

An Internet of Ends

Diego Doval

diego@clevercactus.com

A little about me

- B.Sc. Drexel University, Philadelphia, PA
- Research associate at IBM Research in Yorktown Heights, NY
- First engineer at a startup in Palo Alto, CA
- NTRG@TCD (PhD)
- Co-founder & CTO, clevercactus ltd.

We're at a Tipping Point

- **Fact #1:** The Internet is losing its end-to-end, peer-to-peer nature.
- **Fact #2:** Decentralization and self-organization can help us re-create the Internet based on its original principles.

The bad news

- What we have today has largely evolved without a master plan. Fixing it requires not only coming up with a solution, and not only a solution that leaves the old system intact, but also one that can self-propagate.
- There are powerful economic and technological forces that prefer #1 to #2
- It's up to us to make it happen.

The good news

- At the risk of indulging on syllogisms, by its nature the solution is self-propagating.
- It's not just theory, most of what we need is already there. Even more, the push for decentralization is happening at all levels, physical (AdHoc 802.11), logical (Ad Hoc protocols, some webservices), and content (Weblogs, social networks).
- It's up to us to make it happen.

So the Internet is losing *what*?

- *Its end-to-end, peer-to-peer nature.*
- That is, the Internet is losing the ability of any host connected to it to act both as a server and a client.
- Why? Basically, because we have completely mixed up notions of trust, security, and authentication.

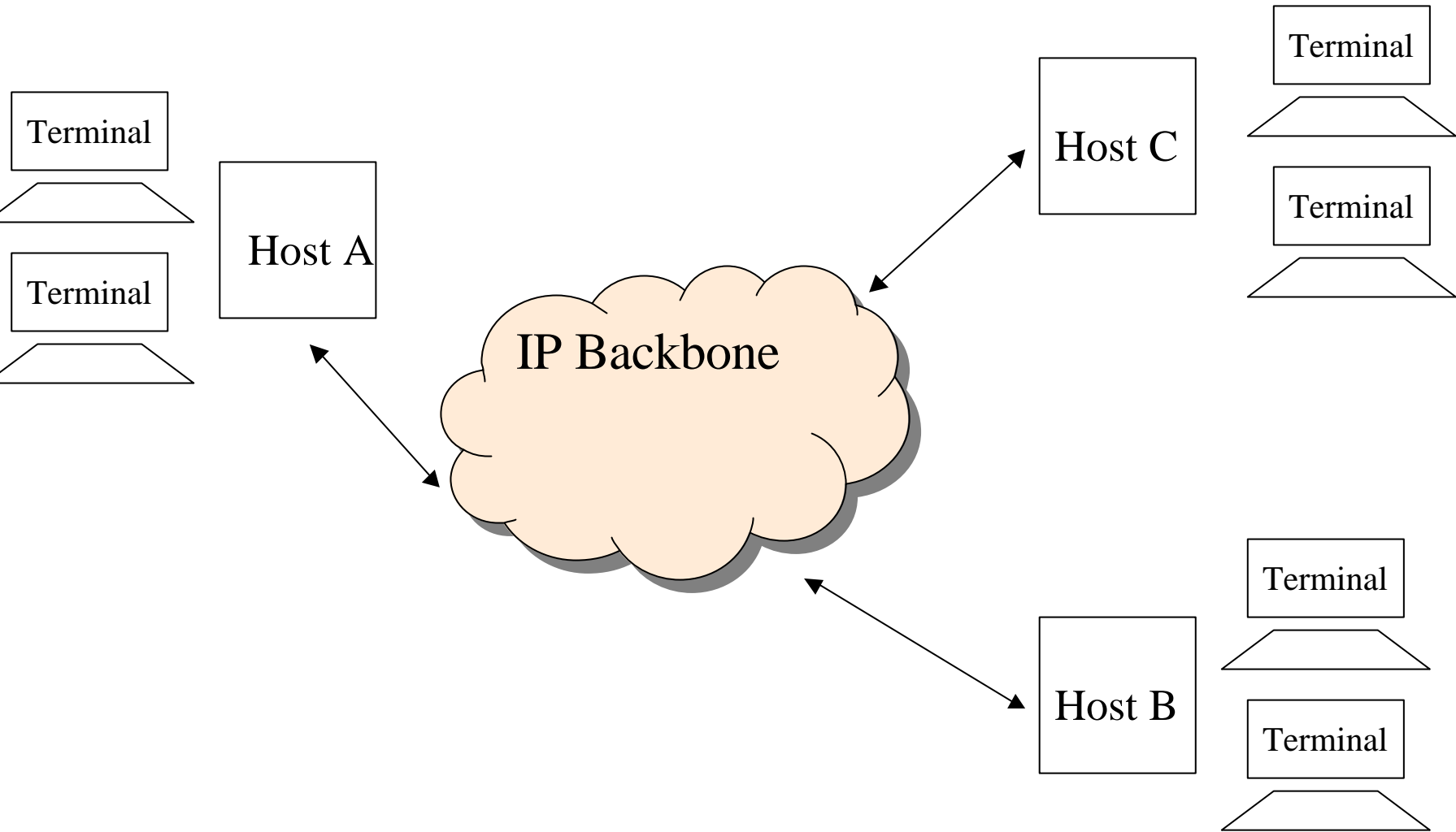
Three Definitions

- *Trust*: knowing that whoever you're talking to is a legitimate peer in a transaction
- *Authentication*: making sure that you are who you say you are (e.g., a login)
- *Security*: knowing that a properly authenticated connection from a trusted peer or user is secure from attacks (e.g., encryption)

The way it was

- Originally, the Internet was peer-to-peer in the most basic sense.
- Hosts talked to each other directly, the backbone carried IP packets – end of story.
- Trust was literally *on the wire*: if you had a connection to the backbone, you were a trusted host. Trust was literally a mapping of real-world relationships.
- Authentication/Security was handled at the application level (think *telnet*).

The way it was (2)



But then...

- As PCs entered the picture (and with them LANs) the old trust system didn't work anymore.
- Early PCs, incapable of handling more than a dialtone, naturally tended towards the notion of client-server. Allocation of IPv4 addresses started to be a concern.
- Corporations in particular saw locally centralized access as a bonus, a form of control.
- But the worst hadn't happened yet...

The Morris Worm

- “*There may be a virus loose on the internet.*” Andy Sudduth of Harvard University, 34 minutes after midnight, Nov. 3, 1988.
- The Morris Worm was a watershed event. It *proved* that what up to that point had been theoretical concerns were in fact very real.

The Morris Worm (2)

- The Worm piggybacked on a well-known vulnerability in *sendmail*.
- *But* at its core what it was really doing was exploiting the “wire-based” nature of trust in the Internet.
- Why? Because, if you “trust the wire”, but *anyone* can be wired, then security and authentication are suddenly the first and last line of defense. If one of those fails, the other might just as well not be there.

And yet...

- The Morris Worm was largely seen as a security hole in a certain application, rather than the manifestation of a systemic problem.
- This gave steam to the endless cycle of security patches that we still live with to this day.
- Conclusion: In the end, not much changed.

Enter the Web – and email

- The Web + email became the killer app that drove millions into the Internet
- ISPs and LANs (both corporate and academics) created an explosion in number of clients, threatening to consume the IPv4 address space.
- Additionally, the still-ignored trust problem compounded security holes and authentication concerns (e.g., dumpster-diving).
- So what did to do?

Eureka?

- These problems were recognized to varying degrees; a solution was necessary.
- Client-server was reinforced as the main solution due to these and other factors.
- In this context, two “solutions” evolved: Firewalls and NATs.

Firewalls

- The “Burning the village in order to save it” solution. :)
- In other words: since IP traffic can't be trusted, let's kill all traffic (instead of fixing the trust problem).

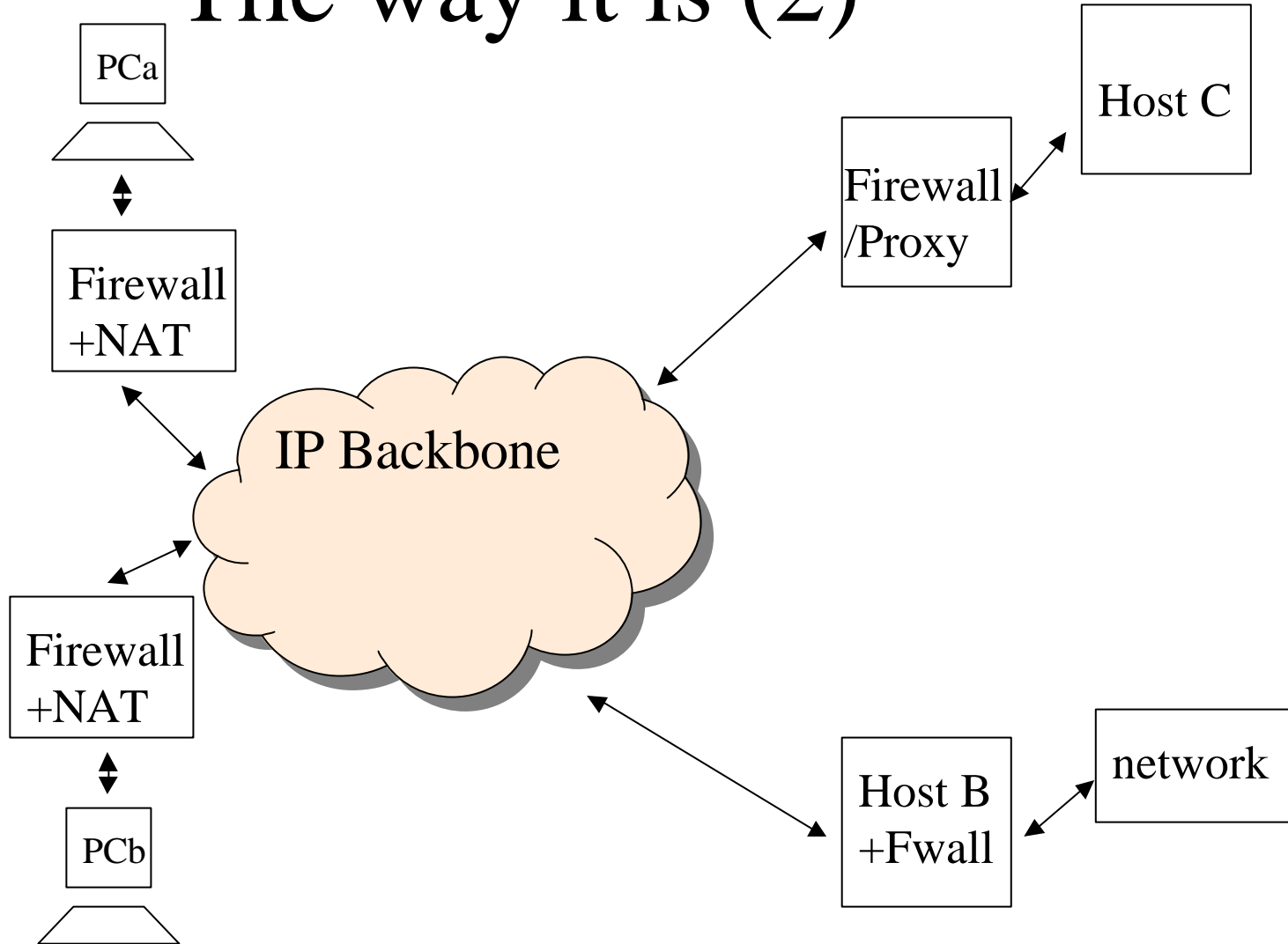
NATs

- NAT (Network Address Translation) hides a complete IP subnet behind a single IP address, artificially enlarging the IPv4 address space.
- In the process, it *also* destroys the ability of a machine outside the NAT box to reference a host inside of it.

The way it is

- The Internet today is a series of walled gardens heavily dependent on centralized infrastructure (e.g., DNS, the Web).
- Trust... is still on the wire. “Security” is provided by stopping traffic outright. Authentication is still based on logins.

The way it is (2)



The way it is (3)

- NATs remove the ability of hosts to reference each other directly
- Even if port-mapping is configured (don't count on it), it's more than likely that a Firewall somewhere will stop the traffic. (Proxies can be considered as a special type of firewall).
- This is bad. But what's worse is that Firewalled NAT boxes are being sold by the tens of millions as access points for broadband connections.

And still...

- Worms and viruses are rampant.
- Security patches are sometimes part of the problem.
- After all this investment and work, *we haven't solved the problem at all.*

But P2P apps...

- P2P Apps today for the most part depend on transfer-proxies, that is, systems that are unprotected (mostly by default) to transfer the data.
- This is obviously unsustainable in the long run.

A solution

- A solution to these problems has to come from the application level, rather than, say, the protocol level (i.e., IPSec on IPv6 isn't going to do much).
- **We need to** return to the end-to-end nature of the Internet, move away from the hub-based system that is entrenching itself today and **let the ends be what matters.**

A solution (2)

- This requires, first, a change in how applications define trusted interactions across the network.
- Second, plain logins have to give way to more meaningful authentication (e.g., certificate-based)
- Third, security has to be based on the needs of the application (e.g., strong encryption for personal communications, no encryption for linux distro downloads)

A solution (3)

- *Trust should be user-defined.* Applications must allow users to map their real-world trust relationships and treat data accordingly based on the trust level of sources
- Once trust is properly defined, authentication and security become easier to define as well.

This isn't new!

- No.
- *But our constant focus on protocols, underlying technologies and low-level interactions is ignoring the wealth of context information that actual use, and yes, users, can give us.*

So what you're saying is...

- We need to start looking at applications top-down. Forget the protocol, the fancy OR diagrams and high-speed M-Tree traversal.
- Create applications based on what *users need* rather than based on what the underlying technologies can do (best example: email).
- The problem: this means we actually have to start listening to users! (the horror!)

And how would this help?

- We're at a transition point. Applications created today still can squeak through and cross these barriers.
- Once these applications form an installed base, the infrastructure will adapt to them over time.
- At a minimum, the need for further low-level barriers can be stopped, and we can all go back to using and developing software instead of installing security patches and complaining about spam and worms.

What we need

- Blank-slate R&D on communication interfaces and applications, taking into account these factors from the start.
- R&D into self-organizing, self-healing, loosely coupled systems. Keyword: biomimetic.
- Shameless plug: NTRG <http://ntrg.cs.tcd.ie/>
- Even more shameless plug: clevercactus <http://www.clevercactus.com/>

For more information

- Check my weblog for references:
<http://www.dynamicobjects.com/d2r/>
- Or write: diego@clevercactus.com

Questions?