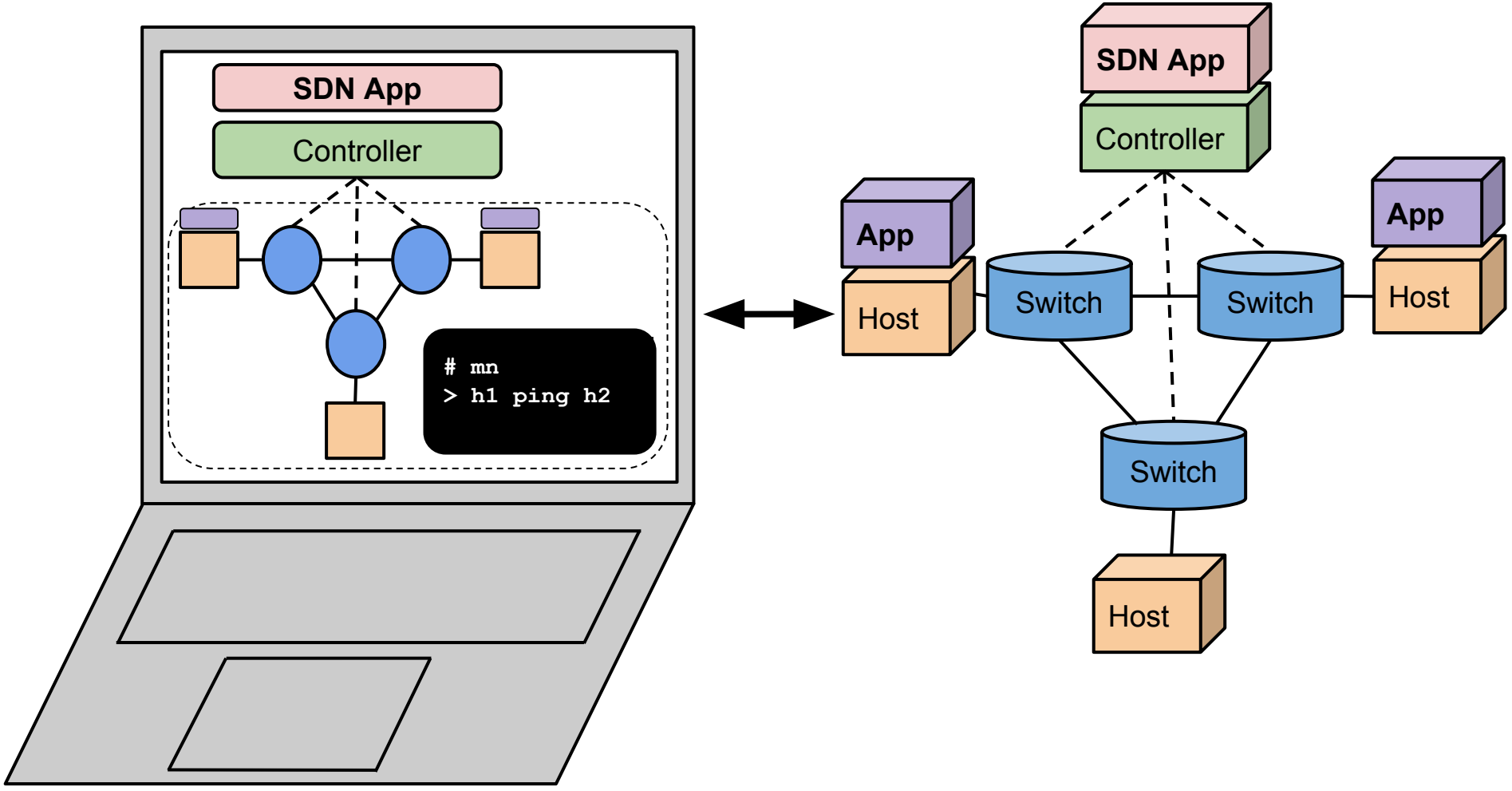
Mininet and the Importance of Software Research Projects

Bob Lantz
Mininet Project (founder)
Open Networking Laboratory

Outline

- Why Mininet?
- Why Software Research Projects matter
- How we can improve SOSR, SIGCOMM and systems research in general!

What is Mininet?



Emulated Network

Hardware Network

Why Mininet?

"Networks are hardware without an OS"

But to develop, you need a *platform*.

Yiannis's jet engine

Belief: "High-impact" project

Mininet: Impact of a Software Research Project

Research: 40+ papers, 1000+ citations* (google scholar, ymmv)
(Hopefully some of you have found it to be useful also!)

Replication: 107+ replicated experiments
(Thanks to Nick McKeown et al. and students in multiple years of Stanford CS244! reproducingnetworkresearch.wordpress.com)

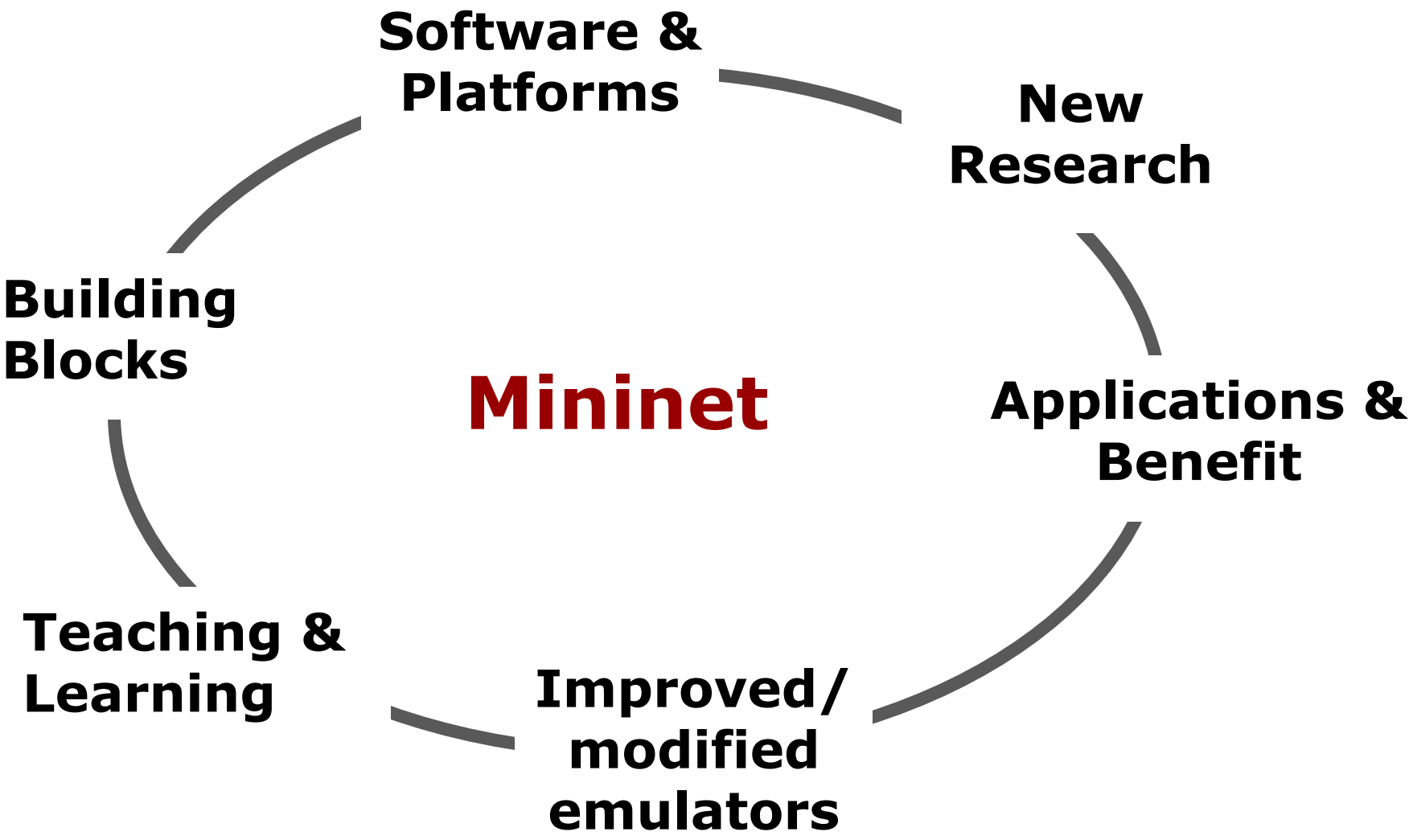
Teaching: 50,000+ students in MOOCs; courses, labs, self-teaching

Industry: Nearly all open source and commercial SDN solutions leverage Mininet somehow

Community: 450K downloads, 41+ contributors, 2000 people on mailing list

Most important impact isn't quantitative, but **qualitative**!

Projects like Mininet help to create Positive feedback loops



Why Software Research Projects Matter

Software Research Projects are research projects that largely consist of building a usable software system (tool, platform, app…), usually over several years, to solve an important unsolved problem or set of problems faced by the field.

So why are they particularly significant or beneficial?

Why Software Research Projects Matter

Software Research Projects are research projects that largely consist of building a usable software system (tool, platform, app…), usually over several years, to solve an important unsolved problem or set of problems faced by the field.

So why are they particularly significant or beneficial?

- **Direct benefit to researchers and practitioners**
- **Believable, reproducible**
- **Generate/help solve additional research problems**
- **In-depth knowledge and insight**
- **Facilitate long-form scholarship**
- **Open source benefits/leverage**
- ***Multiplicative* impact and positive feedback loops**

(dis)Incentives *against* Software Research Projects

(Over)emphasis on *novelty*
- difficulty of publishing multiple papers on a single system
- "new" systems valued vs. extending existing systems
- sometimes resulting in "name confusion"

Difficult, long, hard work
- Interferes with publishing, working on other projects
- Opportunity cost
- Possibility of real or apparent stagnation/burnout

Minimal personal reward/benefit
- Many positive externalities, few direct rewards
- Fewer bullet points on your (research/academic) C.V.
- Hard(er) to get ongoing funding

How we can improve SOSR, SIGCOMM, and Systems Research in General

How we can improve SOSR, SIGCOMM, and Systems Research in General

1. **Encourage Software (and systems) Research Projects**
   - Expanded demo sessions at conferences (with associated mini-papers ala SIGCOMM)
   - More workshops/conferences involving real, usable systems
   - Awards and recognition (such as **this SOSR Systems Award!)**
   - Encourage and recognize collaboration and extension of existing systems
   - Align Software Research Projects with goals of students and researchers

How we can improve SOSR, SIGCOMM, and Systems Research in General

1. Encourage Software (and systems) Research Projects

- Expanded demo sessions at conferences (with associated mini-papers ala SIGCOMM)
- More workshops/conferences involving real, usable systems
- Awards and recognition (such as **this SOSR Systems Award!)**
- Encourage and recognize collaboration and extension of existing systems
- Align Software Research Projects with goals of students and researchers

2. Reward Follow-through and long-term scholarship

- Revitalize conference paper -> long form (article, book, etc.) pipeline
  - also: updates/revisions/corrections
- Synthesis, Experiences, Retrospectives
- Validation and Replication

How we can improve SOSR, SIGCOMM, and Systems Research in General

1. Encourage Software (and systems) Research Projects

- Expanded demo sessions at conferences (with associated mini-papers ala SIGCOMM)
- More workshops/conferences involving real, usable systems
- Awards and recognition (such as **this SOSR Systems Award!)**
- Encourage and recognize collaboration and extension of existing systems
- Align Software Research Projects with goals of students and researchers

2. Reward Follow-through and long-term scholarship

- Revitalize conference paper -> long form (article, book, etc.) pipeline
  - also: updates/revisions/corrections
- Synthesis, Experiences, Retrospectives
- Validation and Replication

3. Enable/Incentivize *Reproducible Research*

- ACM/IEEE/... should host permanent archives of all digital artifacts, not just PDFs
- Goal: ***Runnable papers*** - point/click/replicate!
- Badging and rewards for reproducible/reproduced experiments
- Plug: join us for Reproducibility '17 at SIGCOMM

7 "crazy" research questions your advisor doesn't want you to know about - could Mininet and SDN help to answer them?

Emulators - (How) can we trust them?

How can system software support fast, accurate network emulation?

(How) can we prevent SDN from devolving into "configuration automation" and "NFV plumbing?" Make SDN great again?!

Network Operating Systems - what are they good for?

Do we really need VNFs? ("Network Function Elimination?")

Is SDN (and Networking in general) Harmful or Irrelevant in the age of RDMA?

(How) can we avert the (privacy/security/IoT) apocalypse?

Acknowledgments

Mininet would not exist or have had such a positive impact without the outstanding assistance of everyone who has contributed to the project, including:

Original Mininet Contributors: Brandon Heller, Nikhil Handigol, Vimal Jeyakumar, Brian O'Connor

Arista colleague and creator of network namespaces: Eric Biederman; Linux, software switch developers

Stanford/OpenFlow Colleagues

Instructors and Students of Stanford CS244 and CS144 (notably Nick McKeown)

Open Networking Laboratory

The Mininet Community: users, code contributors, students, teachers, researchers, industry practitioners - thank you!!

The SDN Community: including the organizers of this conference and all of you!

Summary

Why do I (we) do this?

Interest/Fun/Learning, Useful, Positive Impact, Enabling the Success of Others!

Software (and hardware) Research Projects Matter

We should incentivize their construction and enable more researchers/students/practitioners to create them!

How we can improve SOSR, SIGCOMM, and Systems Research in General

1. Encourage Software (and systems) Research Projects

- Expanded demo sessions at conferences (with associated mini-papers ala SIGCOMM)
- More workshops/conferences involving real, usable systems
- Awards and recognition (such as **this SOSR Systems Award!)**
- Encourage and recognize collaboration and extension of existing systems
- Align Software Research Projects with goals of students and researchers

2. Reward Follow-through and long-term scholarship

- Revitalize conference paper -> long form (article, book, etc.) pipeline
  - also: updates/revisions/corrections
- Synthesis, Experiences, Retrospectives
- Validation and Replication

3. Enable/Incentivize *Reproducible Research*

- ACM/IEEE/... should host permanent archives of all digital artifacts, not just PDFs
- Goal: ***Runnable papers*** - point/click/replicate!
- Badging and rewards for reproducible/reproduced experiments
- Plug: join us for Reproducibility '17 at SIGCOMM