DoS: Fighting Fire with Fire

Michael Walfish,

Hari Balakrishnan, David Karger, and Scott Shenker[*]

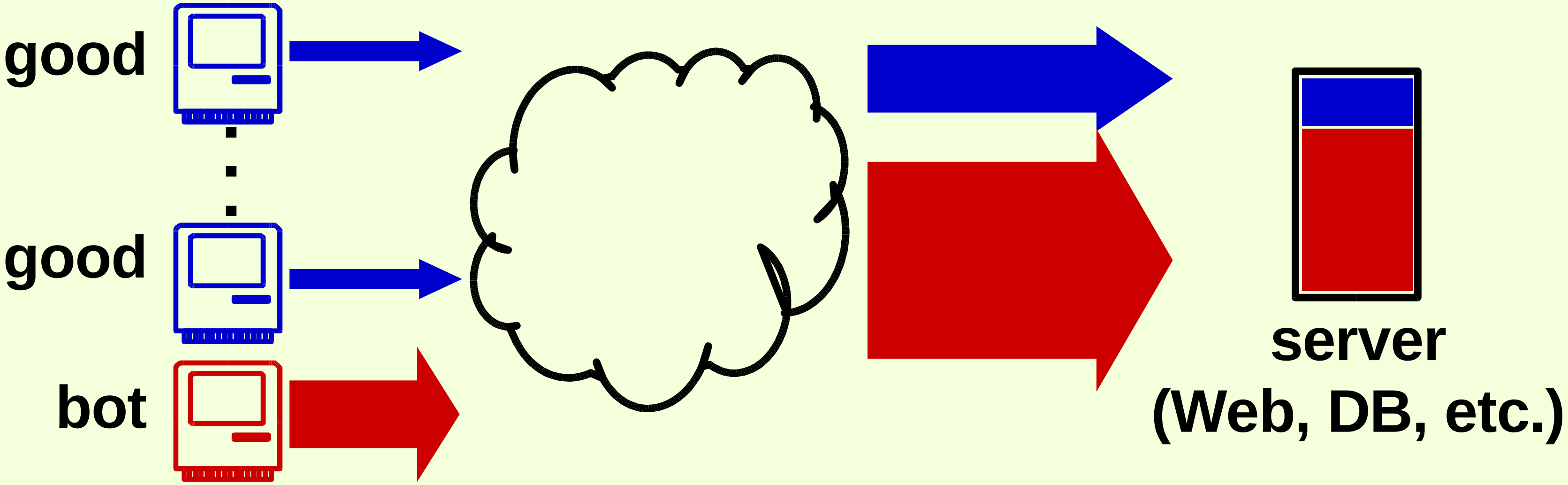
MIT Computer Science and AI Lab

[*] *UC Berkeley and ICSI*

15 November 2005

The Scenario and the Problem

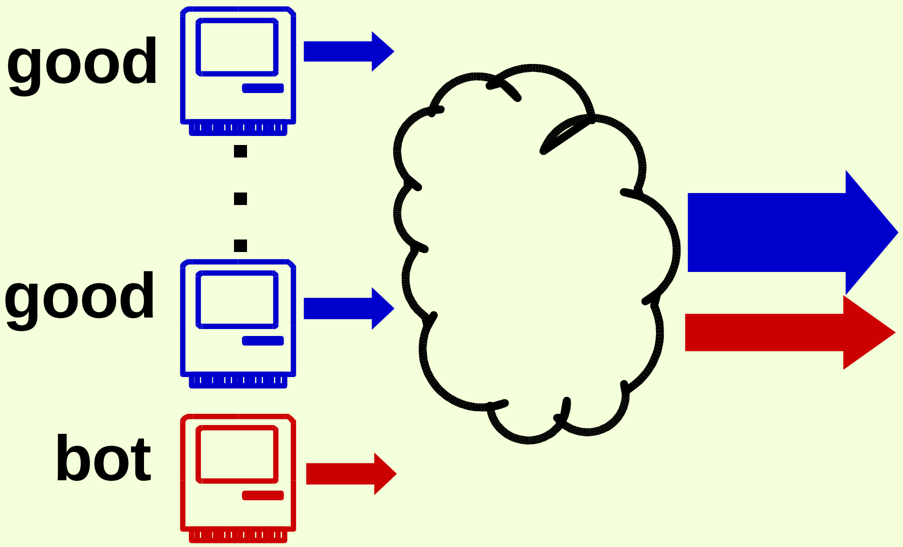
- Server with scarce computational resources:
  - CPU, memory, expensive DB software, etc.



- DDoS: many legitimate-looking requests from bots
- Hard to differentiate bots and good clients
  - Bots not anomalous, just heavy users
  - Proofs-of-humanity (CAPTCHAs) not ideal

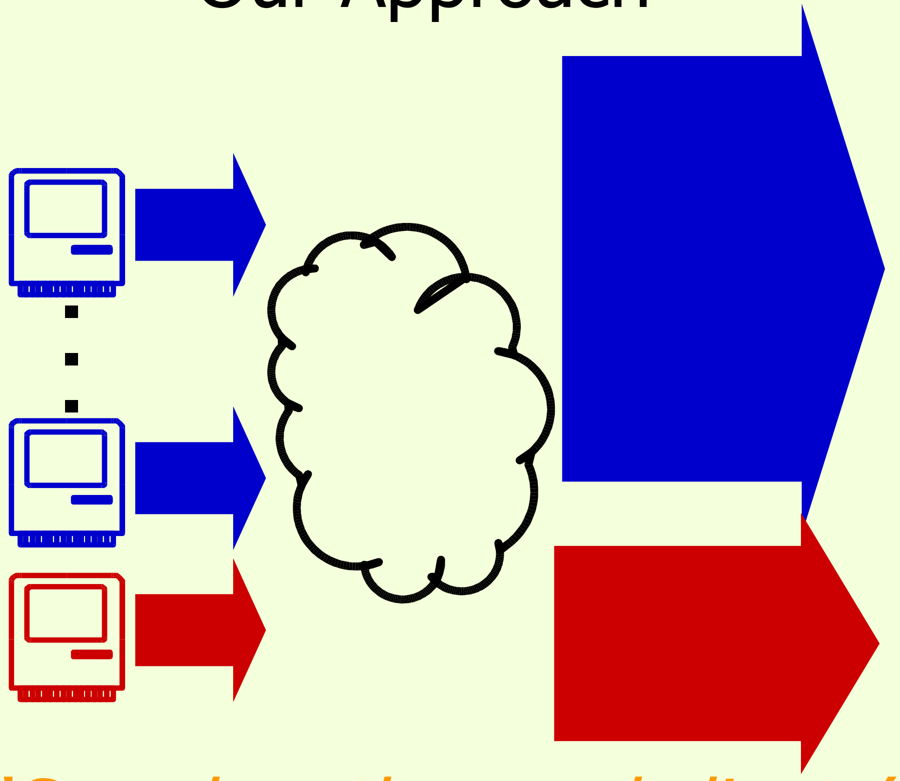
Goal: Bots Behaving Like Good Clients

One Possibility (e.g., IP throttling, proof of work)



"Slow down the bots"

Our Approach



"Speed up the good clients"

For now, assume more good clients than bots

Rest of the Talk

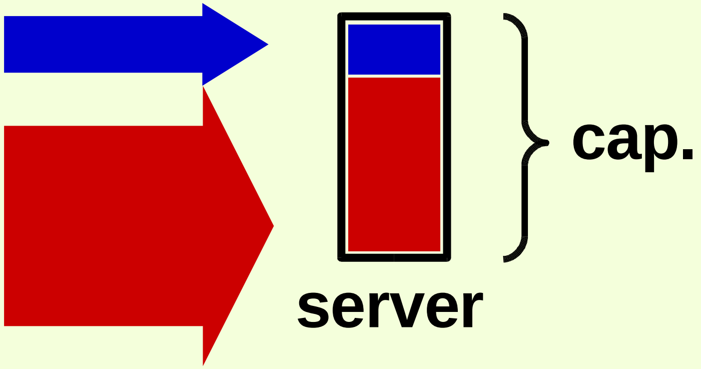
I. Mechanism

II. When useful?

III. Compare to other defenses

Assumptions and Status Quo

Status Quo

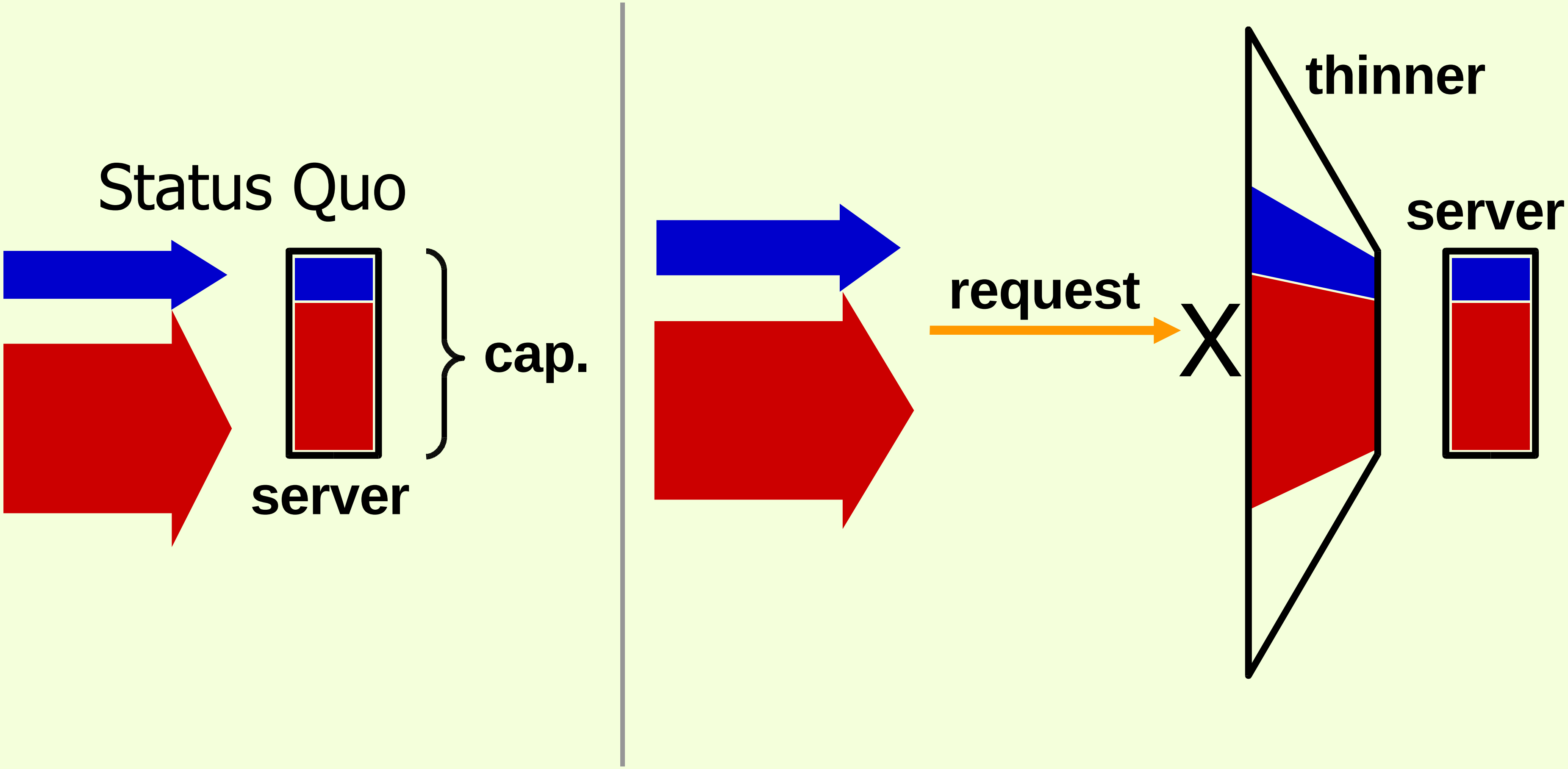


Assumptions

- Each bot sends at high rate
- More goods than bots
  - Will revisit
- Server capacity is known
- All requests cost server same
  - Paper relaxes this

Approach in a Nutshell

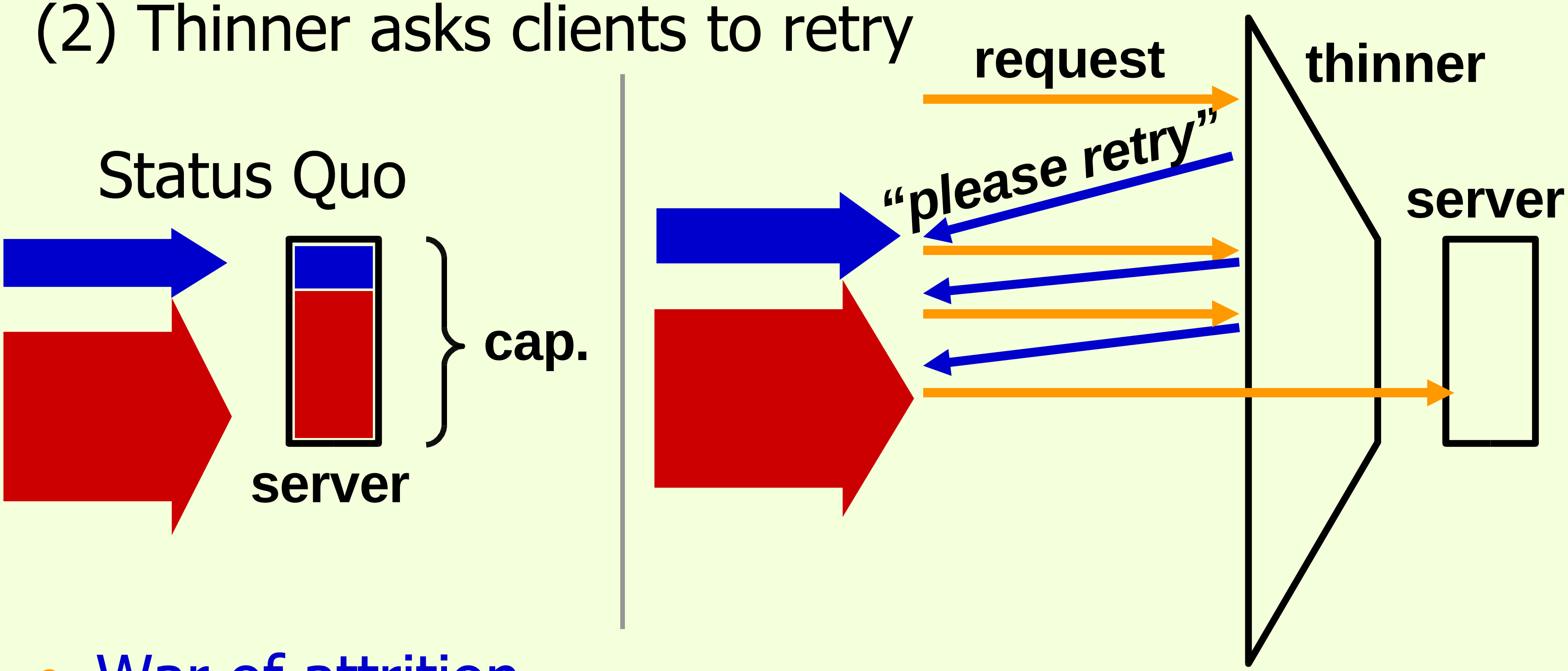
(1) *Thinner* (server front-end) randomly drops excess



Approach in a Nutshell

(1) *Thinner* (server front-end) randomly drops excess

(2) Thinner asks clients to retry

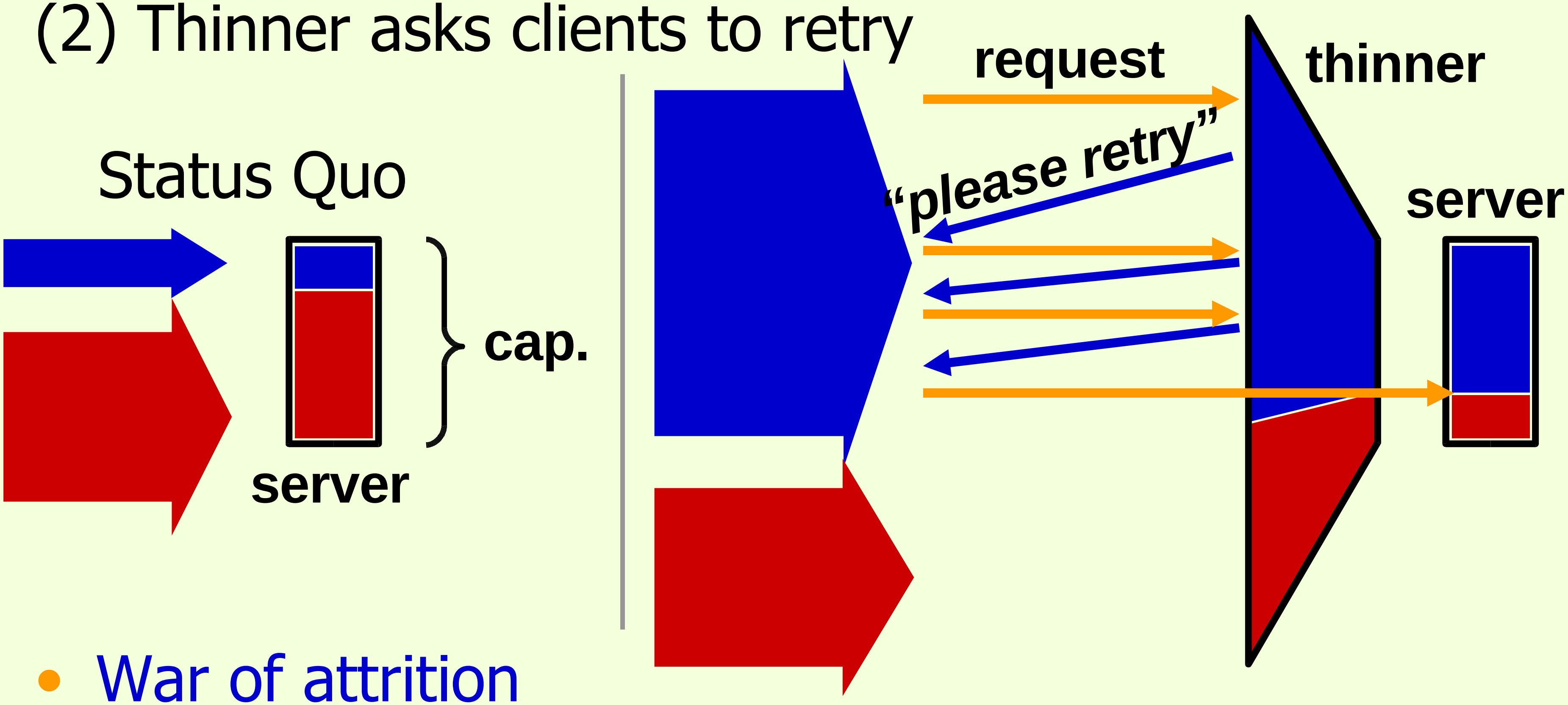


- War of attrition

Approach in a Nutshell

(1) *Thinner* (server front-end) randomly drops excess

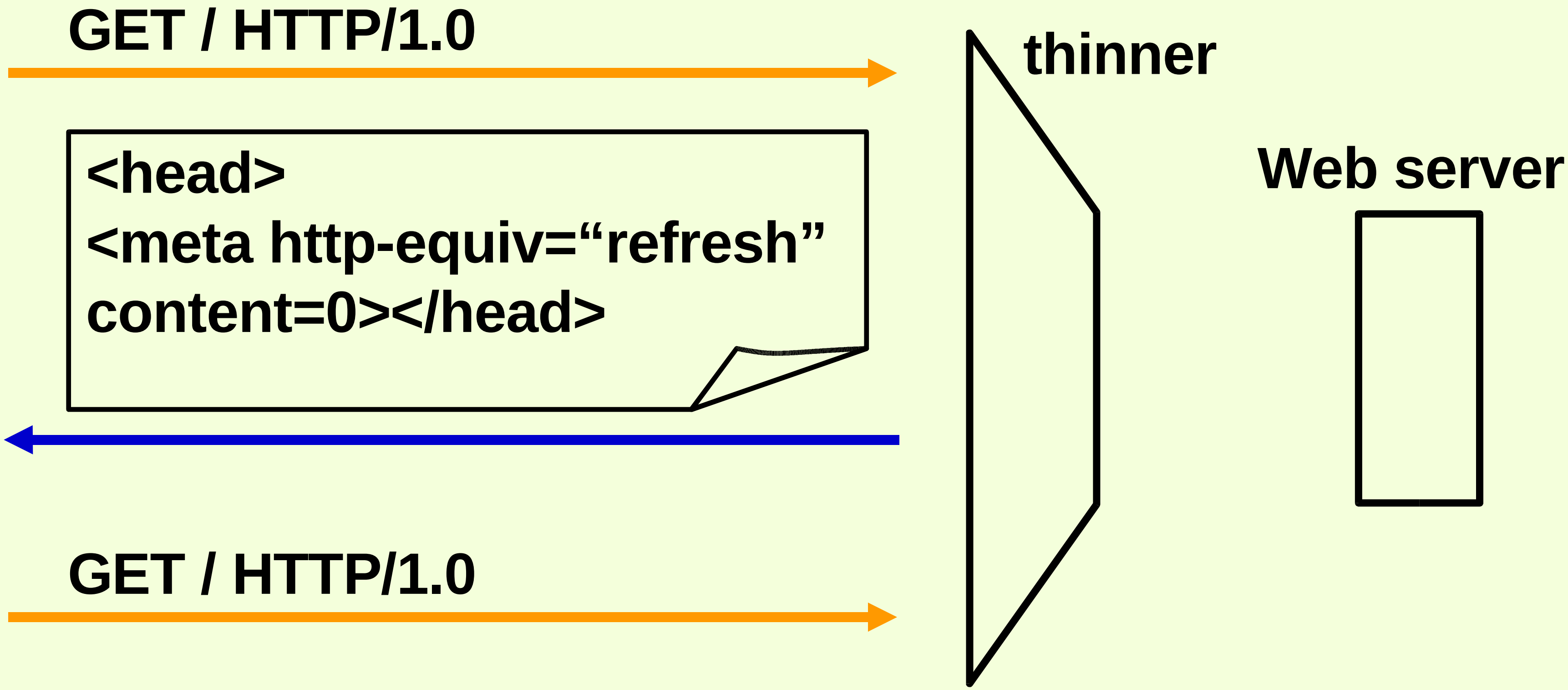
(2) Thinner asks clients to retry



- War of attrition
- Pay bandwidth to reach server: *proof of net-work*

Net-work for Web; No Client Changes

- Thinner is HTTP front-end
- “please retry” is automatic, zero-delay HTML refresh



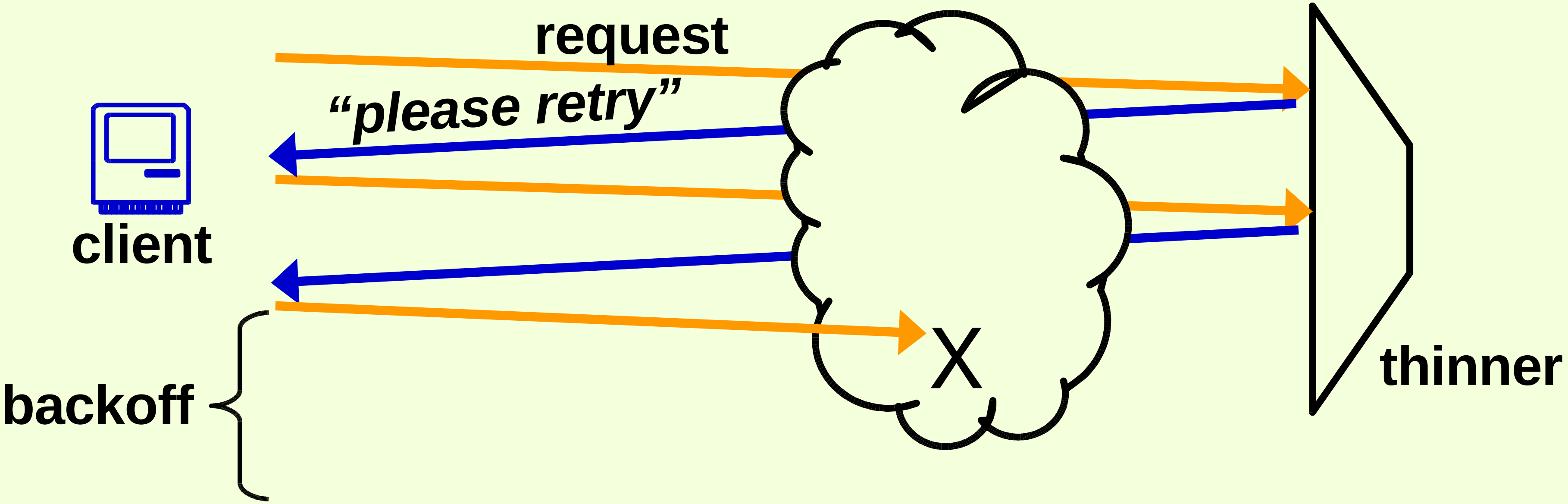
GET / HTTP/1.0

```
<head>
<meta http-equiv=“refresh”
content=0></head>
```

GET / HTTP/1.0

We Think This Won’t Harm the Network

- Standpoint of total capacity:
  - Core is over-provisioned (by rumor)
  - Inflation only in traffic to attacked sites
- Standpoint of transient congestion:
  - Application does consume more bandwidth …
  - … but controls congestion with packet conservation:



Outline

I. Mechanism

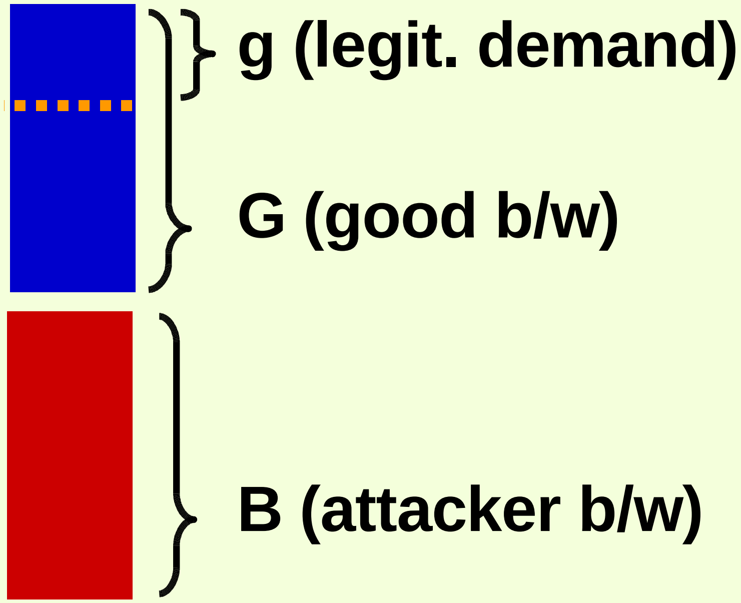
➡ II. When useful?

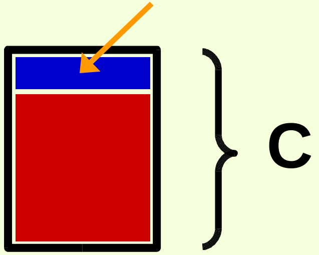
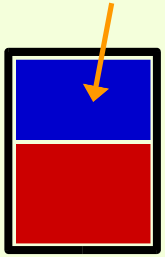
III. Compare to other defenses

When is Net-work Useful?

- You might think: goods need much more b/w than bots
- Not true!

Net-work Levels the Playing Field



Status Quo	With Net-work
$\left(\frac{g}{g+B}\right)C$  server's resources	$\left(\frac{G}{G+B}\right)C$  server's resources

- Net-work lets good clients capture up to $\left(\frac{G}{G+B}\right)C$
- Is a level playing field enough?
  - To satisfy good clients, need $\left(\frac{G}{G+B}\right)C \geq g$
  - Translates to *provisioning reqt*: $C \geq g(1 + B/G)$

Answering "When is Net-work Useful?"

- Provisioning reqt. now $g(1 + B/G)$; was $g(1 + B/g)$
- If $G >> B$ or $G \approx B$, provisioning reqt. not terrible
- If $G << B$? Likely, $C < g(1 + B/G)$. (eg, tiny flower shop)
  - Good clients still get better ratio
  - Global abilities of bots decrease
  - These are weak answers. Is there hope?
- Anecdotally: DDoS victims are popular sites and services
  - Not small flower shop

Outline

I. Mechanism

II. When useful?

➡ III. Compare to other defenses

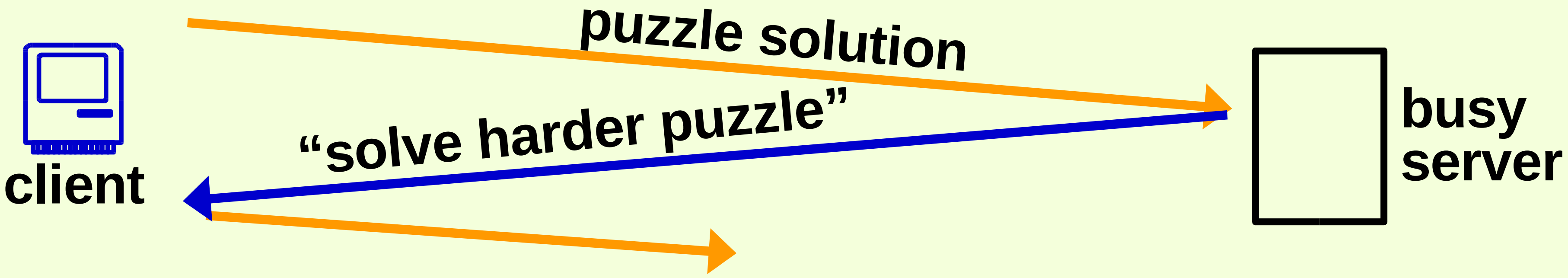
Net-work Uses Bandwidth as a Currency

- Other currencies: CPU cycles, mem cycles, money
- Price under net-work: # of retries (calc'd in paper)
- All currency schemes: attackers still get service
    - $C \geq g(1 + B/G)$ applies to all
    - To do better: must tell apart legit. and bot
    - Not always feasible, as discussed on slide 1

We now compare bandwidth to other currencies . . .

Advantages of Bandwidth as Currency

- Price (# of retries) emerges "naturally"
  - Clients aren't told price
  - They need not guess; just keep retrying
- Payment is observable (puzzles can be broken)
- Bandwidth plays a role in other currencies anyway:



Disadvantages of Bandwidth as Currency

- Possibly undemocratic: low bandwidth clients
    - Good point
- Some customers pay per-byte
    - But most servers aren’t attacked most of the time

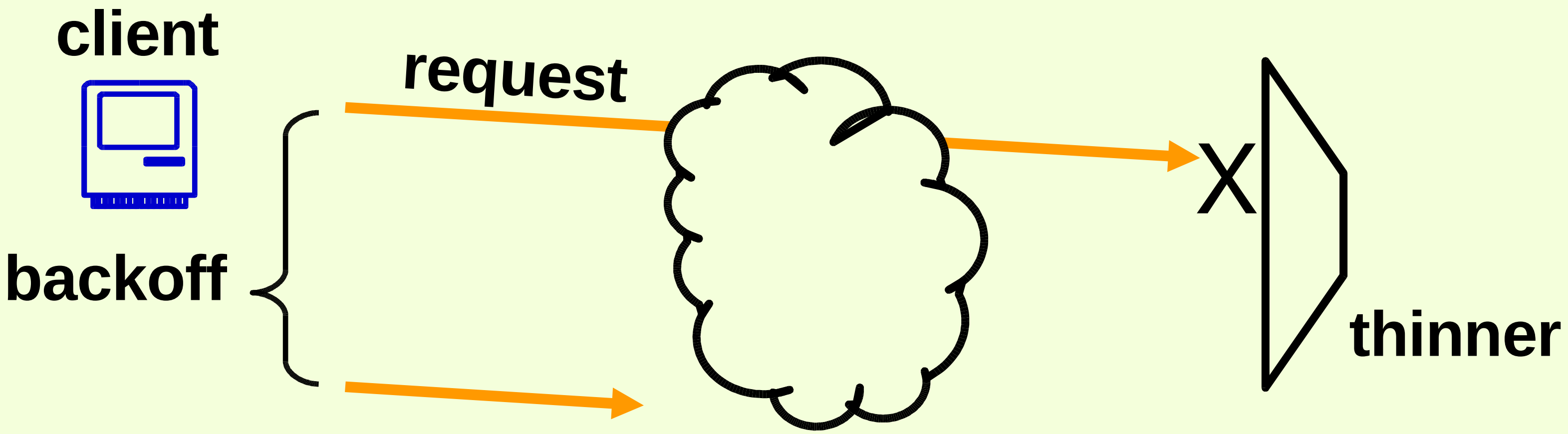
At the End of the Day

- Is this Internet vigilantism?
- Net-work treats bots and legitimates equally
  - Is a level playing field enough?
  - Depends
- Is bandwidth the right way to level the playing field?
  - Possibly more undemocratic
  - More natural than other currencies

Appendix Slides

Thinner Needs Lots of Bandwidth

- Thinner must be uncongested



- So much bandwidth may be expensive. Solutions?
  - Co-locate thinner?
  - Service provider or overlay? (i3, Mayday, SOS…)

Why not . . .

- . . . Proof-of-humanity (CAPTCHA)?
  - Assumes human clientele
  - Not all humans want to answer CAPTCHA [Killbots]
- . . . IP throttling?
  - Source address spoofing for UDP requests
  - Attackers hijack IP space with bogus BGP advts.
  - NAT (many clients, lots of bandwidth; one IP addr)
- . . . Capabilities?
  - Good point
  - These aren’t exclusive; combine them?

How Many Retries?

- Recall provisioning requirement: $C \geq g(1 + B/G)$

- If provisioning requirement satisfied:
  - Average number of retries is $B/(C - g)$
  - See paper for simple derivation

- If provisioning requirement not satisfied:
  - Good clients spend everything, G
  - Allow probability is $C/(B + G)$
  - Average number of retries is $(B + G)/C$

Extension

- If request-retry loop brings unacceptable latency…
- … thinner can explicitly calculate price, r retries
- Price is ratio of inbound request rate to capacity
- Thinner communicates price, r, to clients
  - Clients send r-1 retries over cong.-controlled stream
- Still “natural”?
  - Yes.
  - Easy for thinner to calculate price

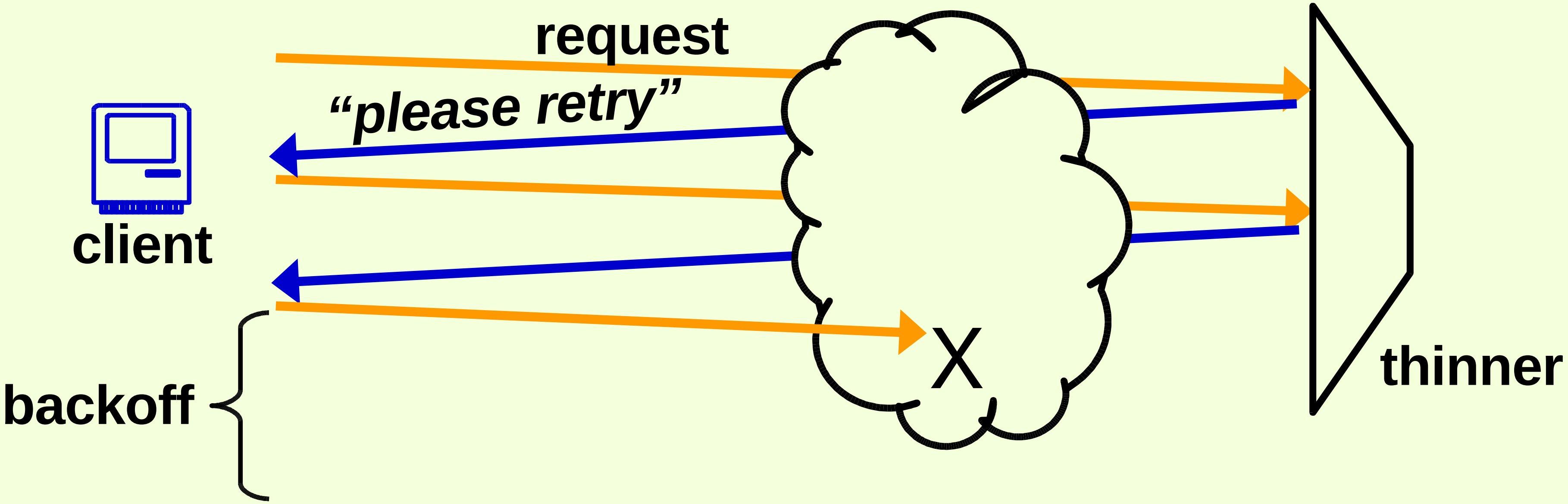
Is Upload Bandwidth the Right Constraint?

- What if constraint is clients' download bandwidth?
  - Much of net-work still applies

- Why think server's computational resources more expensive than its bandwidth?
  - Enterprise application licenses are expensive
  - Requests can be tiny yet cause much work (e.g., travel sites)

“But Bots Won’t Control Congestion . . .”

- Recall picture:



- Bots won’t be so polite in their malfeasance
- True
- But failing to back off is a link attack; exists today