Mosh
An Interactive Remote Shell for Mobile Clients

Keith Winstein (with Hari Balakrishnan)

keithw@mit.edu

May 31, 2012

Secure Shell, 1995

- Connects local terminal to remote terminal.
- Conveys over TCP:
  - user keystrokes $\rightarrow$ server
  - octet stream (coded screen updates) $\rightarrow$ client terminal
- Connection endpoints dictated by IP:port on both sides

Post-1995 problems with SSH

- Can't roam:
  - ...across Wi-Fi networks.
  - ...from Wi-Fi to cell or vice versa.
- TCP times out if data unacknowledged after n minutes.
  - Session dies if laptop goes to sleep.
- TCP responds poorly to packet loss.

More problems with SSH

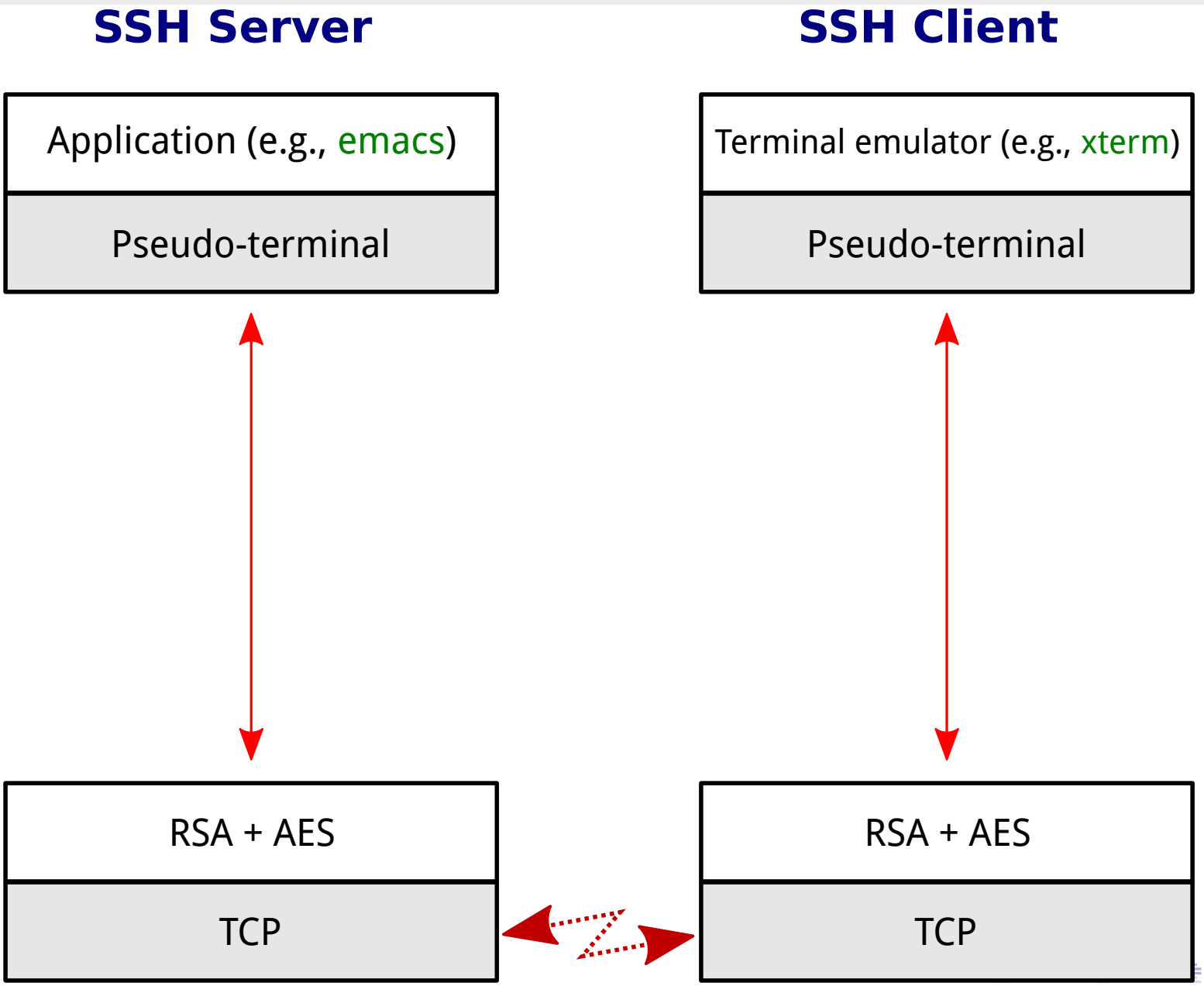
- Byte stream is wrong layer of abstraction for screen.
  - Want client at *latest* screen.
  - SSH doesn't understand data, so must send everything.
- Typing and editing on high-latency path is frustrating.

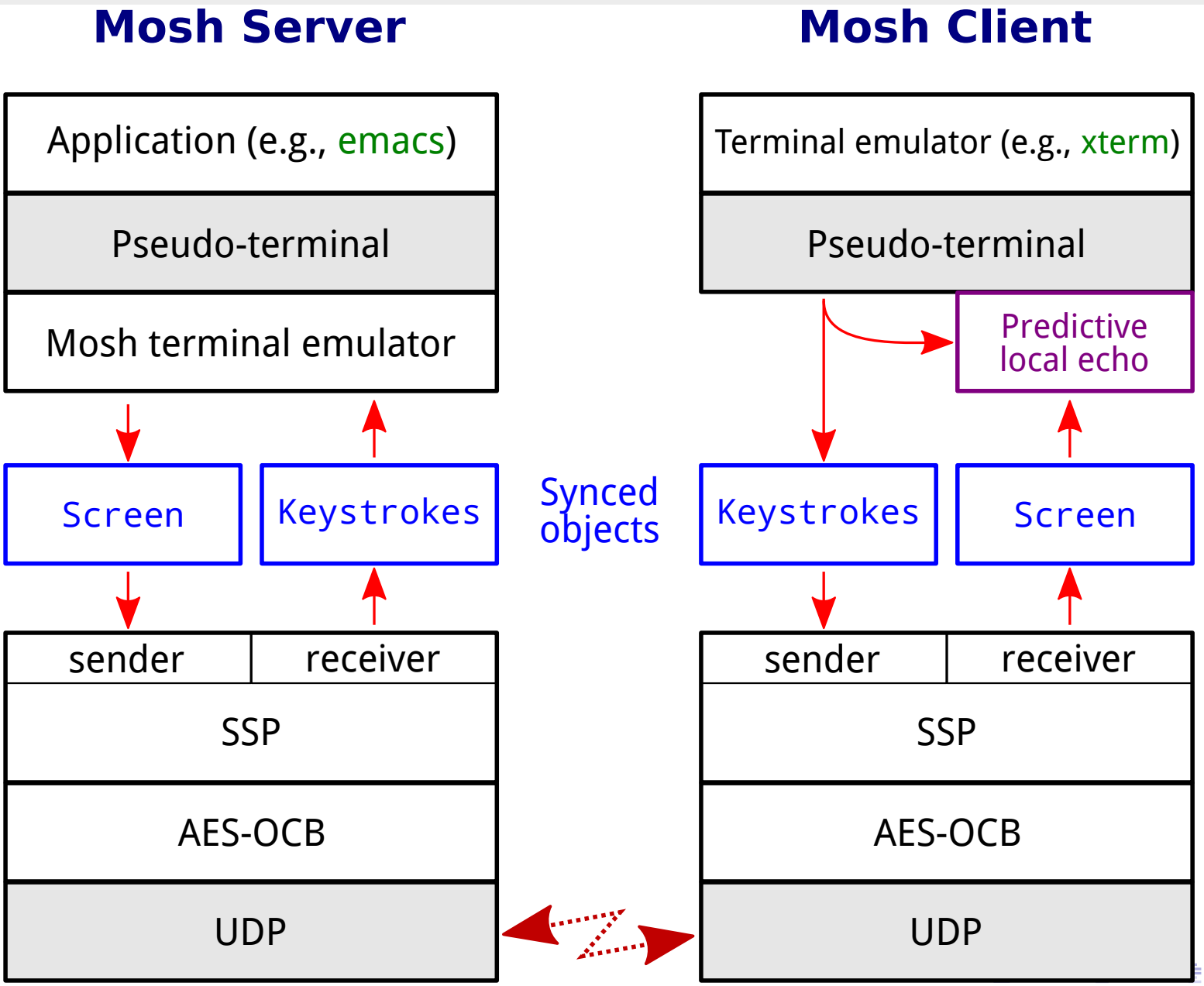
Solution 1: State Synchronization Protocol

- Runs over UDP.
- Instead of synchronizing *octet streams*, synchronize *objects*.
- Object represents state of endpoint.
- Implements simple interface:
  - `diff`: make vector from state $A \rightarrow B$
  - `patch`: apply vector to A, producing B

State Synchronization Protocol (cont.)

- Protected by AES-OCB (Krovetz 2011)
- Key exchange happens out of band.
  - Uses SSH to bootstrap.
  - No privileged code, no daemons.
- Roaming is easy:
  - Source address of latest authentic datagram from client $\Rightarrow$ new destination address for server.


SSH Server
SSH Client
Application (e.g., emacs)
Pseudo-terminal
Terminal emulator (e.g., xterm)
Pseudo-terminal
RSA + AES
TCP
RSA + AES
TCP


Mosh Server
Mosh Client
Application (e.g., emacs)
Pseudo-terminal
Mosh terminal emulator
Screen
Keystrokes
sender
receiver
SSP
AES-OCB
UDP
Synced objects
Terminal emulator (e.g., xterm)
Pseudo-terminal
Predictive local echo
Keystrokes
Screen
sender
receiver
SSP
AES-OCB
UDP

Solution 2: Speculative Local Echo and Editing

- Client anticipates server response.
- Runs predictive model in the background.
  - If user hits keystroke, predict key will appear where cursor was.
- Make predictions in *epochs*.
- If any prediction from epoch n is confirmed, show all predictions made in that epoch.
- If user does something difficult to handle, become *tentative* by incrementing epoch.

```
mosh: Last contact 10 seconds ago. [To quit: Ctrl-^ .]

Mosh Web site ideas

* What should it look like?

** Ideas

*** Boring free software Web site...

*** Old-timey newspaper: "Amazing remote shell program sweeps nation!!!!"

*** Make it look like a fake startup company. <-- Let's go with this.

* Benefits of Mosh

** Roam across Wi-Fi networks or to cell without dropping connection.

** More pleasant to type -- intelligent local echo is instant.

** No need to be superuser to install.

** Mosh doesn't fill up buffers, so Ctrl-C works quickly on runaways.

** Designed from scratch for Unicode; fixes bugs in SSH, other terminals.

** Free / open-source software.
-UU-:**--F1  site.txt        All L19    (Org)------------------------------
```

Demo

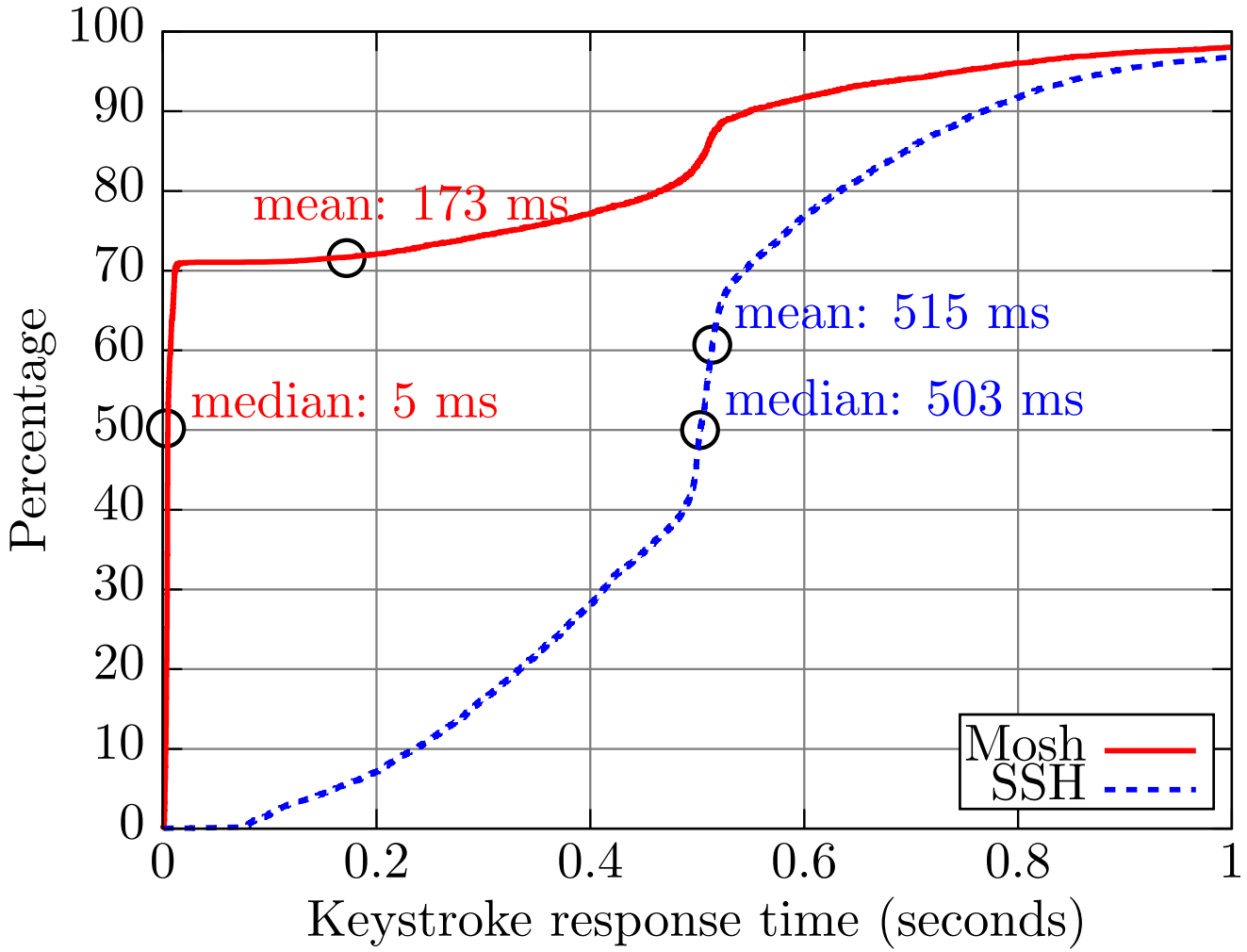
Benefits

- Roaming and suspend/resume:
  - Sleep and wake up later.
  - Change networks at will (Wi-Fi, cellular, wired, VPN).
- Helpful warnings:
  - ...if displayed state is stale.
  - ...if downlink working but not uplink.
- Interactive over flaky paths.
  - Works even across 50% loss paths.
  - Good interactivity even when RTT is > 100 ms.
  - Semantically appropriate flow control (won't fill up queues, Ctrl-C always works, no beeping fits).
- Security
  - Uses SSH to bootstrap: no privileged code, no daemons.
  - AES-OCB
- Better Unicode support.

Evaluation

- Collected 40 hours of terminal usage from six users.
- Covers 10,000 keystrokes using shell, e-mail, text editor (emacs and vi), chat, Web browser.
- Replayed over:
  - Sprint 1xEV-DO (3G)
  - Verizon LTE (4G)
  - MIT-Singapore
  - 50% loss path
- Result: 70% of keystrokes predicted instantly.
- Prediction errors $< 1\%$

Sprint 1xEV-DO cumulative keystroke response distribution



Evaluation (cont.)

Verizon LTE service in Cambridge, Mass., running one concurrent TCP download:

	Median latency	Mean	σ
SSH	5.36 s	5.03 s	2.14 s
Mosh	$<$ 0.005 s	1.70 s	2.60 s

Deployment

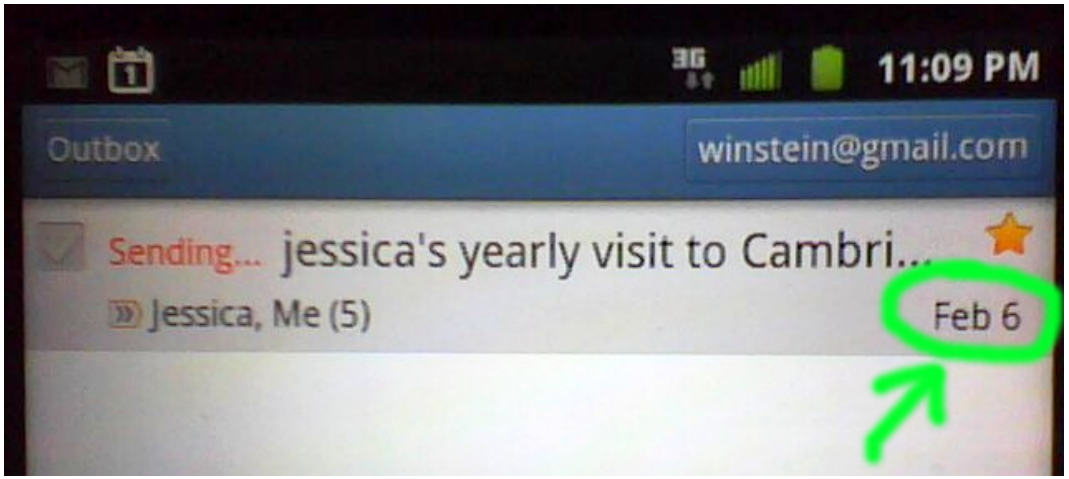
- Distributed in Debian, Ubuntu, Fedora, Gentoo, Arch, Slackware versions of GNU/Linux.
- Available via EPEL for Red Hat, CentOS, Oracle Linux.
- Included in MacPorts, Homebrew, FreeBSD ports collections.
- Works on Cygwin and Solaris, (very raw) on Android and iOS.
- News stories in April on Hacker News, Reddit, The Register, Twitter, Slashdot, Barrapunto.
- Top repository of the month on GitHub.
- 200,000+ page views, 50,000+ downloads, 1,200 "followers" of version control repo.

Reception

- "one of those times you don't realize something is broken until you see it fixed" — @xlfe
- "If you are an SSH user, check out mosh.mit.edu - the user experience really is dreamy." — @adamhjk
- "mosh is awesome. Tested it for two weeks and it really made my life easier: faster feedback and no more reconnects(!)" — @esmolanka
- "Finalement, la vie d'admin c'est pas si Mosh que ca" — @korben
- "There is very (if any) little research content." — USENIX review
- "ISO 2022 locking escape sequences oh flying spaghetti monster please kill me now." — USENIX review

State Sync Protocol for all?

- We believe SSP may be appropriate for many network problems.
- Android Gmail, Google Chat cannot roam without failure.
- **May 15, 2012**:



- Neither can Gmail (Web edition).
- These problems can be expressed as state synchronization.

Next Steps

- Mosh paper to be presented at USENIX ATC (June 2012).
- Essay to appear in *;login:* magazine.
- Mosh software under development by a team of contributors.
- We are working to apply SSP to mobile videoconferencing.
- We hope to show quantitative improvement on standard metrics (latency, quality), plus features like roaming.

Summary

- SSP is a secure datagram protocol that synchronizes abstract objects across a roaming IP connection.
- Mosh uses SSP to synchronize a terminal emulator with predictive local echo.
- In evaluations with 10,000 real-world keystrokes from six users, Mosh markedly reduced user-visible latency across several Internet paths.
- We think SSP will be useful for other applications as well.
- `http://mosh.mit.edu`