

# Paper Summary

## HA 2.a

Damodar Sahasrabudhe

School of Computing, University of Utah

August 30, 2015

### 1 Title of paper: **NOX: Towards an Operating System for Networks**

Paper discussed in this summary is "NOX: Towards an Operating System for Networks"[1].

#### 1.1 First pass information

1. *Category:* Paper proposes a new research idea about "Network Operating System" - a platform for network developers. Paper also provides evidence gathered from experiments and example codes.
2. *Context:* Problem addressed in paper is ease of network management.
  - It builds more on OpenFlow[2] and proposes more sophisticated model - a Controller(s) which not only control(s) the switch but the whole enterprise network.
3. *Assumptions:*
  - Network will not be attacked and hence security features are not added as of now, but kept for future work.
  - Application developers using NOX will be aware of will build fault free applications.

4. *Contributions:* Ease of network management and network application development / research.

#### 1.2 Second pass information

*Summary:*

- Introduction: Paper introduces how Operating systems encapsulate hardware / architecture details from application programmers making it easier to code, debug and port applications. OS takes care of IPC, memory management, program execution etc. which makes it easier for user to manage the machine. Paper suggests necessity of similar common platform for networks as well.
- Overview:
  - Components - NOX consists of OpenFlow switches, a database housing network view, a server which runs NOX controller. NOX controller hides all the low level hardware implementation and provides abstract interface for applications used to manage network.
  - Operation - Operation is similar to OpenFlow - Controller reads first packet from new flow and forwards to appropriate destination. Rest of the packets follow the same direction.

- Scalability - Packet forwarding is handled by switches at line speed. Flow initiations are handled by Controller - which can handle 100,000 flows per second is seems sufficient for enterprise wide network. The global database of network view can also be maintained as "changes in network" are not as frequent as packets. As a result NOX seems scalable enough to handle dynamic changes in network or the routing rules.
- Security measures are must as NUX, when implemented is going to be heart of the network.
- Fault tolerance is needed.
- *Future work:* Development of new equipments or protocols should be accompanied by "device drivers" to interface them
- Programming Interface - This section lists typical "Events" (similar to interrupts used by microprocessor), Network view (which maintains mapping between host names and real address) and higher level services offered to developers.
- Examples - The best example LOC required to build Ethane - with and without NOX. Using higher level abstraction provided by NOX, LOC for Ethane[3] reduced by around 80

## References

- [1] NOX: Towards an Operating System for Networks
- [2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: enabling innovation in campus networks. SIGCOMM Comput. Commun. Rev., 38(2):6974, 2008.
- [3] Martin Casado, Michael J. Freedman, Justin Pettit, Jianying Luo, Nick McKeown, Scott Shenker. "Ethane: Taking Control of the Enterprise"

### 1.3 Third pass information

- *Strengths:*
  - Clarity of the paper
  - Idea to provide OS is good. It possibly can do same thing in networks what Android did in smart phones.
  - Evidence provided using working examples.
- *Weaknesses:*
  - Against principle of Internet i.e. distributed control.
  - Lack of security at the heart of the network.
- *Questions:* Need more details about implementations and APIs if one intends to use NOX.
- *Interesting citations:* Z. Cai, F. Dinu, J. Zheng, A. L. Cox, and T. S. E. Ng. Maestro: A Clean-Slate System for Orchestrating
- *Possible improvements:*