IQ for DNA

Interactive Query for Dynamic Network Analytics

Haoyu Song

www.huawei.com

HUAWEI TECHNOLOGIES Co., Ltd.

Motivation

- **Service Provider’s pain point**
  - Lack of real-time and full visibility of networks, so the network monitoring and optimization capability is limited
- **Network visibility by data analytics is a SDN killer application**
  - Theoretically global view
  - First half of the full SDN control loop
  - For network planning, engineering, security, diagnosis
- **Network visibility is a big data problem**
  - Need standards for data collection, encapsulation, and presentation
  - Need to dig data plane potential for better data collection and preprocessing
  - Data source needs to cover the entire infrastructure

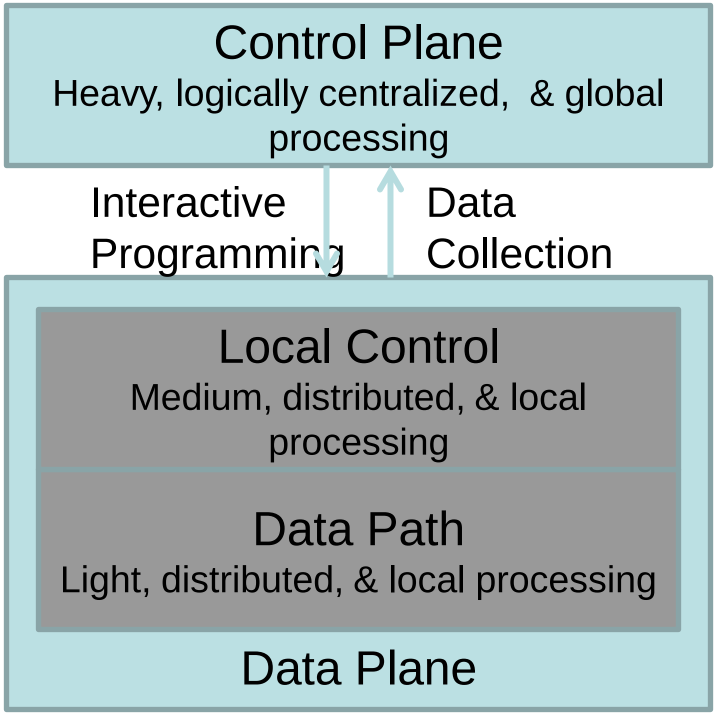
QoE	•Flow jitter, latency, drop rate measurement •Fault diagnosis •Application analysis
Security	•DDoS detection •Deep packet inspection •Stateful flow monitor
Customer Care	•Custom statistics •Flow tracing •Root cause analysis
Optimization	•Load estimation •Traffic matrix calculation •Elephant flow identification

Requirements for Network Analytics (1)

- **Network data analytics must be dynamic**
  - Why static methods doesn’t work
    - Difficult to predict all probe & measurement tasks in advance at design time
    - Pre-allocate resources for all potential data collection and processing tasks in data plane is prohibitively expensive
    - Data plane reconfiguration for new emerging tasks is too slow and can cause service interrupt
- **Therefore, Dynamic Network Analytics (DNA) is needed**
  - Incremental real-time and on-demand reconfiguration
    - Anytime, anywhere, & any action with dynamic resource allocation
    - Hitless in-service data plane modification
    - One data plane supports multiple parallel data analytical applications

Requirements for Network Analytics (2)

- **Decoupled network data analytics is inefficient**
  - Raw data drawn for data plane consumes control bandwidth and incurs long latency
  - No standard interface existing between the analytical application and SDN controller
  - Limited data extraction capability due to the inflexible data plane
- **Network analytics should rely more on in-network computing**
  - Close to the data source
  - Use processing capability of data path chip and local control processor
- **An integrated DNA system is needed**



Enabling Data Plane Technologies

- **Programmable data plane**
  - Allow customize the data plane forwarding application
  - Allow dynamically modifying the data plane behavior
  - Allow arbitrary actions on packets
- **Server-grade local control plane**
  - Enhanced CPU, memory, non-volatile storage and interconnection bandwidth with forwarding chips
  - Scale-out routers have dedicated server or server cluster as local control plane
  - Micro-service and VNF can be deployed in local processor
- **Affinitive, integrated, and efficient DNA implementation**
  - Combining the above two technologies, each network node can directly become a part of the big data analytical application software stack
  - Programming is the key to achieve this

What is the Gap

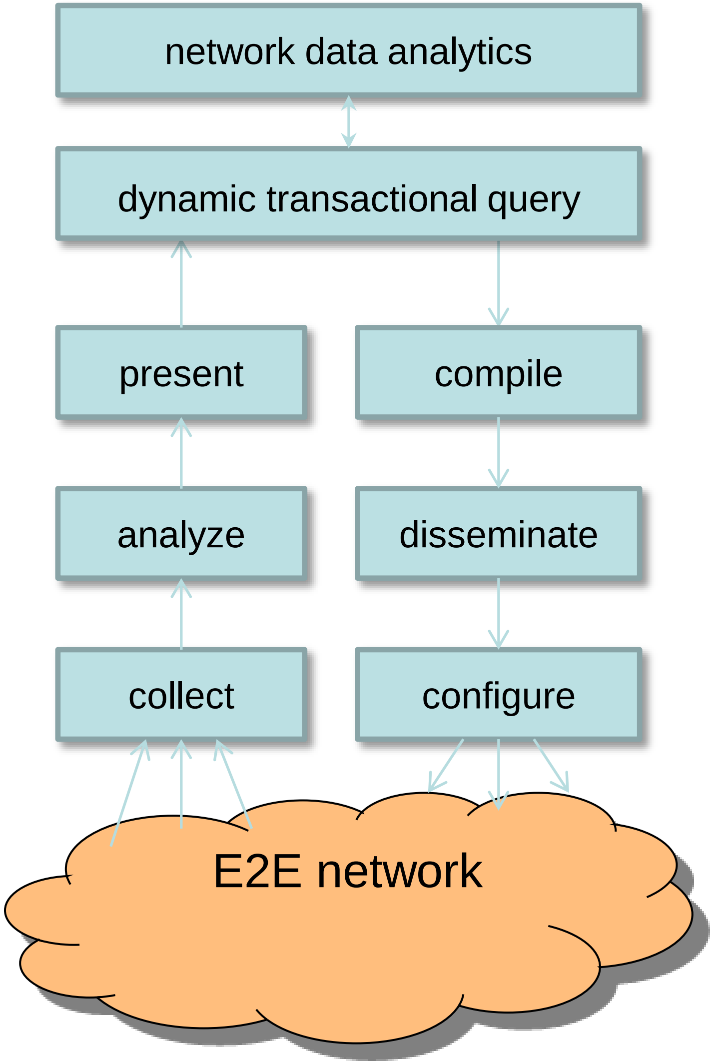
- **Programming model**
  - Common query API
  - Map Reduce
- **Programming language**
  - Interactive programming: real-time and on-demand
  - JIT compiler & common runtime interface
  - Programming abstraction
- **Target platform**
  - NP – fully programmable, but sensitive to modification
  - CPU – no distinction of data path and local control
  - ASIC – limited flexibility
- **Ecosystem**
  - Infrastructure scale visibility – E2E coverage
  - Virtual and physical platforms

Dynamic Network Probes

- **DNP is data probe deployed at designated locations in data path at runtime**
  - In-network stateful processing – control-data plane bandwidth efficiency
  - Dynamic resource allocation – data plane resource efficiency
- **DNP is essentially a finite state machine for data preprocessing**
  - Counter
  - Event trigger
  - Packet filter and sampler
- **DNP has many advantages**
  - Realtime deploy and revoking
  - Reduce bandwidth between data path and controller
  - Reduce overall latency of data analytical applications

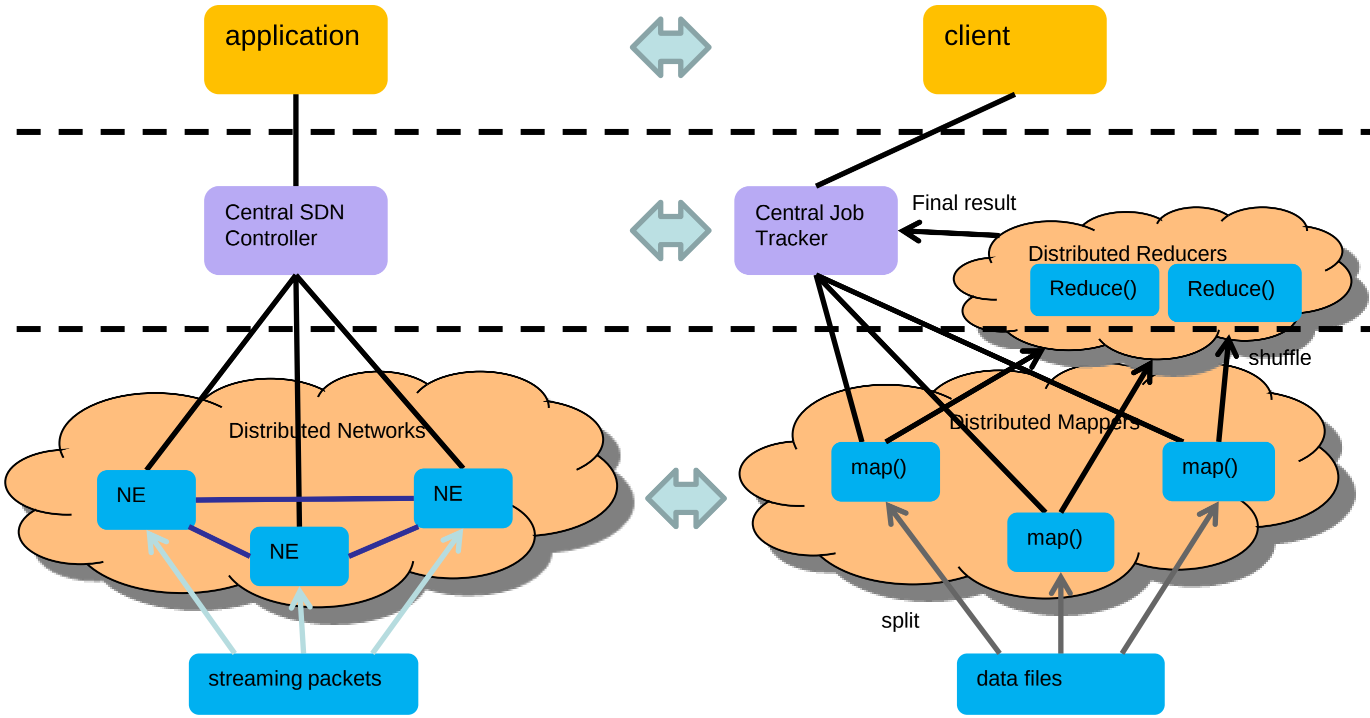
Programming Model – Standard Query API

- **API is used to define the data plane probing capability**
  - Isolate malicious attacks
  - Good for backend compiler
- **What's the right level of API abstraction?**
  - Application and data plane, which should be smarter and more knowledgeable?
  - SQL-like API is feasible
  - Any more possibilities?

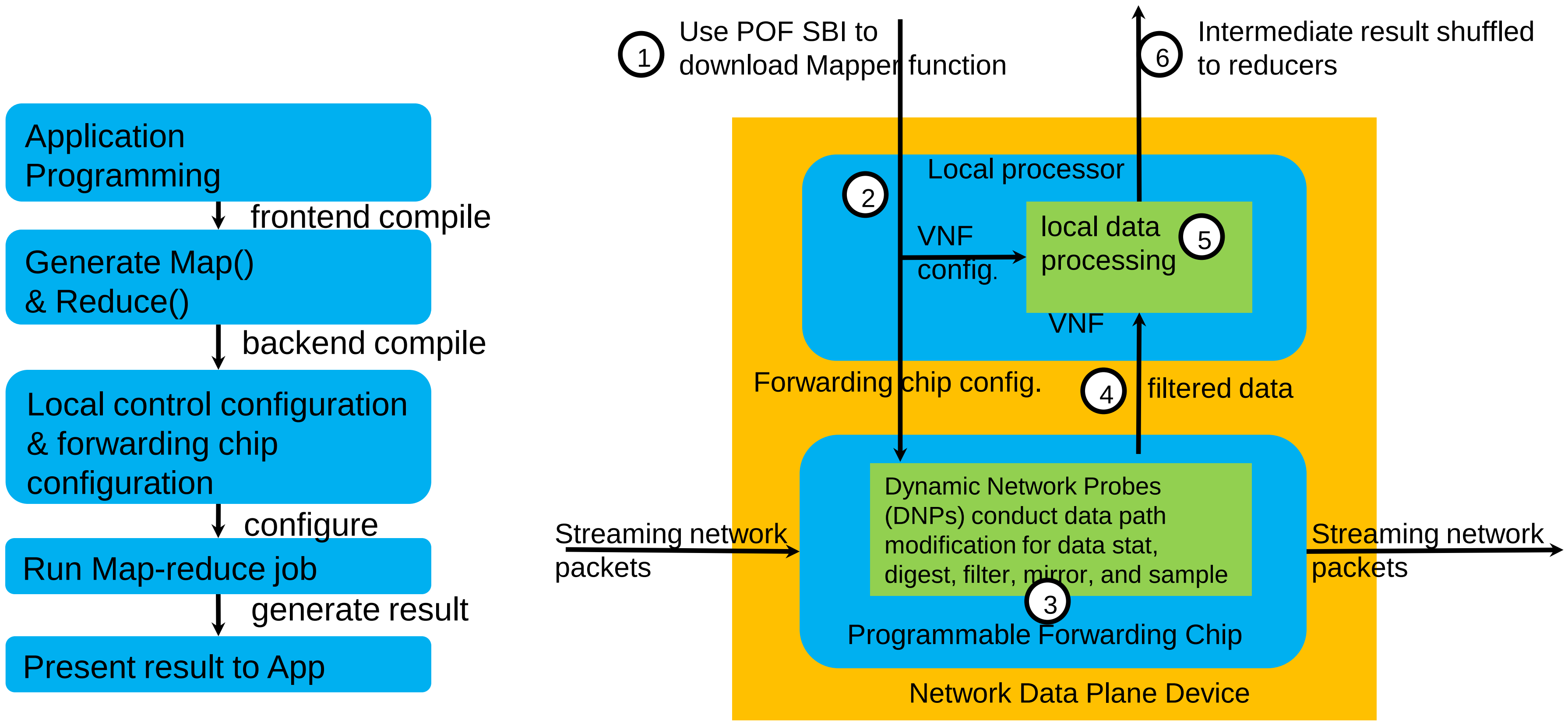


Programming Model – Network Map Reduce

- **Explore similarity between SDN architecture and MR programming model**
  - Data plane NE can serve as mappers and/or reducers
  - Controller can serve as job dispatcher and tracker



Network Map Reduce Architecture



- **map() is executed in both NE data path chip and local control**
- **reduce() is executed in NE local control or server at central controller**

Use Cases for NMR

- **DDoS Attack Detection**
  - Pick all portal switches as mappers and a few other switches as reducers
  - map()
    - Forwarding chip filters all unique flows which go to the target servers
    - Local processor calculate {k, v} pairs
  - reduce()
    - Calculate global {k, sum(v)} pairs and trigger alarm if threshold is passed
- **Traffic Matrix**
  - Pick all edge routers as mappers and a few other routers as reducers
  - map()
    - Forwarding chip labels each ingress packets with router id and keeps statistics for all egress packets from each edge router
    - Local processor read the counter periodically and push the {k v} pairs to reducers
  - reduce()
    - Summarize mapper inputs and generate the traffic matrix
- **Many other applications**
  - e.g., network congestion monitoring, elephant flow detection
  - As long as the application can be partitioned into two distributed functions

Research Challenges

- **Dynamic Network Probes**
  - Safety and Security
  - Consistency and Synchronization
  - Performance impact
  - Chip architecture
- **Interactive Programming/Query Language**
  - Parallel task orchestration
  - High level data analytical primitives
  - Streaming network system
  - Programming model and corresponding compiler

Prototype on Protocol Oblivious Forwarding

P4/C forwarding app

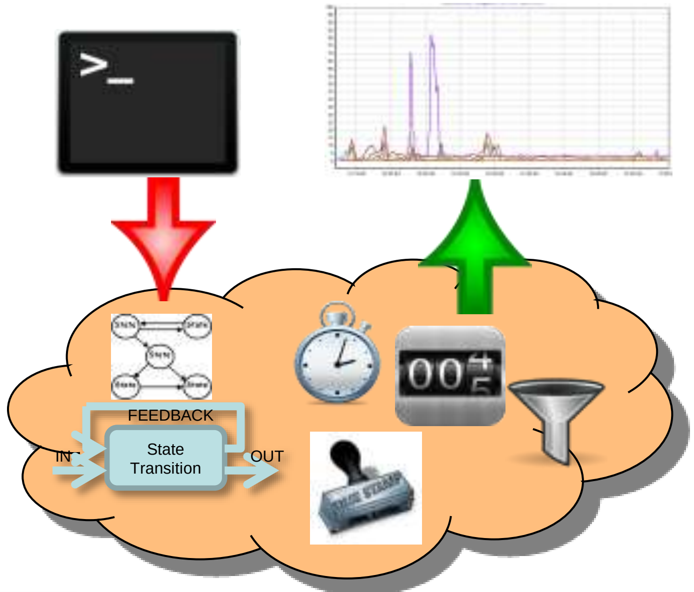
probe & measure app

POF IR

POF SBI

POF Device

Demo Use Cases		
DNA API		SDN Controller integration
DNP Compiler		
POF Interface		
vSwitch (OVS)	NE40E-based Prototype	



dynamically loaded table

match key	param.	pointer
dynamically loaded entry		
dynamically loaded entry		

dynamically loaded action in POF-FIS

dynamically loaded action in POF-FIS

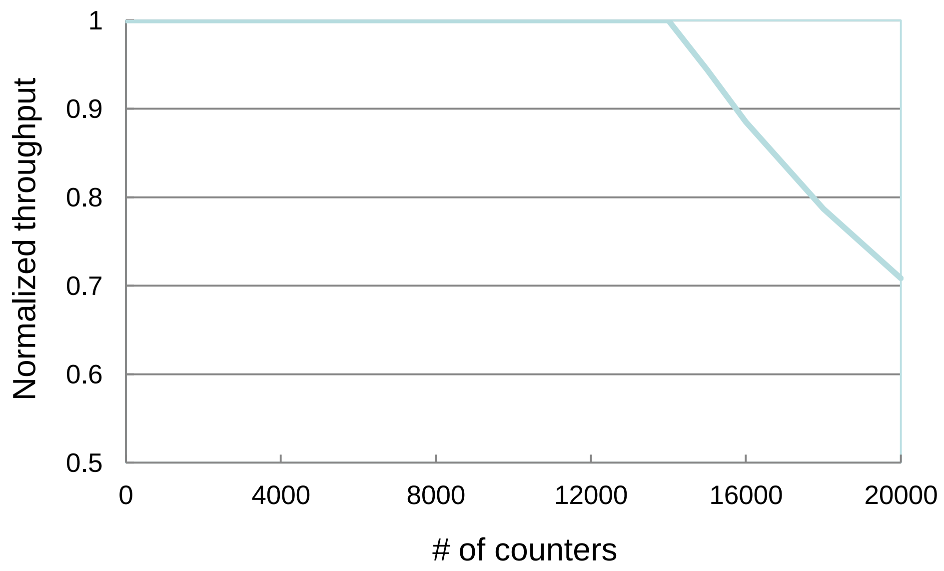
dynamically shared global resource (e.g., counter, register)

Prototyped on a router platform with 200G NPU-based line card

Performance Evaluations

	Compiling Latency	Configuration Latency	Total
Static Programming	1 s	1 s	2 s
DNP	0	50 ms	50 ms

~40 times latency gain when deploying a counter probe



DNP's performance impact when deploying counter probes

Related Works

- **In-band Network Telemetry (INT) & In-band OAM**
  - Static programming, not real time
- **Compiling Path Queries**
  - For Openflow forwarding model only
- **Stream Map Reduce**
  - Standalone system

Conclusion

- **Network analytics need runtime interactive data plane queries**
    - Dynamically programmable data plane is needed
    - In-network computing is needed
- **Multiple programming model exists**
    - Common APIs
    - Network Map Reduce
- **Dynamic Network Probe is a key element for DNA**
    - POF is ideal for real-time and on-demand DNP
    - Prototyped with high performance
- **Open research questions**
    - Data analytical abstractions and primitives
    - Interactive Programming and Query languages
    - Compiler technologies for distributed networks & heterogeneous targets

Thank you

www.huawei.com