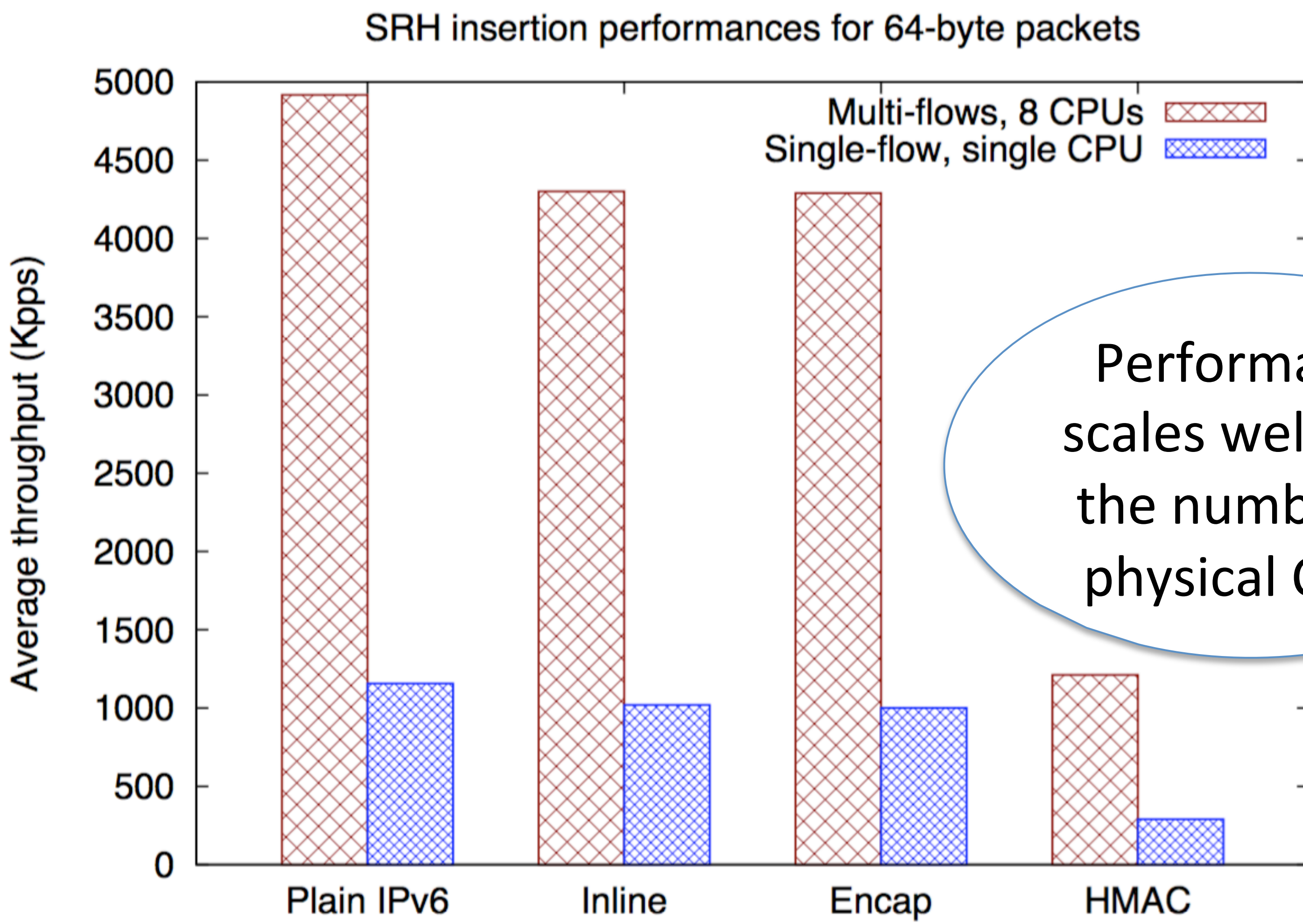


Cost of HMAC



Performances comparison (HMAC)
Throughput (Kpps)
1200
1000
800
600
400
200
Encap
HMAC (generic)
HMAC (ssse3)
Pure C code
Special intel instructions

Leveraging multiple cores



Performance scales well with the number of physical CPUS

Conclusion

- IPv6 Segment Routing has matured
  - Stable specification
  - Various use cases
- Implementation in the Linux kernel 4.11+
  - Endhost functions for clients and servers
  - Router functions
- Performance evaluation
  - Good forwarding and encap/decap performance
  - Unsurprisingly HMAC TLV affects performance

http://www.segment-routing.org