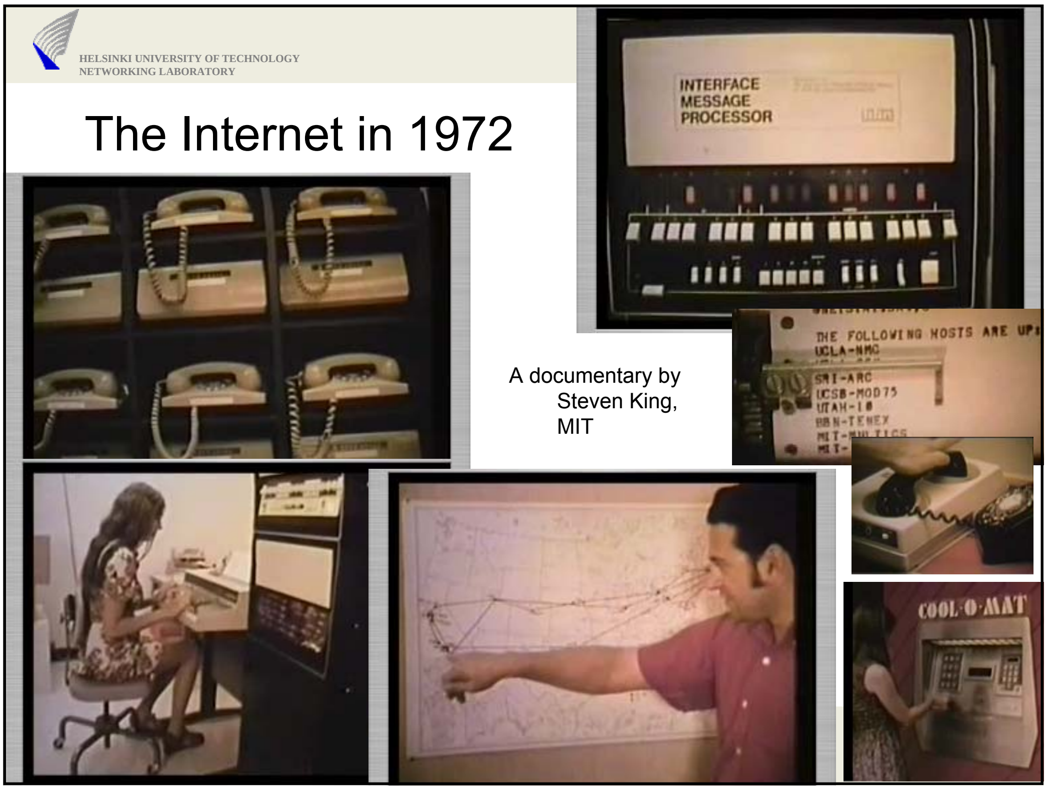
Beyond Technology:

# The
# Financial and Political Layer

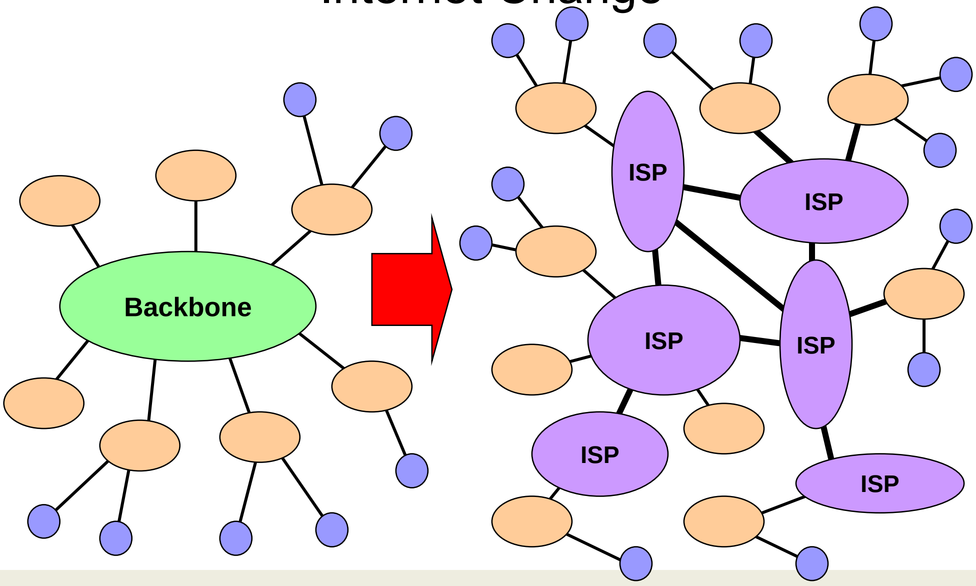
Protocol Design

## The Internet in 1972



A documentary by
Steven King,
MIT

# Internet Change



# How is the Internet paid for?

- Generally: cost is distance insensitive
  - Strong promoter of globalization
  - There are some incentives to keep traffic local, though (Throughput ~ 1/RTT)
- Dial-up
  - per minute (peak hours, off-peak)
  - monthly flat rate
- Direct connection
  - volume bands or per “k bytes”
  - more likely: flat rate
  - typically independent of time and destination
- Attempt to change:
  - pay for reserved bandwidth?
  - pay for enhanced service profiles (market differentiation)
- Trend: pay for additional services
  - Within the provider’s network only

# Who runs the Internet?

- “*Nobody*”
- Network: site network providers, ISPs (Internet Service Providers), NAPs (Network Access Providers), ...
  - Trend towards “value-added services” beyong simple packet carrier
- Lines/Fibers: telephone companies, railroads, utilities, ...
- Names and Numbers:
  - ICANN (Internet Corporation for Assigned Names and Numbers)
  - Numbers: IANA (Internet Assigned Numbers Authority)
  - Names: RIPE (Europe), ARIN (USA), APNIC (Pacific)
- Standards: IETF
- Technology: vendors (standards-based + proprietary)
- Content: “*everybody*”

# The Internet Landscape Today

- Users
- Commercial ISPs
  - Working for profit
- Private sector network providers
- Governments
  - Want to care, need to care
- Intellectual Property Right (IPR) holders
- Providers of content and higher level services
  - Streaming, telephony, media, ...

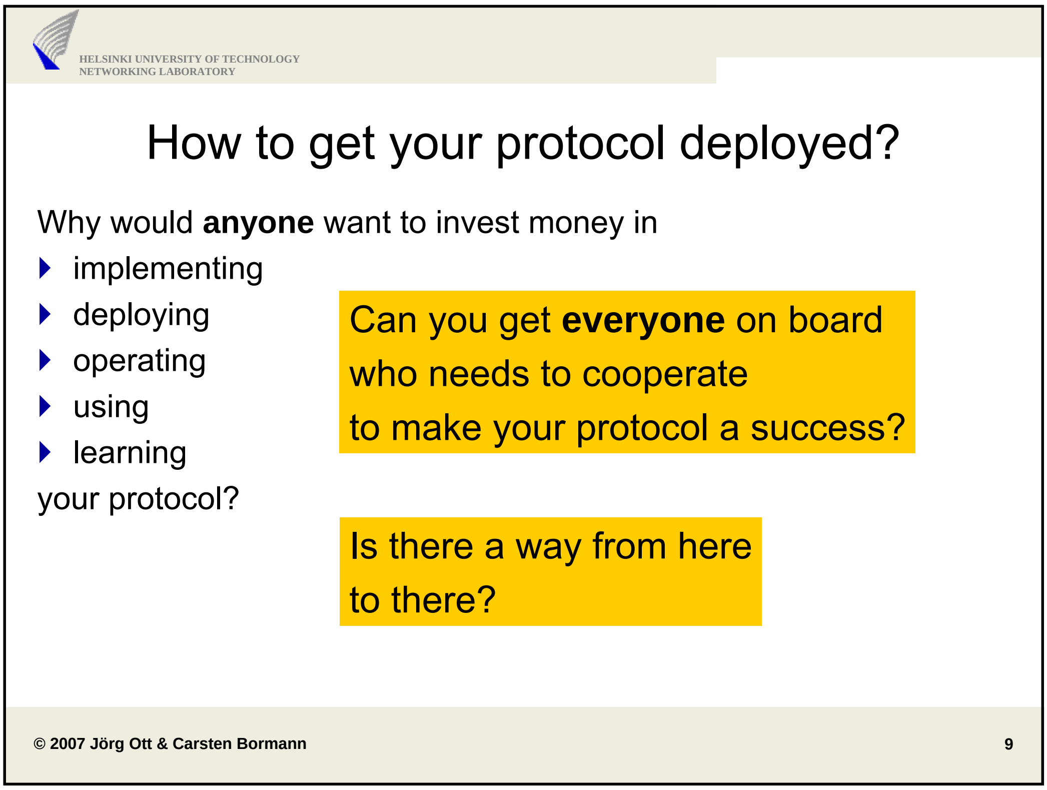
- Tensions between interests of the various parties
- “Support” for applications, users, etc.

# Changes over time…

- From closed academic environment to global society
  - Trusted users ➔ non-trusted users
  - Users who know what they do ➔ users who don’t want to (need to) know
- From research to commercial
- New stakeholders in the Internet
  - Internet Service Providers (ISPs)
  - Application Service Providers (ASPs)
  - Governments
- Third parties (to facilitate interactions)
  - Trusted entities, caches, proxies, ...
- ...

# Protocol design does not happen in a vacuum

- With exceptions:
  - Some protocols never leave the □closed environment they were designed for
  - but many surprisingly do!
  - It makes sense to think bigger
    - It also makes sense not to burden a design with issues it need not be burdened with
  - Use judgement.
- Even so:
  - staying in the mainstream will make life easier for those poor people that will have to maintain your protocol in the future.
  - you have to “sell” your protocol within your own organization
    - which may have a slightly different, but still quite difficult, “political” situation.



# How to get your protocol deployed?

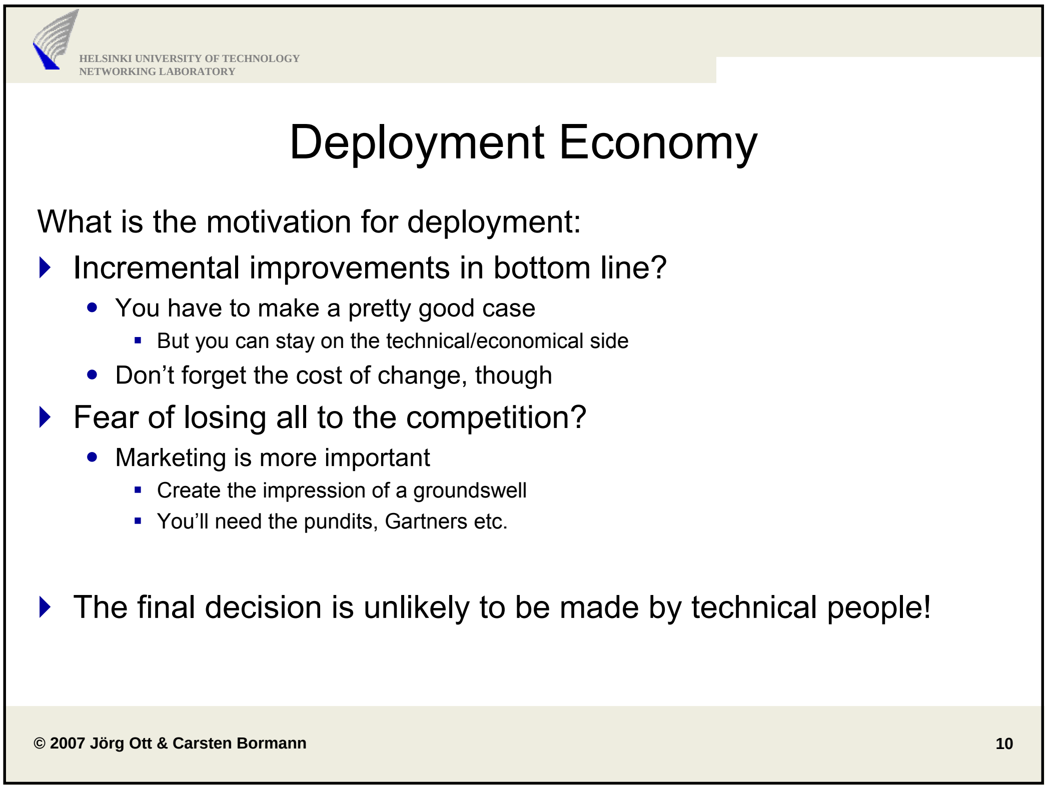
Why would **anyone** want to invest money in

- implementing
- deploying
- operating
- using
- learning

your protocol?

Can you get **everyone** on board who needs to cooperate to make your protocol a success?

Is there a way from here to there?

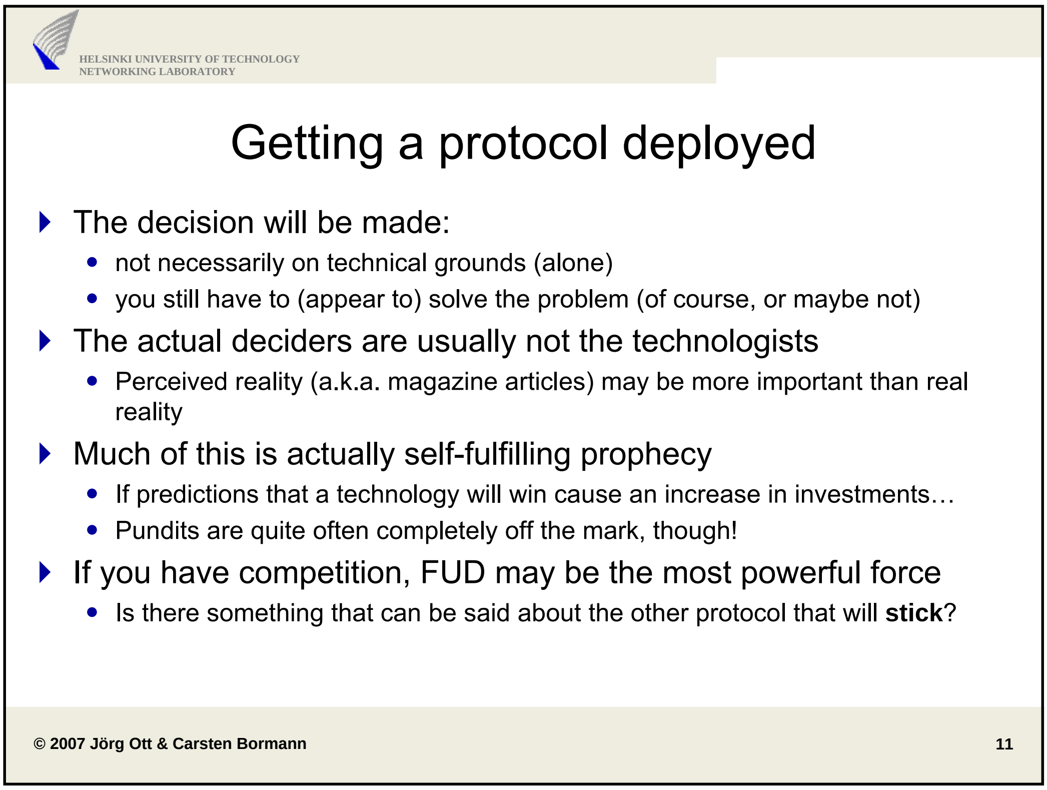


# Deployment Economy

What is the motivation for deployment:

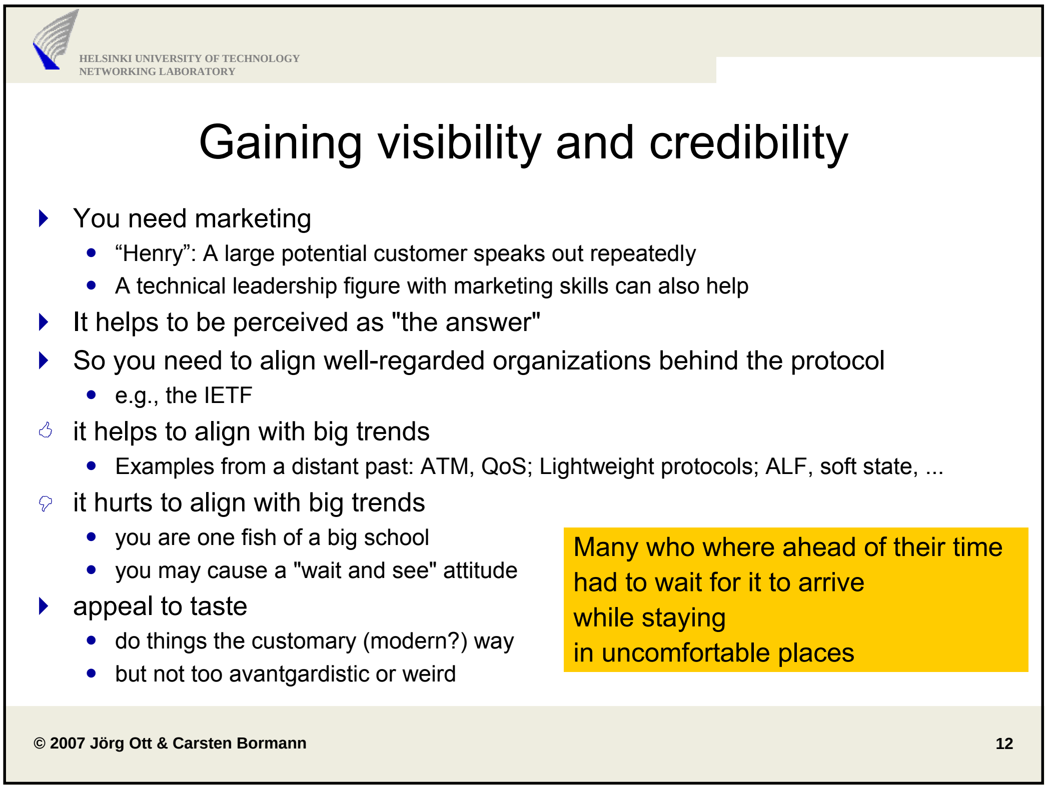
- Incremental improvements in bottom line?
  - You have to make a pretty good case
    - But you can stay on the technical/economical side
  - Don’t forget the cost of change, though
- Fear of losing all to the competition?
  - Marketing is more important
    - Create the impression of a groundswell
    - You’ll need the pundits, Gartners etc.

- The final decision is unlikely to be made by technical people!



# Getting a protocol deployed

- The decision will be made:
  - not necessarily on technical grounds (alone)
  - you still have to (appear to) solve the problem (of course, or maybe not)
- The actual deciders are usually not the technologists
  - Perceived reality (a.k.a. magazine articles) may be more important than real reality
- Much of this is actually self-fulfilling prophecy
  - If predictions that a technology will win cause an increase in investments…
  - Pundits are quite often completely off the mark, though!
- If you have competition, FUD may be the most powerful force
  - Is there something that can be said about the other protocol that will **stick**?



# Gaining visibility and credibility

- You need marketing
  - “Henry”: A large potential customer speaks out repeatedly
  - A technical leadership figure with marketing skills can also help
- It helps to be perceived as "the answer"
- So you need to align well-regarded organizations behind the protocol
  - e.g., the IETF
- 👍 it helps to align with big trends
  - Examples from a distant past: ATM, QoS; Lightweight protocols; ALF, soft state, ...
- 👎 it hurts to align with big trends
  - you are one fish of a big school
  - you may cause a "wait and see" attitude
- appeal to taste
  - do things the customary (modern?) way
  - but not too avantgardistic or weird

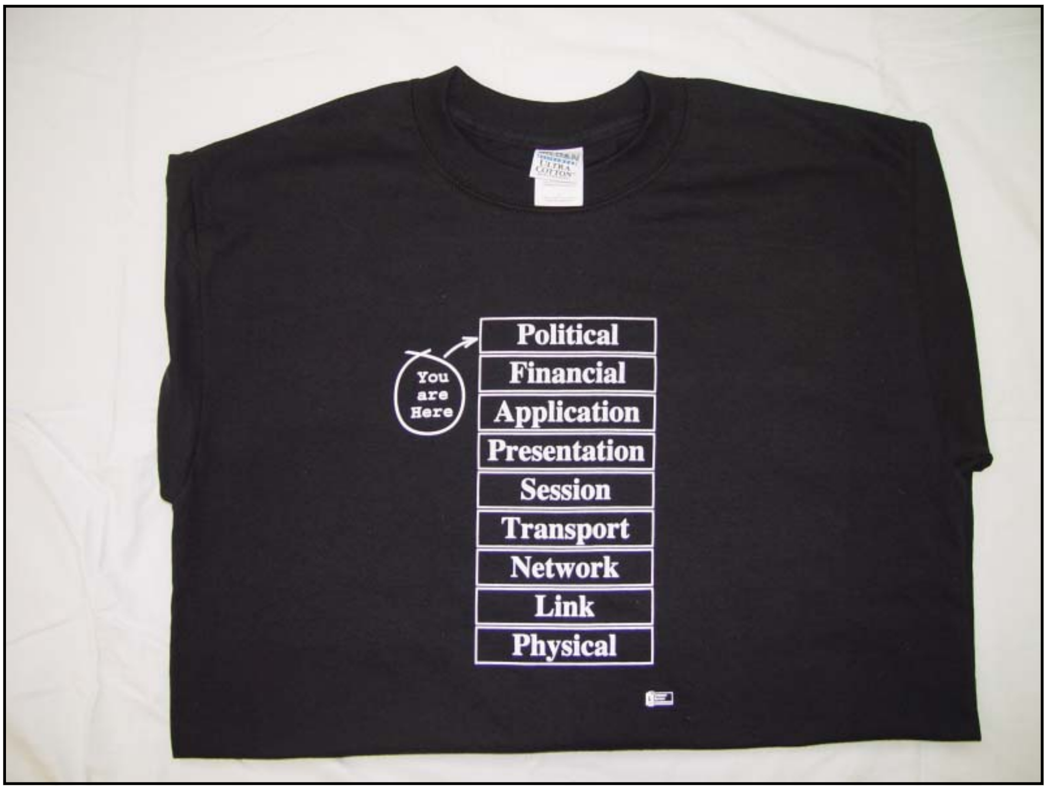
Many who where ahead of their time
had to wait for it to arrive
while staying
in uncomfortable places

# Don’t put in showstoppers

- Make sure deployment does not depend on factors you cannot control
  - don't commit error 33
- Make sure you don’t turn up on the losing side of a market fight
  - hard to predict!
  - make sure your protocol is not perceived as aiding that side
- Patents (see later)

# Be timely

- Moore's law is going to negate any performance benefit if its complexity causes delaying productization
- **release early, release often**
  - but then, make sure you don't get known for a losing release
  - creating one big splash may also be important for marketing (if it comes in time)
- an open-source **implementation** will help tremendously
  - helps the technologists understand the issues
  - demonstrates concept (to technologists and deciders)
  - eases entry (as a reference or as the actual implementation going live)
    - builds out your coalition
  - can be used for interop testing
  - allays fears of a “cabal protocol” that can only be implemented by an in-group of expensive consultants
  - (and helps debug your protocol as well)



# Is your protocol “just technology”?

Will your protocol be **used** for

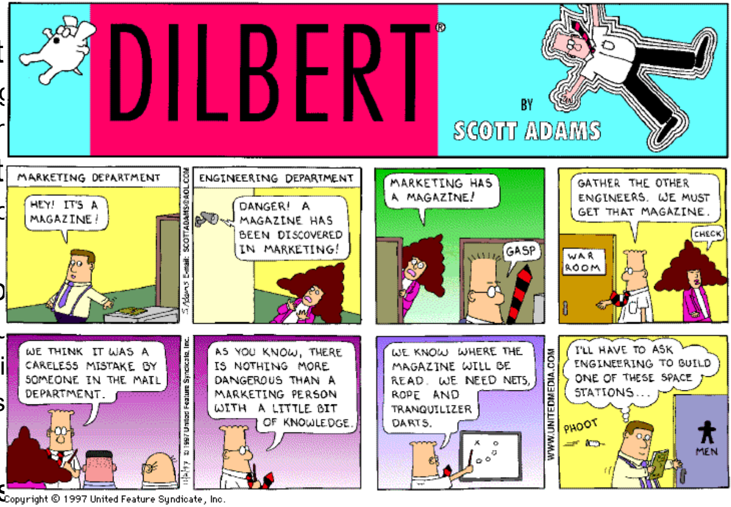
- improving efficiency in an existing market
- creating a market
- impeding creation of a market
- furthering political change
- impeding political change

or all of the above?

To be successful, protocols need to interact properly with the financial and political space.

# The decision makers are fighting a different fight

- Posit
  - E.( ck
  - Fir
- Posit
  - Mc
- Supp sts
  - Mu
  - Pri
  - Us
- “Tuss


DILBERT®
BY SCOTT ADAMS
MARKETING DEPARTMENT
HEY! IT'S A MAGAZINE!
S.Adams E-mail: SCOTTADAMS@AOL.COM
ENGINEERING DEPARTMENT
DANGER! A MAGAZINE HAS BEEN DISCOVERED IN MARKETING!
MARKETING HAS A MAGAZINE!
GASP
GATHER THE OTHER ENGINEERS. WE MUST GET THAT MAGAZINE.
CHECK
WAR ROOM
WE THINK IT WAS A CARELESS MISTAKE BY SOMEONE IN THE MAIL DEPARTMENT.
AS YOU KNOW, THERE IS NOTHING MORE DANGEROUS THAN A MARKETING PERSON WITH A LITTLE BIT OF KNOWLEDGE.
WE KNOW WHERE THE MAGAZINE WILL BE READ. WE NEED NETS, ROPE AND TRANQUILIZER DARTS.
WWW.UNITEDMEDIA.COM
I'LL HAVE TO ASK ENGINEERING TO BUILD ONE OF THESE SPACE STATIONS...
PHOOT
MEN

# Guidelines for keeping protocols out of trouble (1)

- Design to win regardless of outcome
  - The tussle should take place within your design, not distort it
- Do not design to dictate the outcome
  - You may have a preference, but the opponents will fight you and your protocol
- “Provide Mechanism, not Policy”
  - The right policy may not even have been invented at deployment time
  - (But then, it is hard to design mechanism that can support **any** policy)
- Isolation of conflicts of interest: If there are tussles, separate functions in the tussle from those outside the tussle
  - Even if there is no technical reason

# Guidelines for keeping protocols out of trouble (2)

- Design for choice
  - E.g., decentralize, allow for parameters selecting entities, etc.
  - May require its own set of protocols: e.g., number portability
- Design for change
  - Assumptions may not hold forever — don’t wire them into the protocol
  - May need to take explicit action to maintain changeability during protocol evolution
  - Resist short term optimizations for specific uses or operation points
    - But then: may have to compromise to encourage deployment

# Limitations of Protocol Design

- Remember:
  Don’t try to provide technical solutions for every social problem; some problems need to be solved in a non-technical fashion!

E.g.:

- Floor control in small conferences is best done socially
- Hardening security may cause people to route around it
  - E.g., password expiry schemes lead users to choose guessable passwords
  - People may entirely avoid a protocol if its security is too cumbersome
- Providing a little technical help for social processes is OK, though
  - Cf. Slashdot moderation points

# Further Tussle: Regulation

- The market is often not left alone to decide
- Governments (have to) pursue various interests
  - To protect their citizens
  - To protect the economy
  - To protect themselves
- May take the shape of regulations and policy enforcement
- May follow national or international (e.g., EU) rules
- Regulation sets the stage for technology deployment
  - Pre-scribes non-functional requirements
  - Adds functional requirements
- Uses technology to achieve its goals

# Regulation Example: (IP) Telephony (1)

- Many countries guarantee privacy rights to their inhabitants
  - Example: Privacy of telephony and (postal) mail
  - Protocol world: perform (strong) encryption
- but at the same time reserve the right for making exceptions
  - Example: Eavesdropping, collecting call history of users
  - System world: counter encryption, demand eavesdropping systems, keys, …
    - Demands and requirements are not always clear about practical implication
- Another example: anonymous calling
  - Allow hiding the caller’s identity
- Exception: perform malicious call tracing and accountability
  - Ensure that the caller’s identity can be determined by the authorities later on
- Applicable beyond telephony
  - Tracking actions of Internet users: for web access, peer-to-peer usage, etc.

# Regulation Example: (IP) Telephony (2)

- Adding functional requirements to a protocol or system
  - Which may lead to “more expensive” protocol design and operation
- Example: Emergency calling
- Comprehensive requirements from traditional landline service
  - Locating the emergency caller
    - Has been somehow easy when using fixed landlines
  - Routing the call to the closest “Public Safety Answering Point” (PSAP)
- Implications for IP-based technologies
  - Need to provide location information about IP phones
    - Despite the ability of the user to move
  - Need to identify a call as an emergency call
    - Regardless where the user is
  - Obey privacy rules for highly sensitive location information

# The Grey and Dark Sides: Blocking Access

- Basically legitimate goals
  - Parental control of Internet usage
  - ISP control of users
    - Block spammers
    - Sources of DoS attacks, viruses
  - Governmental control
    - Restrict access to legally prohibited contents (e.g., anti-constitutional, subversive)
    - But also: limit freedom of information

**Net Neutrality?!**

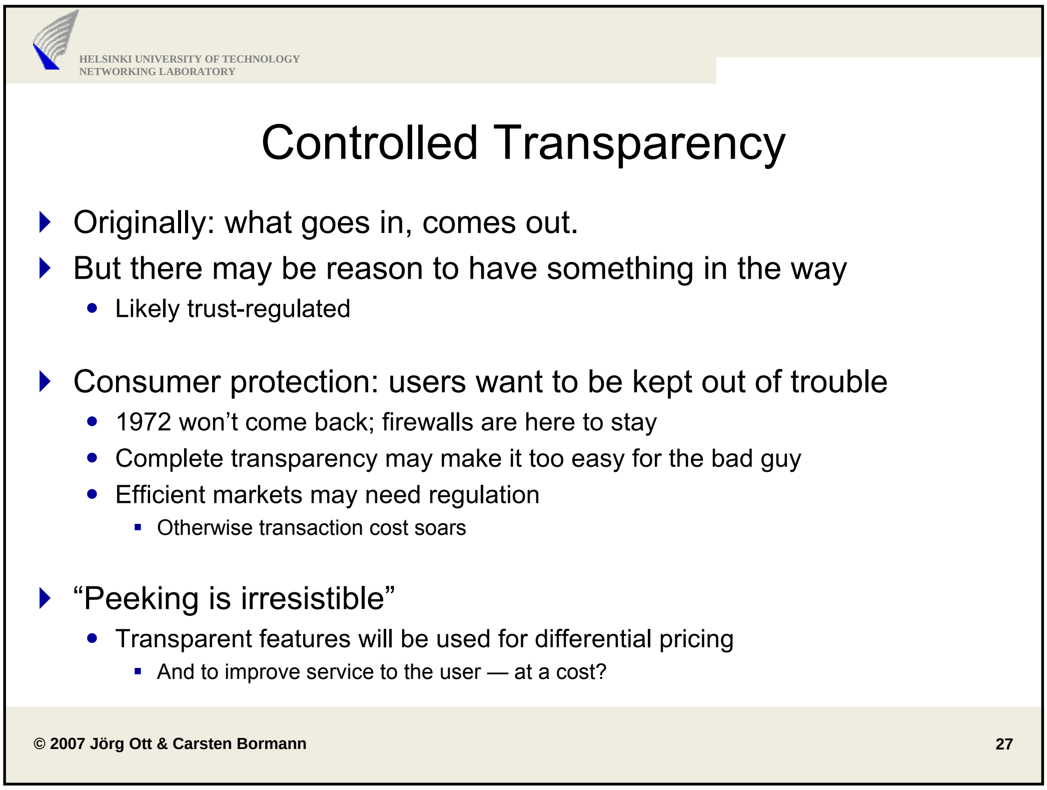
- May succeed somehow easily with the masses
  - But may also have quite a few “false positives” beyond intentions
- But: potential for yet another technology race for the bad guys
  - There are usually technical ways around

# The Spam Tussle (1)

- Problem: Internet lowers transaction cost considerably
  - Anyone can send messages to many at near zero cost
  - There **is** a (human) cost for consuming a message, though
- Conflict: How to stay open?
  - Do I want to accept messages from unknown sources?
  - “Known-sources only” becomes limiting quickly
- Technological response:
  - Spam filters try to detect “unsolicited bulk” messages
  - Arms race, limited success (spammers are hard to trace, use botnets)
- Economical response:
  - Re-introduce “cost” for a message
  - Might be waived for messages that actually were “wanted”
  - Issue: How to design for choice?

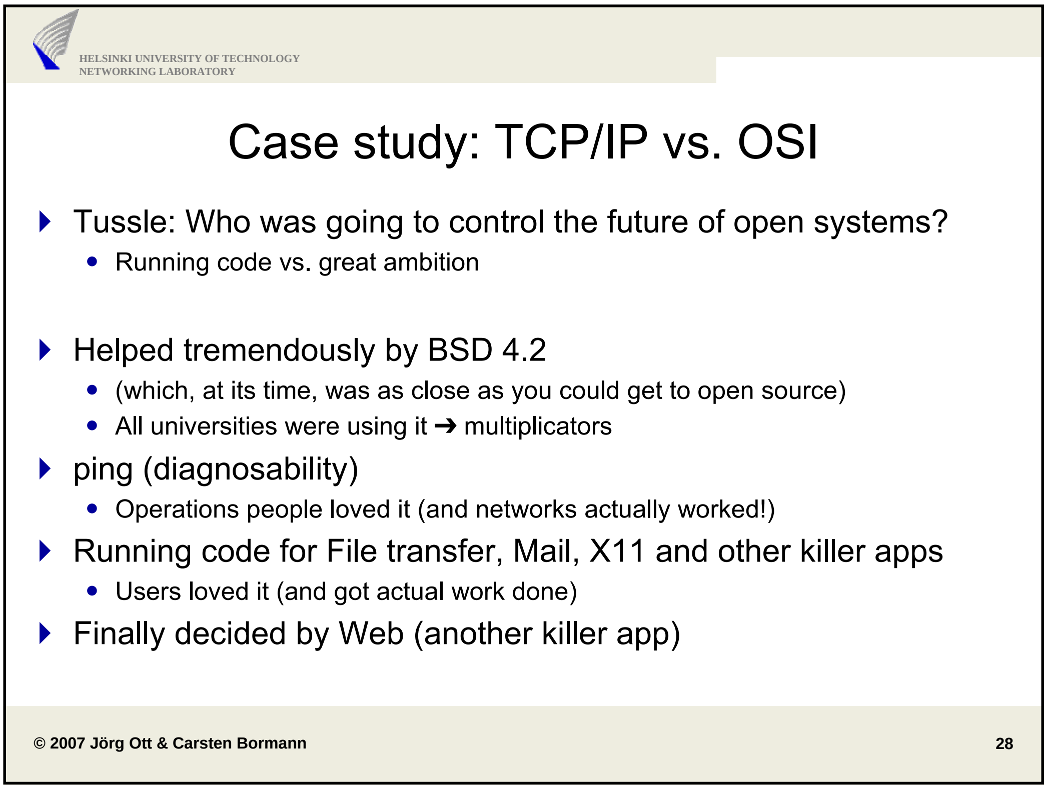
# The Spam Tussle (2)

- Nominally, everyone is “against spam”
  - This is not about protocol features shot down because they “would hurt spam”
  - (But you don’t want to have protocol features that actually would help spam)
- The part of the tussle relevant to protocol design:
  Business opportunities from spam
  - More precisely: from the extreme pain point spam now causes in business
- Use Spam to reign in control lost 10 years ago
  - Use market power to establish patented system as de-facto spam reduction standard
- Establish a service for centralized spam checking
  - Compete by protocol support in dominant implementations
- Provide a Mail service with better spam control than others
  - Real competition!



# Controlled Transparency

- Originally: what goes in, comes out.
- But there may be reason to have something in the way
  - Likely trust-regulated
- Consumer protection: users want to be kept out of trouble
  - 1972 won't come back; firewalls are here to stay
  - Complete transparency may make it too easy for the bad guy
  - Efficient markets may need regulation
    - Otherwise transaction cost soars
- "Peeking is irresistible"
  - Transparent features will be used for differential pricing
    - And to improve service to the user — at a cost?



# Case study: TCP/IP vs. OSI

- Tussle: Who was going to control the future of open systems?
  - Running code vs. great ambition
- Helped tremendously by BSD 4.2
  - (which, at its time, was as close as you could get to open source)
  - All universities were using it ➔ multiplicators
- ping (diagnosability)
  - Operations people loved it (and networks actually worked!)
- Running code for File transfer, Mail, X11 and other killer apps
  - Users loved it (and got actual work done)
- Finally decided by Web (another killer app)

# Case study: PostScript

- Low barrier to use (text based)
  - easy to “write code” to create beautiful type
  - offloading processing to printer allowed upgrade in functionality
- Extensibility over performance
  - widened applicability and allowed growing with the problem set
- Device independence, scalability
  - Black/white first, later extended to color and other new devices
- Active maintenance, reasonable licensing by Adobe
  - (but still limited pick-up in the low-cost market)
  - good enough to spawn emulation market
- ➔ Became suitable interchange format, too
- but: violates “use the simplest language you can use”

# Case study: PDF

- Used PostScript as a lever

- Using market asymmetry (cheap reader/low cost writer)
- Natural replacement for PostScript as an interchange format...
  - remove programmability
    - By then, problem set had become much more well understood
  - add “modern” formats (images, color spaces, compression, etc.)
  - continued evolution

- Microsoft is trying to replace PDF with Metro

# Case study: SIP

- Incessant marketing by “Godfather of SIP”
- Helped by easy “first mile” of text-based, HTTP-like protocol
  - in particular after the H.323 portrayed complexity and PER disaster
  - plus H.323’s “closed group + expensive consultants” image, late open source
- However, damaged in mass market by
  - NAT problems
  - moving target syndrome
  - Configuration complexity (odyssey of a simple client configuration format)
  - dearth of good soft clients
- Does not have a good answer to the “federation problem”
  - May be eclipsed by Jabber/Jingle in certain applications

# Case study: Skype

- Tussle: get new application **VoIP** going despite restrictive firewalls
  - Phone calls at zero incremental cost (beyond broadband already available)
- Usable, polished client (including IM and Video)
  - solves NAT problem
- Low barrier to entry for new users
  - Early adopters: download, try, works — recommend!
  - Metcalfe’s law kicked in soon
- High end user benefit
  - including high connection quality (wideband)
- (Unfortunately, Skype is fundamentally flawed — and not open in the first place)

# Case study: Jabber

- Tussle: whose IM systems will dominate? (AIM, MSN, …)
  - libgaim
- Jabber (XMPP): the standardized protocol in the IM space
  - Well, there are IRC, SIMPLE, …
  - Low-barrier design
- Has a successful federation policy
  - Design for choice
  - (and the other guy is unlikely to be a spammer)
- Once that works, why not use it in place of SIP?
  - google talk, Jingle
- ...we are in the middle of the telephony tussle…

# Case study: RSS

- “Push” did not quite work because of the firewall/NAT problem
- Idea: Provide “push” by repeated “pull”
  - Browser needs to find out if information is “new”
- RSS: Rich site summary/Really simple syndication
  - “Feed” metadata: Title + Link + Updated + Author
  - Array of “Entry” metadata: Title + Link + **Id** + Updated + Summary [+ Content]
- Use XML format
- Problem: Tag Soup effect; multiple RSS versions
- Solution: IETF process ➔ Atom (RFC 4287)
  - Atom is quickly becoming the “Enterprise Message Bus” of the Internet

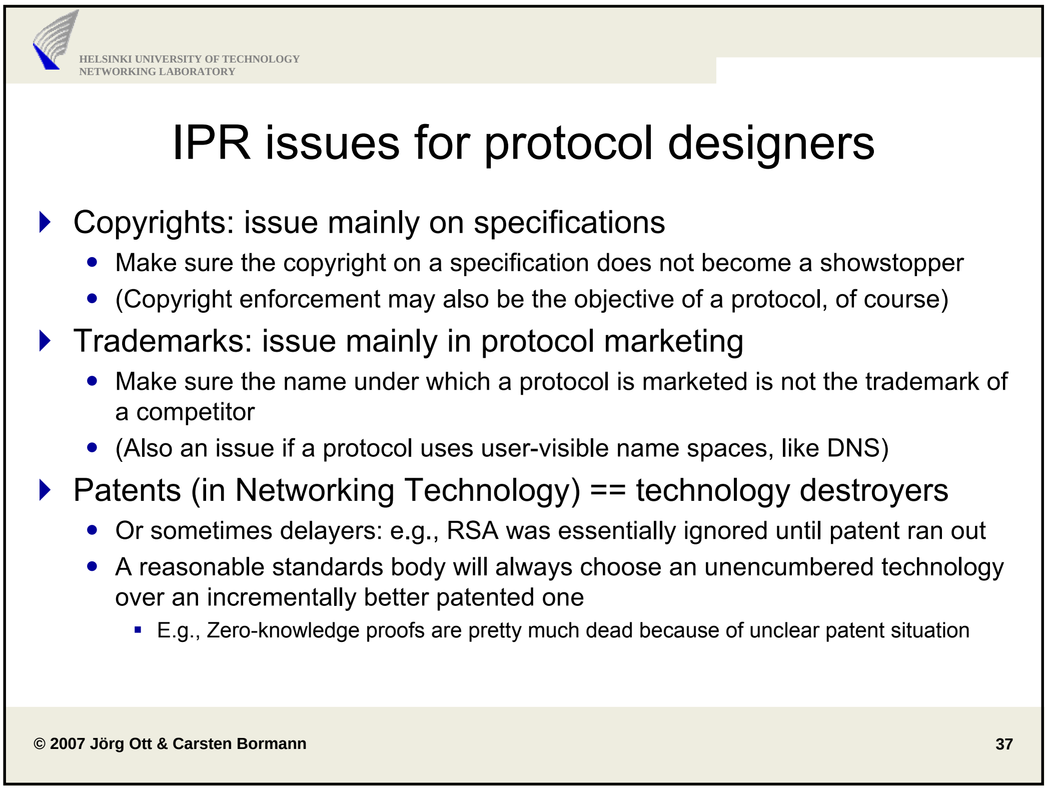
# Case study: DVD-successor

- Tussle 1: Copyright holders against the rest of the world
  - Threaten not to provide pre-recorded HD content unless DRM is draconian
  - Need to control entire **system**
- Tussle 2: Two patent pools fighting each other
  - Indecision between HD-DVD and Blu-Ray
  - Microsoft changing sides every week
- Result:
  - Delayed market introduction (Tussle 1)
  - Immense market confusion (Tussle 2), “wait and see” attitude
- Tussle 1 also makes it less likely that consumers will actually want the “advances” of the DVD-successor
- Interesting development to follow

# Loose ends: Protection Rights (“IPR”)

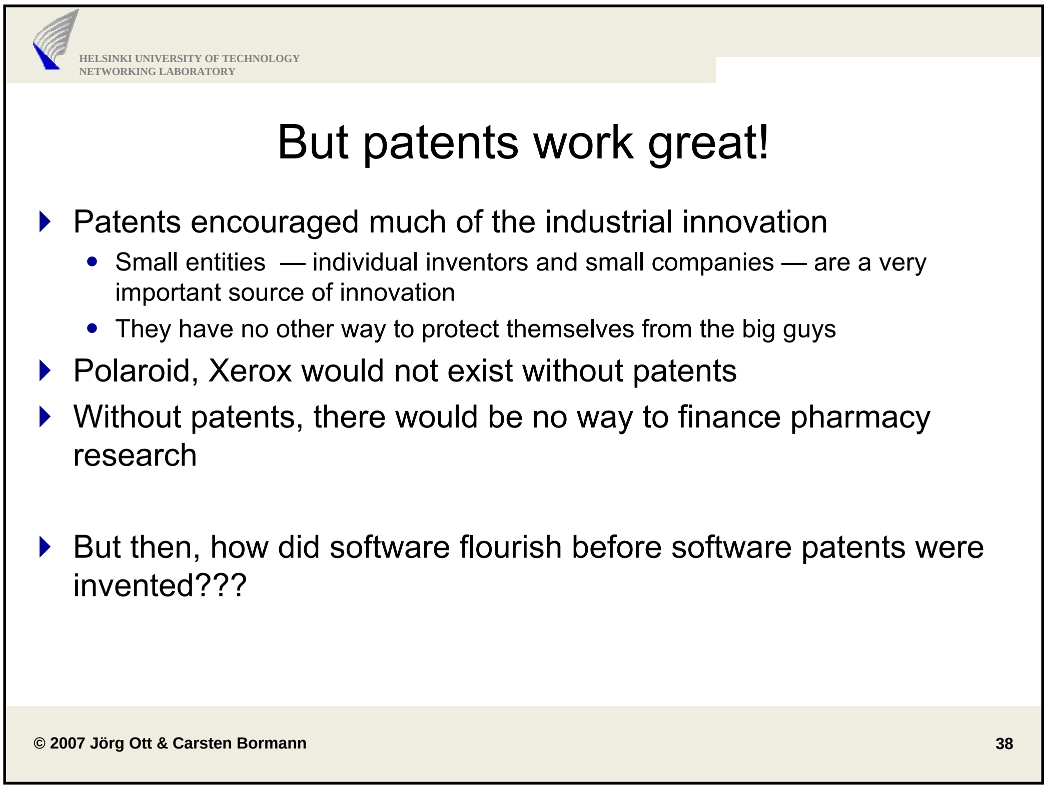
There are several kinds of “protection rights”

- Copyright: protects a work (book, program) against copying
  - Still the basis for the most important revenue models of the information economy
  - A reform is probably inevitable, but might take a couple more decades
- Trademark: protects the branding of a product (“Coca-Cola”)
  - Essentially irreplaceable from a consumers’ rights point of view
  - Somewhat unfortunate side-effects on DNS name space
- Patent: protects ideas, even if they are reinvented
  - Designed for 19th century industrial economy



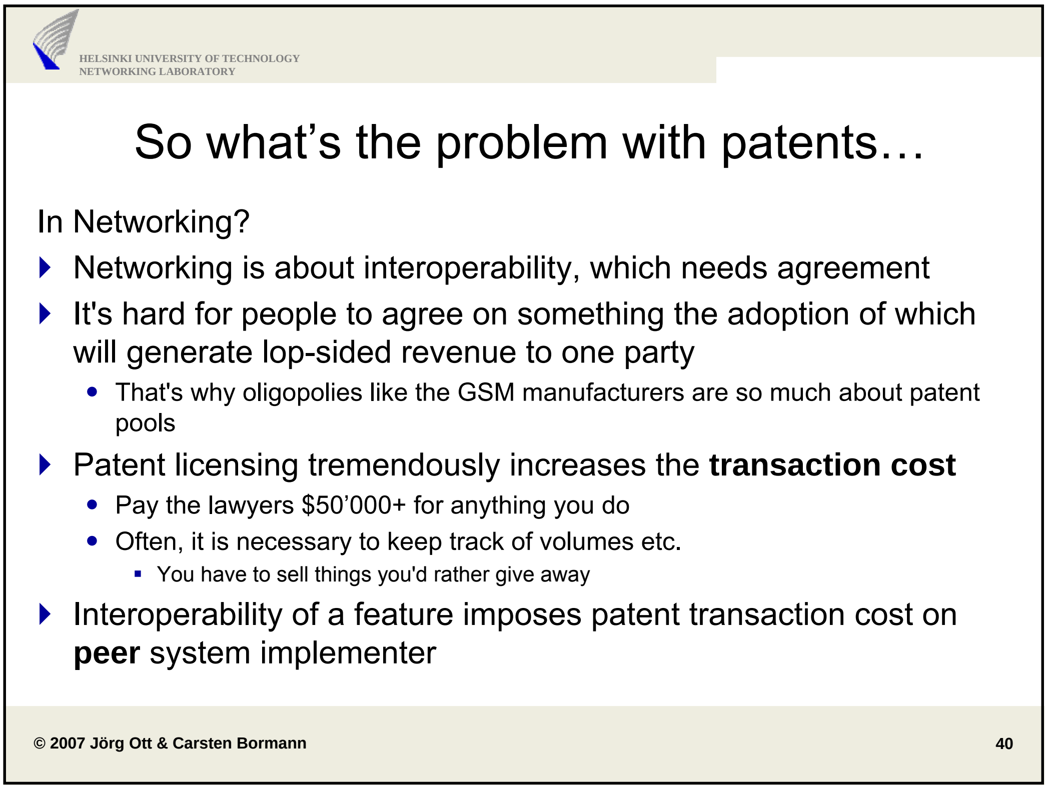
# IPR issues for protocol designers

- Copyrights: issue mainly on specifications
  - Make sure the copyright on a specification does not become a showstopper
  - (Copyright enforcement may also be the objective of a protocol, of course)
- Trademarks: issue mainly in protocol marketing
  - Make sure the name under which a protocol is marketed is not the trademark of a competitor
  - (Also an issue if a protocol uses user-visible name spaces, like DNS)
- Patents (in Networking Technology) == technology destroyers
  - Or sometimes delayers: e.g., RSA was essentially ignored until patent ran out
  - A reasonable standards body will always choose an unencumbered technology over an incrementally better patented one
    - E.g., Zero-knowledge proofs are pretty much dead because of unclear patent situation



# But patents work great!

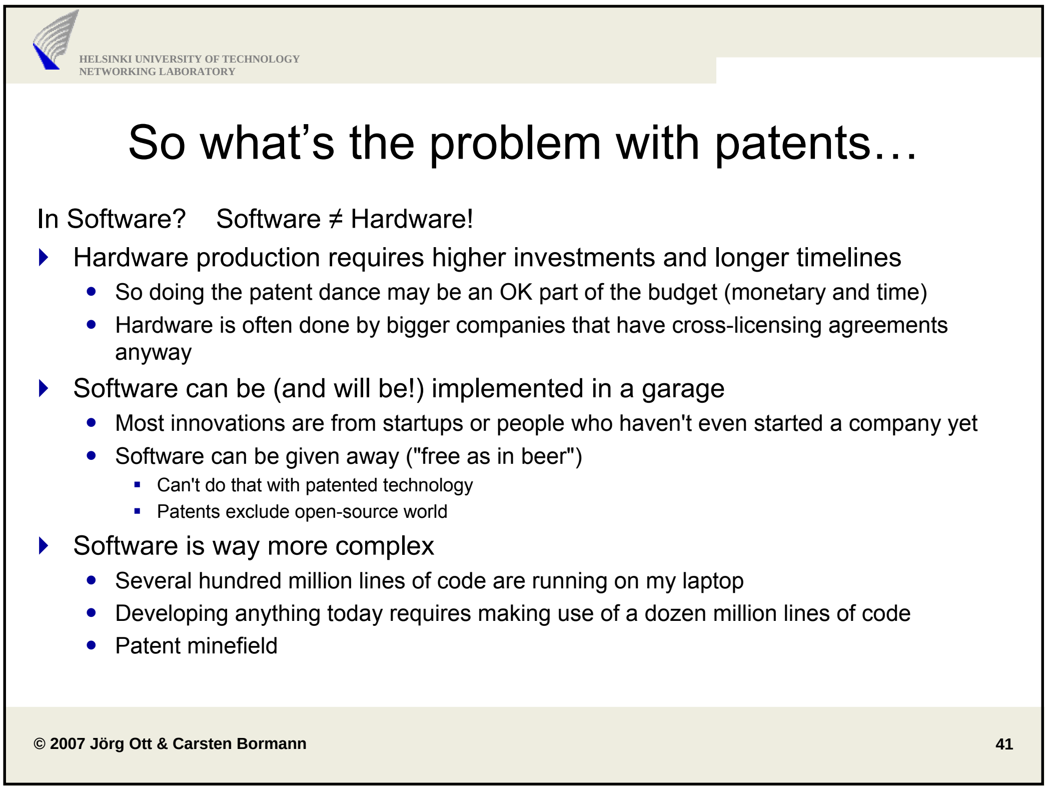
- Patents encouraged much of the industrial innovation
  - Small entities — individual inventors and small companies — are a very important source of innovation
  - They have no other way to protect themselves from the big guys
- Polaroid, Xerox would not exist without patents
- Without patents, there would be no way to finance pharmacy research
- But then, how did software flourish before software patents were invented???



## So what’s the problem with patents…

In Networking?

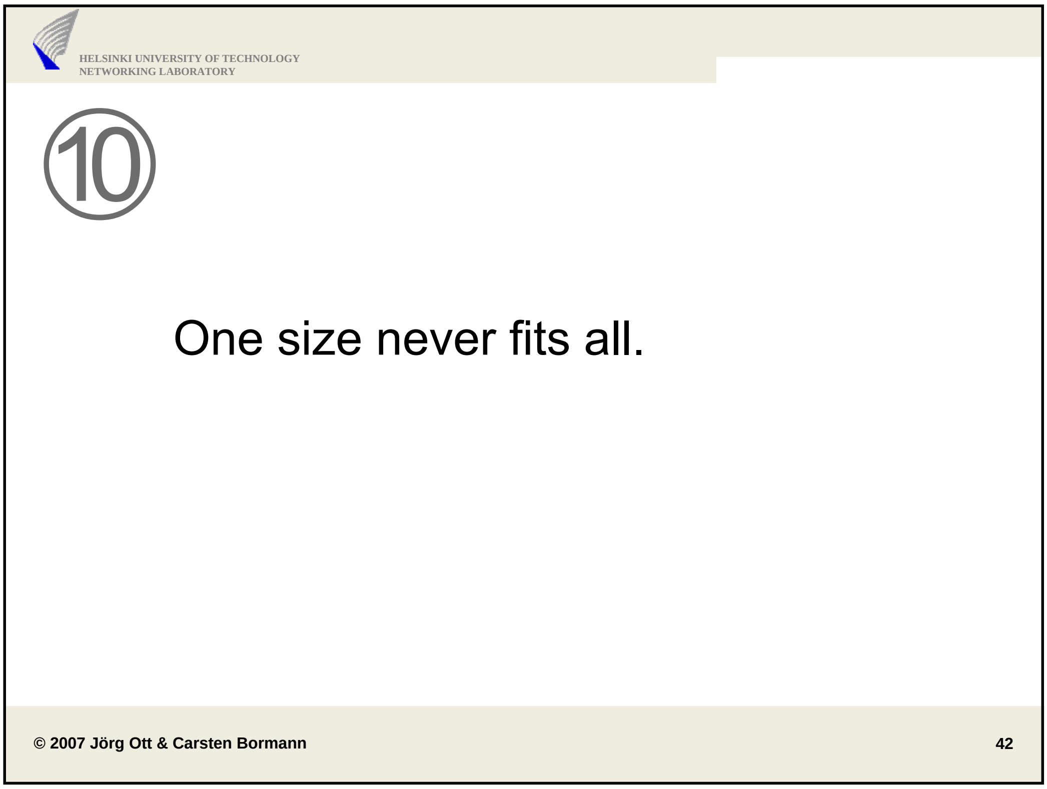
- Networking is about interoperability, which needs agreement
- It's hard for people to agree on something the adoption of which will generate lop-sided revenue to one party
  - That's why oligopolies like the GSM manufacturers are so much about patent pools
- Patent licensing tremendously increases the **transaction cost**
  - Pay the lawyers $50’000+ for anything you do
  - Often, it is necessary to keep track of volumes etc.
    - You have to sell things you'd rather give away
- Interoperability of a feature imposes patent transaction cost on **peer** system implementer



# So what’s the problem with patents…

In Software? Software ≠ Hardware!

- Hardware production requires higher investments and longer timelines
  - So doing the patent dance may be an OK part of the budget (monetary and time)
  - Hardware is often done by bigger companies that have cross-licensing agreements anyway
- Software can be (and will be!) implemented in a garage
  - Most innovations are from startups or people who haven't even started a company yet
  - Software can be given away ("free as in beer")
    - Can't do that with patented technology
    - Patents exclude open-source world
- Software is way more complex
  - Several hundred million lines of code are running on my laptop
  - Developing anything today requires making use of a dozen million lines of code
  - Patent minefield



⑩

One size never fits all.

# Defects in the patent system (1)

- It is relatively easy to obtain a patent (tens of thousand Euros)
  - Very limited expertise on the part of the patent examiners
  - Patents are essentially checked only against earlier patents
  - The “inventor” (applicant) has control over the process
  - Most patents are “trivial patents”
- Patent applications stay a secret for 18 months (or until granted)
  - Submarine patents
  - Even published patents become submarines by novel re-interpretation
- “Prior Art” arguments need to be fought in court
  - In theory, they can be fought in the objection phase after granting
  - But: This gives “inventor” too much control over the process
    - Documents “used up” here are hard to reuse in court

# Defects in the patent system (2)

- Court proceedings:
  - Are obscenely expensive
  - Take a long time
    - during which the technology and the companies using it are branded with a big question mark
  - Are completely unpredictable in their final outcome (≠ logic)
- Challenging a patent is a lopsided exercise
  - Patent holder has high stakes
  - Challenging patent user only has a partial stake in the other side
- Large incentive to “settle”
  - saves court costs
  - gives the “settler” an unfair advantage over its competitors that haven't settled yet
  - might be the more expensive route though, if the patent is finally thrown out
- In the US, patent holder can obtain injunction that essentially stops everything that is using the technology
  - extremely high damage to technology user and its customers
  - absolutely no call for proportionality

# Results of the patent system for networking

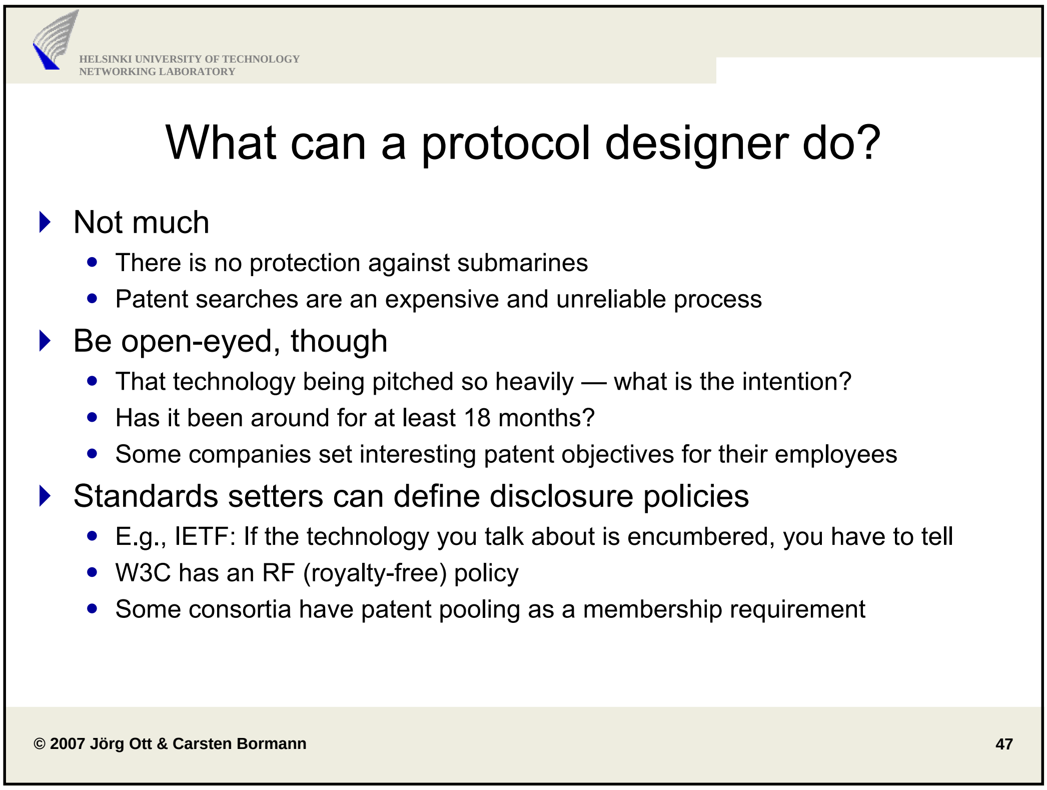
- It is always **unknown** whether a specification is unencumbered
  - in particular, it may be very expensive to say it is
- There is no way to ascertain patent-free status
  - Submarine patents
  - Patents are written in many languages
  - The language of patents is often unrelated to that of technology
    - Or that of humans ("a plurality of...")
- Civilization is about controlling risks
  - Software patents are the anathema of civilization
  - “Technology companies” == wayside robbers
  - Damage to economy (chilling effects) far outweighs proceeds to individuals

# So why are the big guys arguing for software patents?

Battle being fought in Europe right now

- US already have software patents
  - Big companies need to pay the cost there to stay in the game (protection from other patents)
- Big companies can benefit from their US investment
  - Can use patents to squash smaller European innovators

- Another reason: The corporate position on patents is usually defined by ——— the patent department!
  - What do you think would they say?



# What can a protocol designer do?

- Not much
  - There is no protection against submarines
  - Patent searches are an expensive and unreliable process
- Be open-eyed, though
  - That technology being pitched so heavily — what is the intention?
  - Has it been around for at least 18 months?
  - Some companies set interesting patent objectives for their employees
- Standards setters can define disclosure policies
  - E.g., IETF: If the technology you talk about is encumbered, you have to tell
  - W3C has an RF (royalty-free) policy
  - Some consortia have patent pooling as a membership requirement