Self-Programming Networks: Review + Next Steps (Smart NICs)

Balaji Prabhakar and Mendel Rosenblum
Departments of Electrical Engineering and Computer Science
Stanford University
Self-Programming Networks

A quest to make networks

*autonomous*: network should sense and monitor itself; program and control itself

*interactive*: network should be simple and fun to use, especially for 3rd party users

\[\text{SPN} = \text{Engineering} + \text{Design}\]

**Autonomy**
- Systems
- Programs
- ML/AI Algorithms

**Interactivity**
- Visual
- Transparent
- “Friendly and chatty”
Self-Driving Cars

Plain Old Volkswagen Tuareg

Sense + Control

Stanley
The Self-Driving Car
Self-Programming Network (SPN)

Plain Old Data Center

Sense, Infer, Learn and Control (SILC)

New Functionality
- Timestamping As A Service
- Fine-grained Network Telemetry
- App-Network Perf Monitoring

Workload

Operator Policies

Interactive Dashboard & Query Engine
Self-Programming Network (SPN)

New Functionality
- Timestamping As A Service
- Fine-grained Network Telemetry
- App-Network Perf Monitoring

Intuitive DB and QE
- Simple + visual + chatty
- App+network perf views

NIC-centric Architecture
- Sensing and control at NICs
  Smart NICs: big industry trend

Data and ML Intensive
- Use data and NNs to accelerate learning and for real-time processing
Milestones
Graduations

Yilong Geng: Dec 2018

Self-Programming Networks: Architecture and Algorithms

Zi Yin: Dec 2018

Understanding the Functionality, Dimensionality and Domain Adaptation of Word Embeddings: Towards a Better Chatbot for Networking Applications
Papers and Projects

Y. Geng, S. Liu, Z. Yin, A. Naik, B. Prabhakar and M. Rosenblum:
Y. Geng, S. Liu, Z. Yin, A. Naik, B. Prabhakar, M. Rosenblum and A. Vahdat:
“Exploiting a Natural Network Effect for Scalable, Accurate Clock Synchronization,” USENIX NSDI 2018
Z. Yin, V. Sachidananda and B. Prabhakar:
Y. Geng, S. Liu, Z. Yin, A. Naik, B. Prabhakar, M. Rosenblum and A. Vahdat:

Shiyu Liu: Clockchain
Feiran Wang: Multi-Raft, Distributed Tracing
Vig and Vin Sachidananda: Chatbots for Networking Applications
Sean Choi, Shiyu Liu and Muhammad Shahbaz: Smart NICs
Applications to FinTech
SAN FRANCISCO — Computer scientists at Stanford University and Google have created technology that can track time down to 100 billionths of a second. It could be just what Wall Street is looking for.

System engineers at Nasdaq, the New York-based stock exchange, recently began testing an algorithm and software that they hope can synchronize a giant network of computers with that nanosecond precision. They say they have built a prototype, and are in the process of deploying a bigger version.

For an exchange like Nasdaq, such refinement is essential to accurately order the millions of stock trades that are placed on their computer systems every second.

Ultimately, this is about money. With stock trading now dominated by computers that make buying and selling decisions and execute them with blazing speed, keeping that order also means protecting profits. So-called high-frequency trading firms place trades in a fraction of a second, sometimes in a bet that they can move faster than bigger competitors.

The pressure to manage these high-speed trades grows when the stock market becomes more volatile, as it has been in recent months, in part to prevent the fastest traders from taking unfair advantage of slower firms. High-frequency traders typically ac-

Continued on Page A17
Focus so far: Sense and Infer

Future focus: Learn and Control using Smart NICs
NIC-centric Architecture: Sense, Infer, Learn and Control

Applications

App 1
App 2
App 3

DC Interconnect Fabric

Sense, Infer
Learn, Control
Two-tier Control Architecture

**Deliberate Control**
- Acts on a *collection or group* of NICs
- To achieve a *common global objective*
- Involves computation or *decision-making with global knowledge*
- Generally applied over *longer timescales*
- *Networking analogy:* Management plane
- *Financial trading analogy:* Market data → trading strategy
  (performed in large compute clusters)

**Reflex Control**
- Applied at a single NIC
- To perform local tasks derived from/consistent with global objective
- Relies on local information at the given NIC
- Generally applied on every packet or round-trip time or ...
- *Networking analogy:* Data plane
- *Financial trading analogy:* trading strategy → buy/sell bids
  (performed in FPGAs in co-lo facilities)
Instantaneous, Local, Single NIC

Global, Group of NICs

Reflex Control

Deliberate Control

Time

Space

Instantaneous, Local

Episodic, Global

Reflex Control

Deliberate Control
First, use NICs to sense and infer network state

- Synchronize clocks on NICs (Huygens)
- Use synced clocks and edge-based measurements to infer network state (SIMON)
- **Note:** both Huygens and SIMON are amenable to the two-tier architecture
  - Implement them in Smart NICs as co-ordinated distributed systems

Then, devise and apply control; working on

- Controlling RDMA flows
- Congestion control with source routing
Remaining Talks

Shiyu Liu: Smart NICs I
- Deliberate control: Implementation of the Huygens algorithm using Mellanox and Broadcom NICs

Sean Choi: Smart NICs II
- Reflex control: Profiling serverless compute workloads with P4 on Netronome NICs

Shiyu Liu: SIMON with CPU time stamps vs NIC time stamps
- A comparative study of accuracy of network state reconstruction on CloudLab