PacketLab:
A Universal Network Measurement Platform

Kirill Levchenko *with*
Amogh Dhamdhere, Bradley Huffaker, Nicholas Weaver, Vern Paxson

Edge Measurement

- Active measurement from end hosts where vantage point is an experimental factor
  - Censorship and traffic tampering (e.g. Netalyzr)
  - Consumer bandwidth/latency (e.g. SamKnows)
  - Network topology (e.g. Skitter/Ark)

Traditional Model

1. **Design** experiment
2. **Build** endpoint agent and control infrastructure
3. **Deploy** endpoint agents
4. **Measure** and publish

costly, increases barrier to entry

Ideal Model

1. **Design** experiment
2. **Reuse** existing measurement infrastructure
3. **Measure** and publish results

Stakeholders

- **Experimenter** wants to do a measurement experiment
- **Operator** operates measurement endpoints
  - Experimenter and operator are the same in most experiments
- **Access provider** gives operator access to Internet
  - Party legally associated with the Internet connection
  - Trusts operator to honor use restrictions
  - May be non-technical

Obstacles to Reuse

- **Experimenter** needs to port experiment to each platform
- **Operator** (sharer) needs to support outside experiment
  - Incurs cost in resources and effort
  - Must trust the experimenter
- Operator may no longer want to operate infrastructure, endpoints may be abandoned but still useable

PacketLab Goals

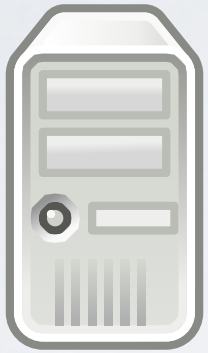
- **Research question:** Can we remove the technical obstacles to sharing measurement endpoints?
  - Reduce experimenter effort
  - Reduce operator effort (or shift cost to experimenter)
  - Give operator control over what experimenter can do

PacketLab Approach

- Universal endpoint that provides an *interface to network*
  - A VPN endpoint with measurement support
  - *Not* an interface to *host* like PlanetLab
- No permanent control infrastructure
  - Endpoint–experimenter rendezvous mechanism
- Fine grained control of what experiment can do

Traditional Model

Control Server



Control logic

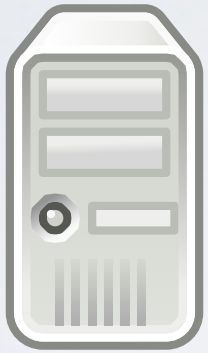
Endpoint

Experiment logic

Network interface

PacketLab Model

Control Server



Control logic

Experiment logic

Endpoint

Network interface

PacketLab Endpoint

- Provides access to network
  - Software agent (like Netalyzr), hardware agent (like SamKnows), dedicated server (like Ark)
- Very simple API
  - TCP/UDP socket open, send, receive
  - Raw IP send and receive
- *Don’t need to update for new experiment*

PacketLab Endpoint

- **Research question:** Can the PacketLab endpoint primitives support a rich set of Internet measurements?
  - Port existing measurement experiment to PacketLab
  - Identify what information endpoints need to export
- **Research question:** Is it possible to have maintenance-free measurement endpoints?

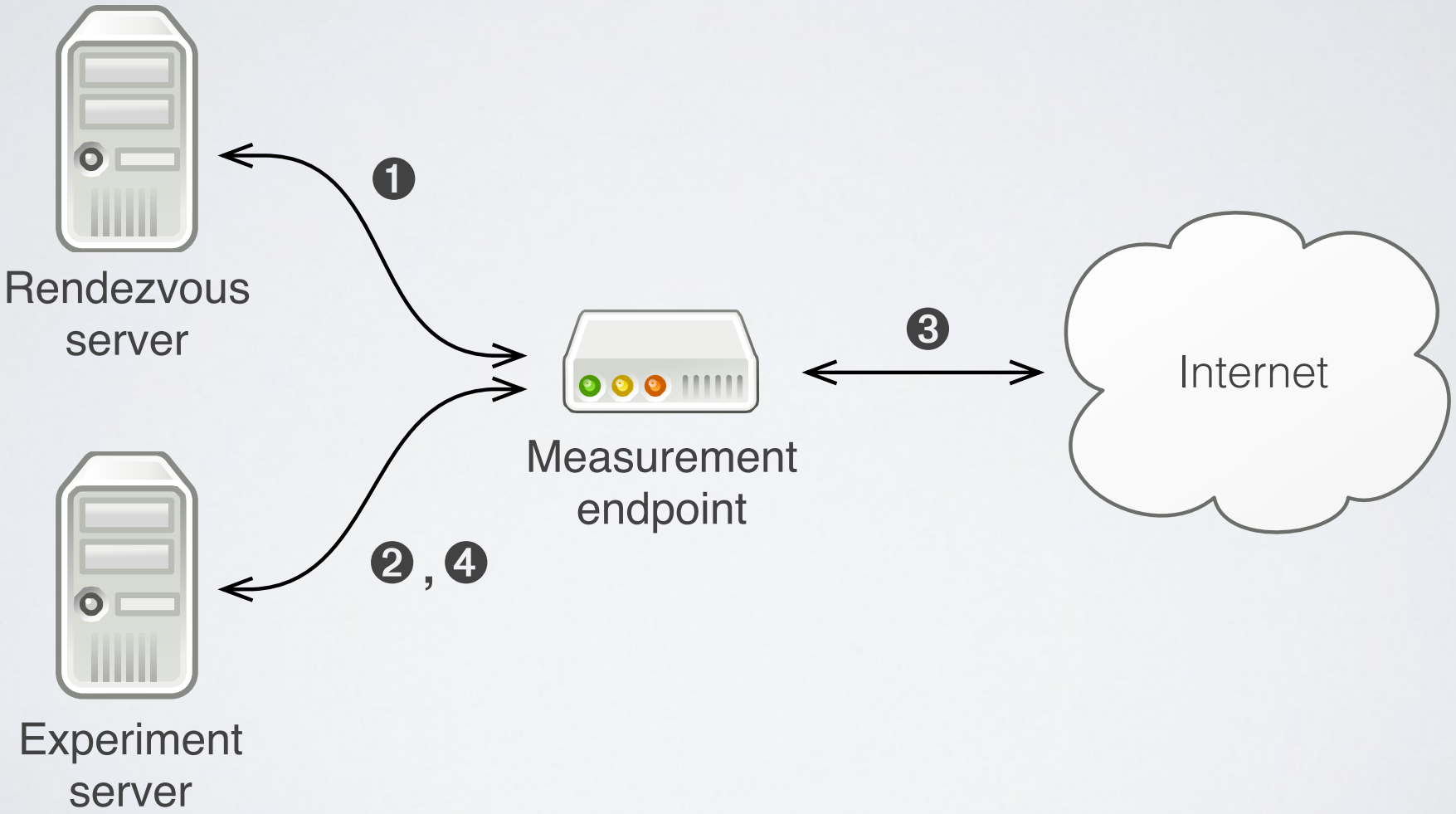
Experiment Control Server

- **Ephemeral:** Exist for duration of experiment only
- Run by experimenter, *not* endpoint operator
  - Shifts cost from operator to experimenter
  - Essential when operator no longer operates the infrastructure
- **Research question:** How does moving the experiment logic from endpoint to experiment server impact experiment design?

Rendezvous

- **Need a way to connect endpoints to experiments**
- Rendezvous server: Directory of experiments
- Experimenters *publish* experiments to rendezvous server
- Endpoints *subscribe* to experiments
- Handful of community-operated servers
  - Like NTP, DNS, or PGP servers

Rendezvous


Rendezvous server
1
Measurement endpoint
3
Internet
2, 4
Experiment server

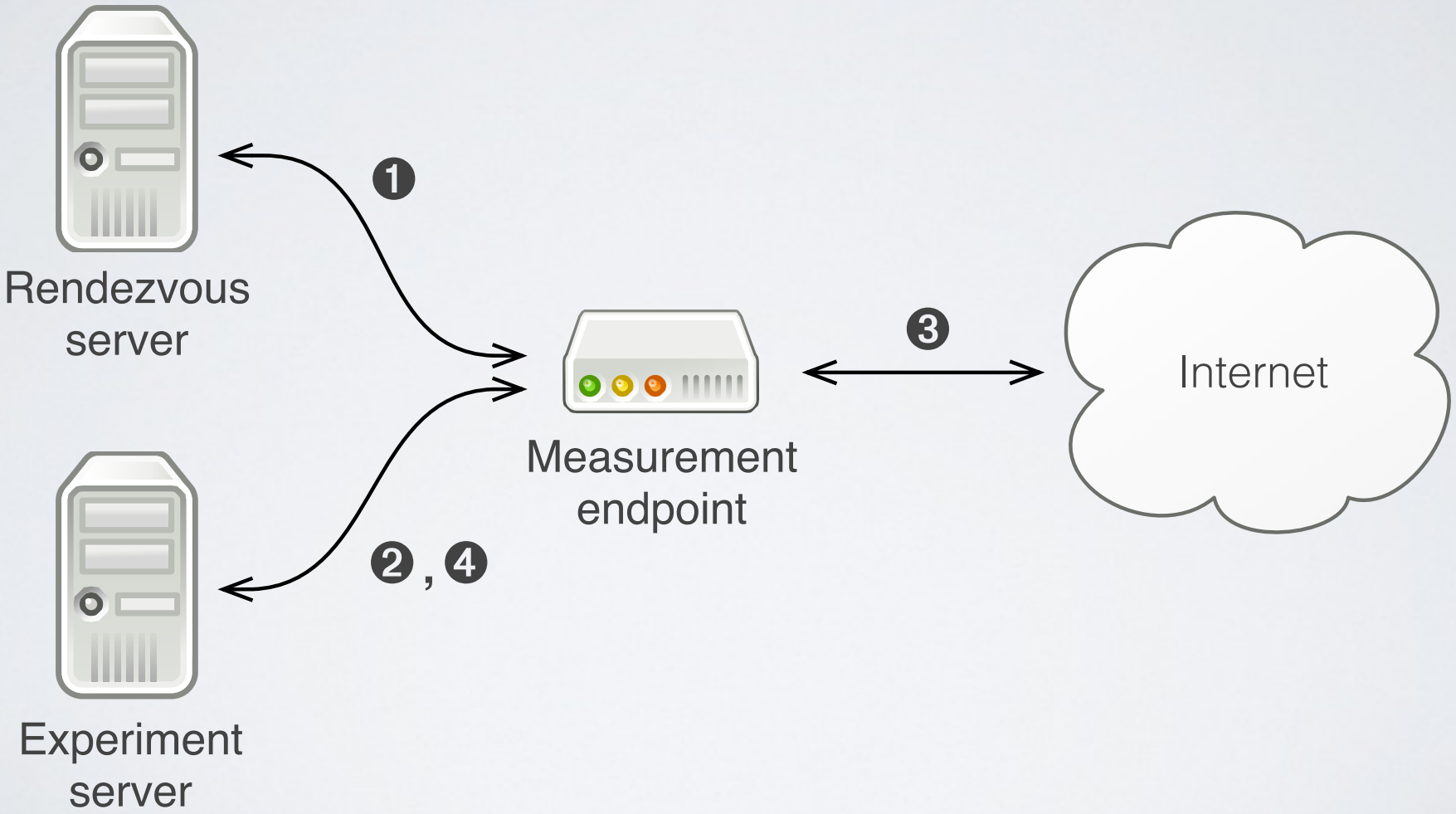
Rendezvous

- Endpoints contact experiment servers directly
  - Operator not involved in experiment discovery or execution
- **Research question:** Can endpoints be deployed without permanent control infrastructure?
- **Research question:** Can endpoints continue to function after their primary operator stops maintaining them?

Access Control

- Operators give experimenters digital certificates granting access to their endpoints
- Each endpoint has its own root of trust
  - Only agrees to do experiment signed by a trusted key
  - Operators install their key when they deploy endpoint
- Experiment server provides certificate to each endpoint
- Certificates can be chained

Rendezvous


Rendezvous server
1
Measurement endpoint
3
Internet
2, 4
Experiment server

Control of Experiments

- Operator wants to restrict the kinds of experiments and experimenter can run on endpoints
  - Today based on trust relationships
- Operator specifies *packet filters* that restrict the kinds of packets experimenter can send during experiment
  - Can use BPF; working on a richer mechanism
- Filters attached to experiment certificates
  - Presented to endpoint with certificate
  - No direct communication between operator and endpoint

Control of Experiments

- **Research question:** Is the experiment filtering mechanism sufficiently expressive?
- **Research question:** Does the ability to restrict experiments encourage endpoint sharing?

Workflow

1. Experimenter designs experiment
2. Experimenter obtains an experiment certificate from endpoint operators authorizing experiment
    1. Certificate includes experiment filter limiting what kind of packets can be sent and collected during experiment
3. Experimenter deploys experiment server
4. Experimenter publishes experiment certificate to rendezvous server

Workflow

5. Endpoint connects to rendezvous server and subscribed to all experiments signed by its trusted root keys
6. Rendezvous server tells endpoint about experiment server that wants to do experiment signed by its root
7. Endpoint connects to experiment server
8. Experiment server presents full experiment certificate (including experiment filter)

Workflow

9. Endpoint checks certificate, experiment begins
10. Experiment server tells endpoint what packets to send where and what packets to capture
11. Endpoint only sends/receives packets allowed by filter
12. When experiment completes, experiment server disconnects

Status

- Implemented basic endpoint
  - No rendezvous, just contacts specified experiment server
  - No experiment certificates
  - No filtering
- Implemented some simple experiments
  - Ping, UDP bandwidth

Conclusion

- **Goal: Remove technical obstacles to sharing endpoint measurement infrastructure**
  - Reduce experimenter effort
  - Reduce operator effort (or shift cost to experimenter)
  - Give operator control over what experimenter can do
- *We want your feedback!*

Endpoint Interface

- **nopen:** open socket (raw, TCP, UDP)
- **nsend:** schedule packet/data to be sent at some time
- **npoll:** retrieve received packets (with timestamps)
- **ncap:** specify packet capture filter (in raw mode)
- **mread/mwrite:** read endpoint information, including current time (for synchronization)

Endpoint Interface

- **Packets are scheduled to be sent** (may, but need not, be sent immediately)
- Packets received by endpoint buffered until requested by experiment server using npoll command
- Gives experiment server control over access link utilization; useful for bandwidth/latency measurements

Filtering

- Filters are programs executing in specialized virtual machine (like BPF) on the endpoint
- Filter program as access to configuration information about endpoint and packet data
- Called for each sent/received packet for allow/deny decision
- Written in C-like language (in development)