# Stateless ICN Forwarding with P4 towards Netronome NFP-based Implementation

Aytac Azgin, Ravishankar Ravindran, Guo-Qiang Wang

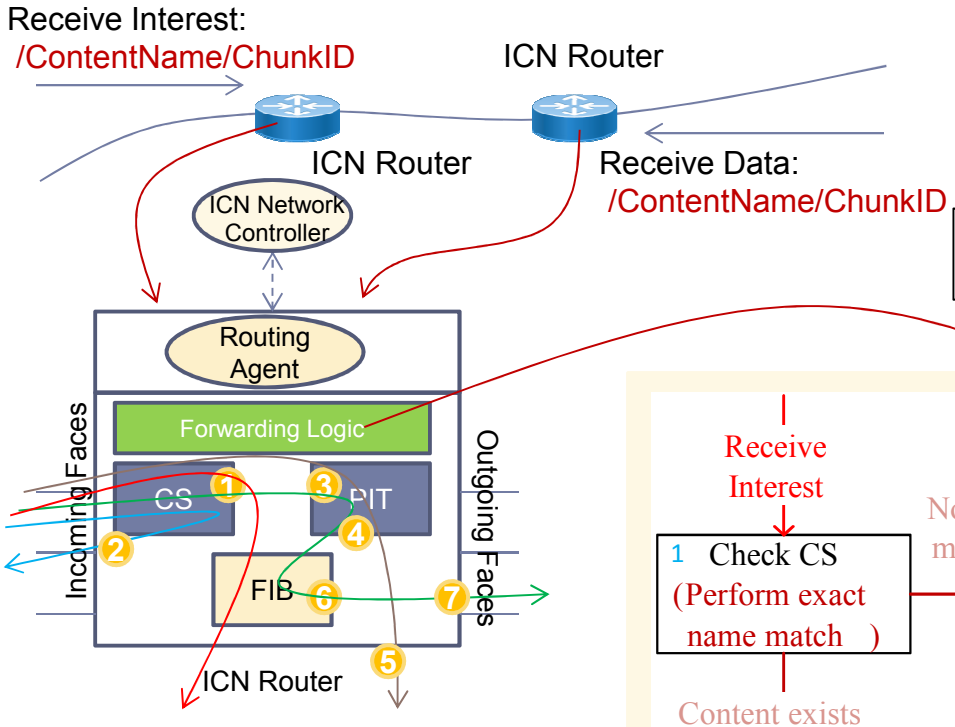
aytac.azgin, ravi.ravindran, gq.wang@huawei.com

Huawei Research Center, Santa Clara, CA

# ICN Summary

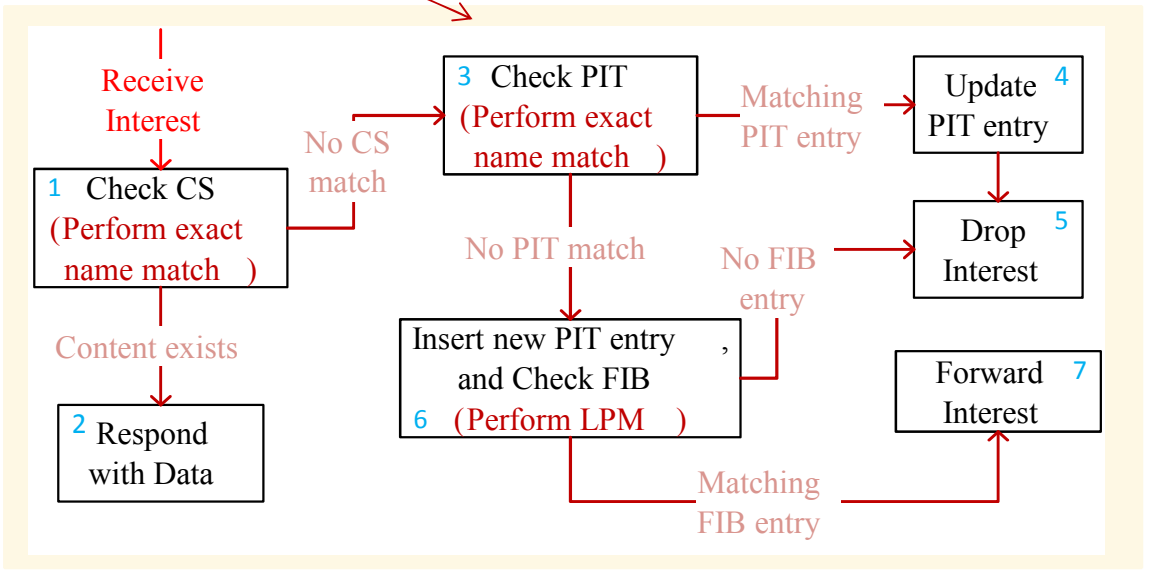
- ICN enables content-centric communication
  - Replaces current IP's host-centric design
  - Content can be fetched from anywhere, irrespective of location
- Various ICN solutions exist
  - Content-centric Networking (CCN) and Named-data Networking (NDN) represent the leading approaches
- CCN/NDN architecture
  - Pull-based solution (i.e., request/response) with Interest/Data primitives
  - Provides additional features:
    - in-network caching using Content Store (CS),
    - stateful forwarding with Pending Interest Table (PIT)
    - integrated security within packets (through signatures)
    - Also supports multicasting, multi-homing, and mobility

# Packet Forwarding in CCN/NDN




| Type (= I) | Content Name | Optional TLV Headers | Interest Message |
|---|---|---|---|

| Type (= D) | Content Name | Optional TLV Headers | Signature TLV | Content Message |
|---|---|---|---|---|

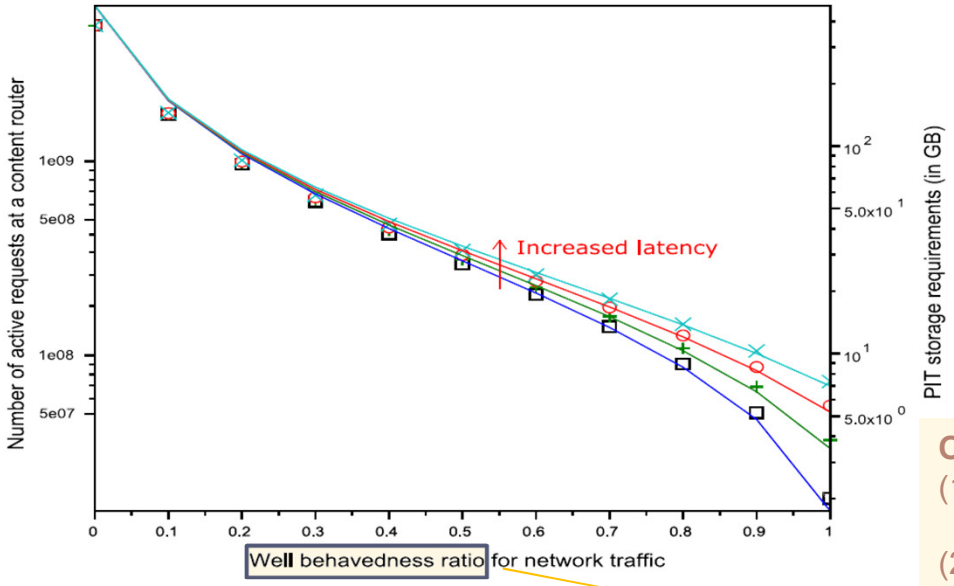


# Stateful Forwarding

- CCN/NDN by default uses **stateful forwarding**
  - In PIT, routers keep information on received requests:
    - content name, incoming/outgoing interfaces, nonces (if implemented), timeout
  - **Stateful forwarding has multiple purposes**
    - Aggregate incoming requests (e.g., same name, different incoming interface and nonce values)
    - Prevent attacks targeting a content name (as requests targeting the same name are suppressed at the edge)
    - Create breadcrumbs for the Data packets (received Data packets are checked with PIT entries for a match)

# Problems with Stateful Forwarding*

- Still, we see concerns with stateful forwarding
  - Aggregation is limited to edges
  - Shown to not fully prevent attacks
  - And introduces additional overhead, in storage and processing




Ratio of Interests that return a Data packet

**Problem:** Interests with no Data return
**What happens:** Entries are stored within PIT, until timeout (~4s)

**Observations:**
(1) Increased memory requirements to represent worst-case scenario
(2) Increased latency to access entries

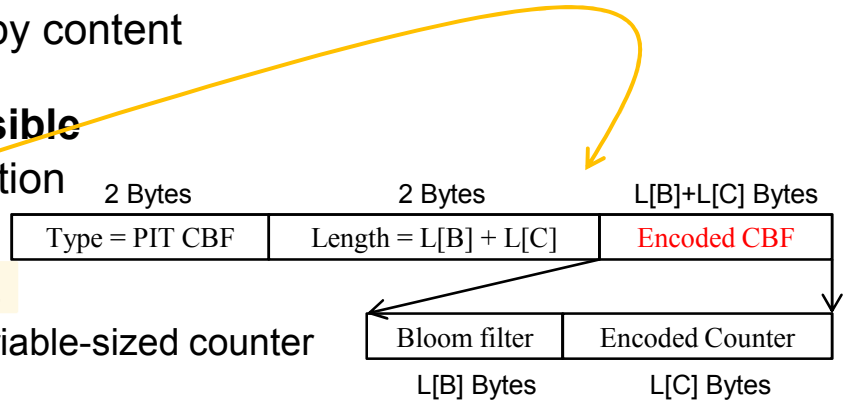
- What remains is the breadcrumb advantage
  - which can be replicated using stateless forwarding with in-packet filters

* "pit/LESS: Stateless Forwarding in Content Centric Networks", A. Azgin, *et al.*

# Stateless Forwarding Design - Choices

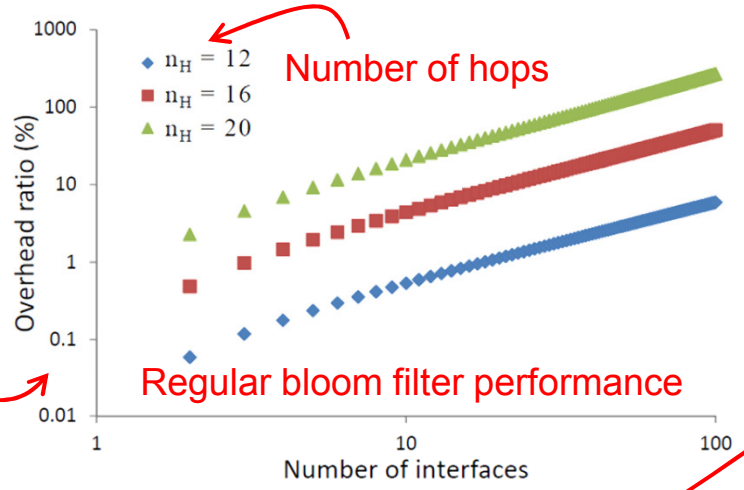
- Use an **in-packet filter**, which carries reverse path information
  - Optional hop-by-hop header, updated at each supported hop along the path

- Different alternatives for the in-packet filter
  - Bloom filter →Static field
    - **Constant size**, bits are set until received by content source
    - On reverse path, **no modification is possible**
    - Only requires **look-up and forward** operation
  - Counting bloom filter →Dynamic field
    - Consists of 2 hop-by-hop optional headers
      - a constant size Bloom filter component, a variable-sized counter field (encoded counter to reduce overhead)
    - On reverse path, **update is possible** (removing checked-entries)
    - Requires, **look-up, update, and forward** operation
  - Dynamic in-packet filter
    - Non-bloom filter based filter

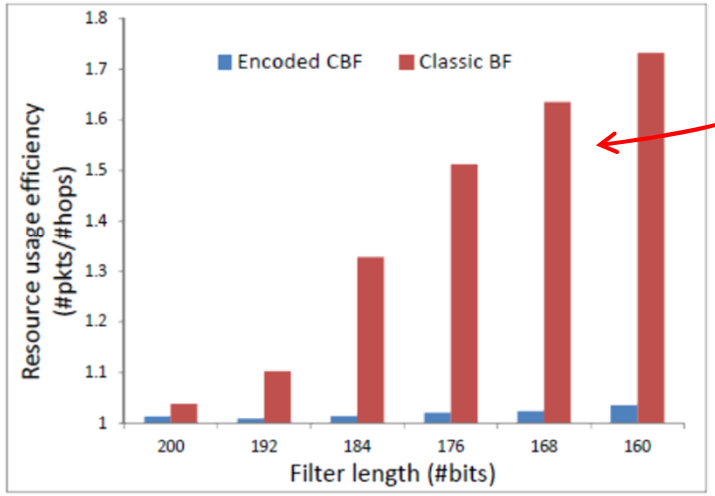


# Which Stateless Forwarding Approach?

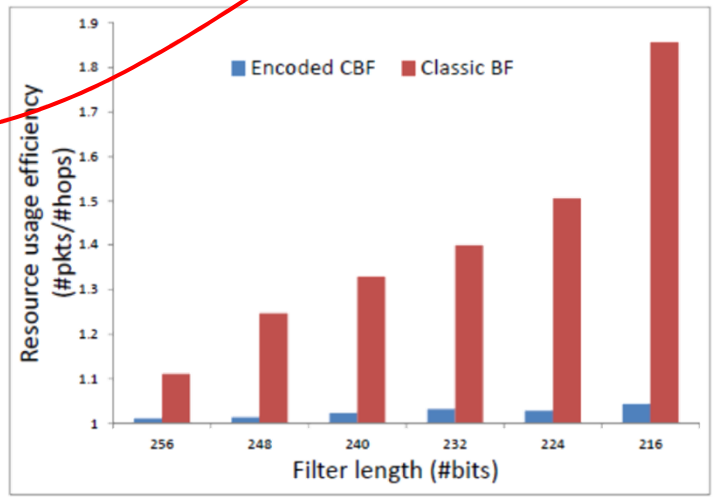
**Observation:**
**Bloom filter introduces significant overhead, triggered by false positives**




**Observation:**
**Proposed counting bloom filter (CBF) avoids false positives significantly and reduces the overall overhead**

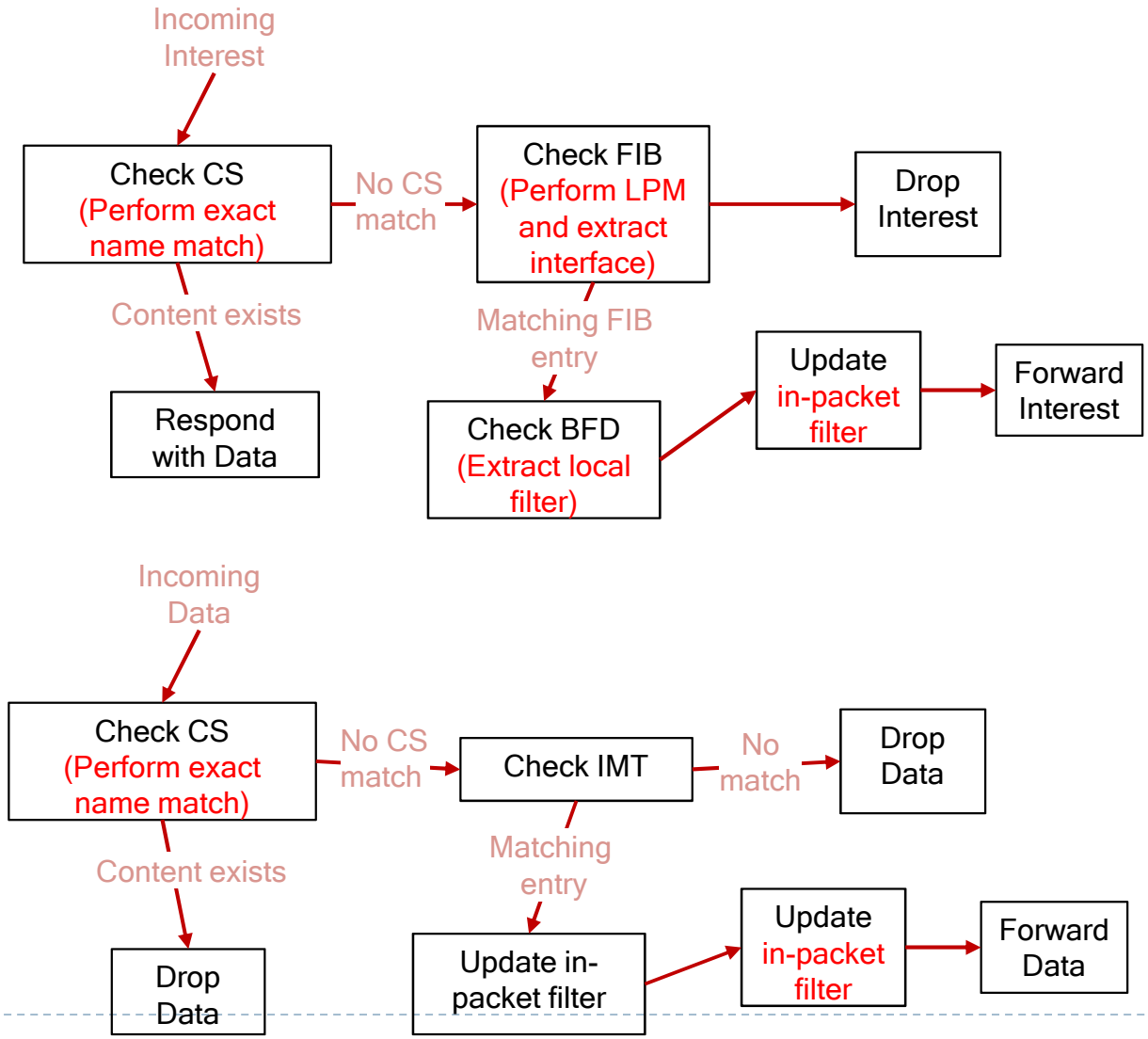



(a) $N = 1000$ nodes.



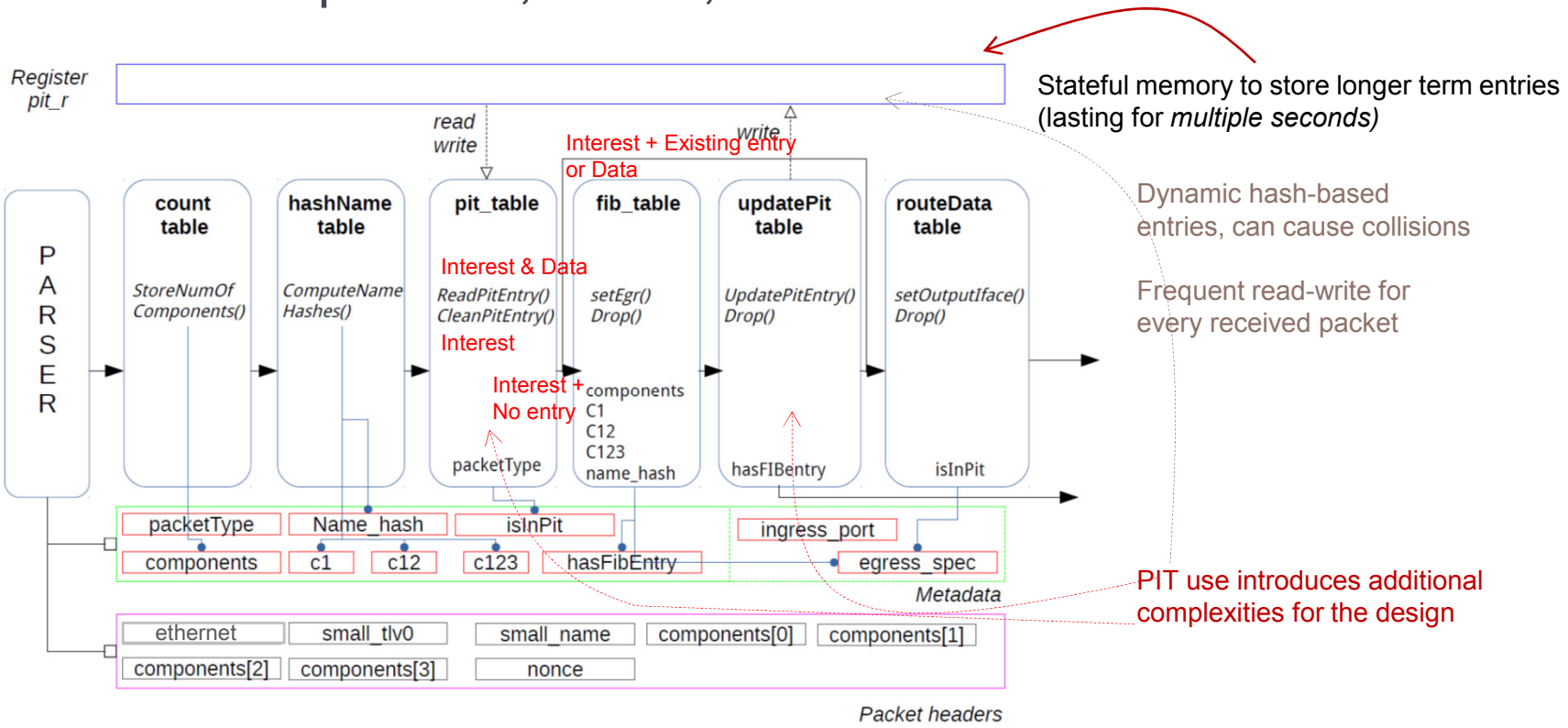

(b) $N = 3000$ nodes.

# Packet Flow in Stateless Forwarding



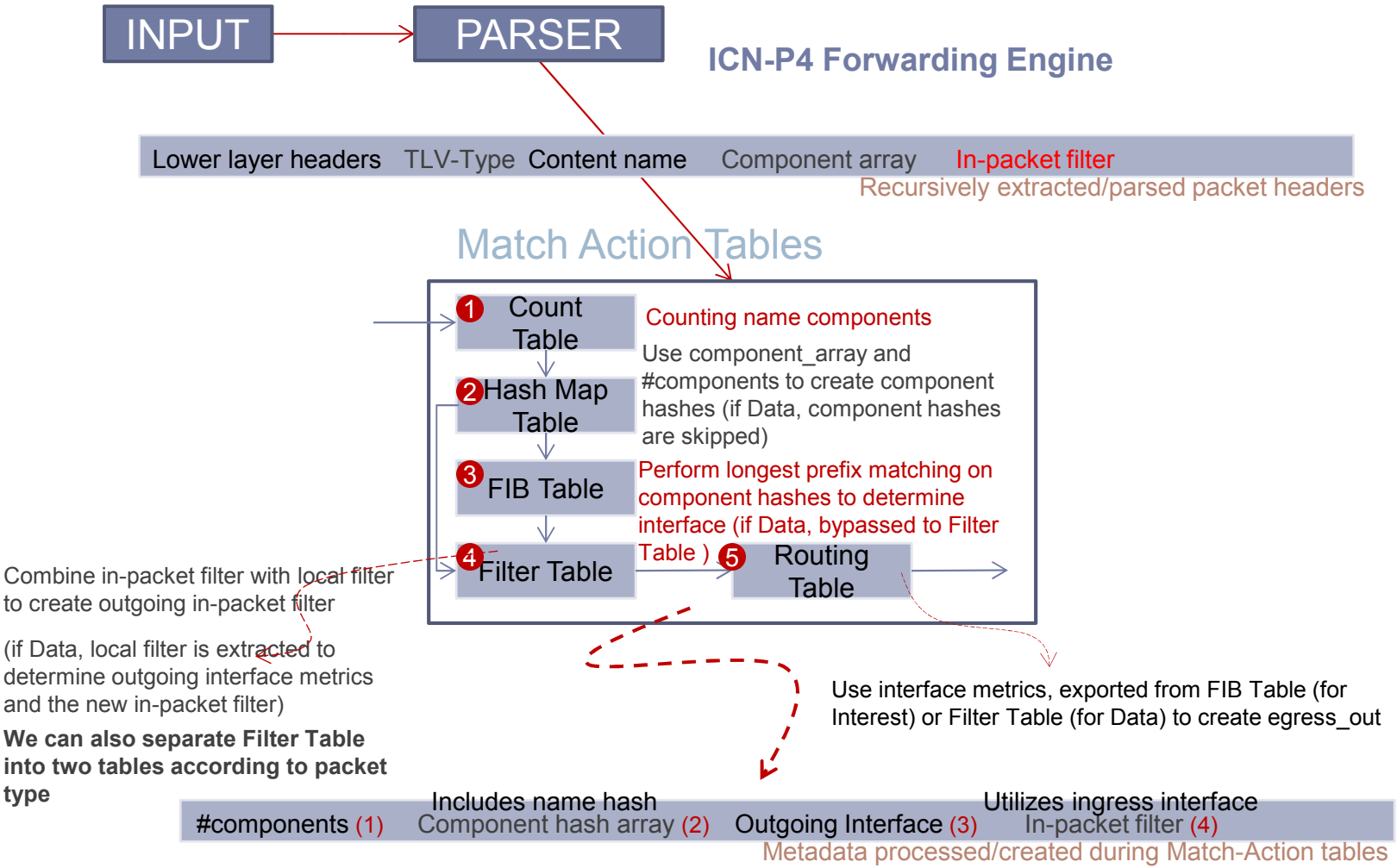
# Stateful ICN Forwarding with P4*

- Parse nested TLVs
  - Encoding dependent fields of packet type, content name, name components, nonce, etc.




* "NDN.p4: Programming Information-centric Data Planes", S. Signorella, *et al.*

# Stateless ICN Forwarding with P4



# Basic Metrics of Interest

- Storage/processing overheads
  - For stateful forwarding, PIT requirements
  - Processing overhead for each scenario
- Forwarding performance
  - Typical forwarding latency for received requests, depending on forwarding operation
- Combined analysis
  - Stateful and stateless traffic at different ratios, impact of one on the other, etc.

# Integrating to Netronome NFP

- Netronome's NFP (used on Agilio ISA) allows for more realistic implementations with better features
  - High parallelized processing capabilities, flexible storage options, and the integration of P4 and C
- As our main purpose is to demonstrate ICN capabilities with improved features, Agilio ISA offers a good design option for us
- We have other testing scenarios to get a better sense on the impact of ICN
  - Label based forwarding in ICN
    - Require a Forwarding Label Table (FLT) to use in conjunction with FIB
    - Additional variable sized packet headers to support the use of forwarding label
  - Flow-driven ICN forwarding
    - Require Flow Tables to store active flow information and to perform lookup
    - Additional packet headers to represent Flow Identifiers