Internet 2.0 – Challenges for the Future Internet

Henning Schulzrinne
(with Victoria Beltran, Omer Boyaci, Jae Woo Lee, Arezu Moghadam, Suman Srinivasan and others)
Dept. of Computer Science
Columbia University
New York, NY

Tampere 2010

Overview

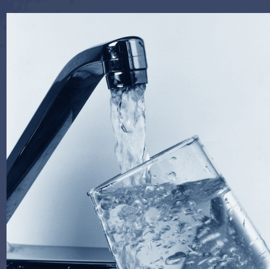
* The Internet as core civilizational infrastructure
* Challenges
  * Network address exhaustion
  * Routing table explosion
  * Network ossification
  * Securing the network infrastructure
  * Usability & towards self-managed networks
* Opportunistic networks
* Future Internet – MIA (Minimal Internet Architecture)

IP as a core civil (izational) infrastructure interface

The great infrastructures

* Technical structures that support a society → "civil infrastructure"
  * Large
  * Constructed over generations
  * Not often replaced as a whole system
  * Continual refurbishment of components
  * Interdependent components **with well-defined interfaces**
  * High initial cost

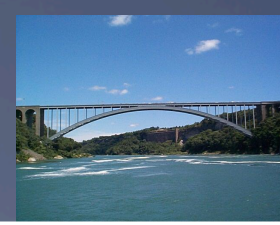
water



Ta

energy

transportation



The Internet as core civil infrastructure

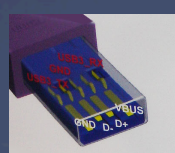
- Involved in all information exchange
  - (in a few years)
- Crucial to
  - commerce
  - governance
  - coordination
  - inter-personal communication
- Assumed to just be there
  - "plumbing", "pipes", …

Interfaces: Energy

110/220V

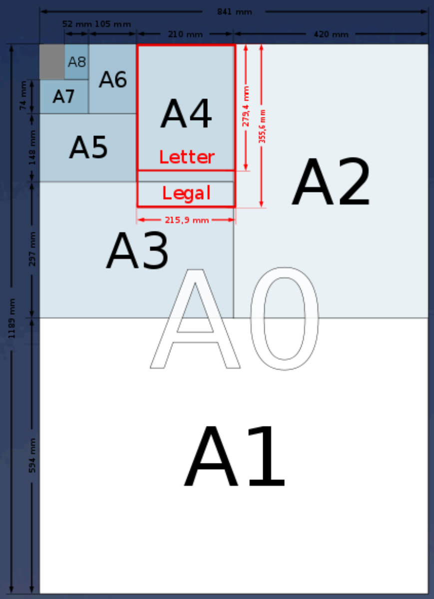
1904

- Lots of other (niche) interfaces
- Replaced in a few applications

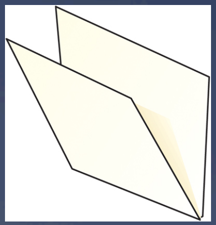


1901

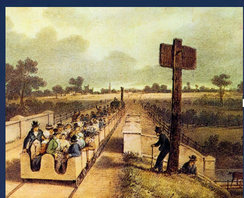
Interfaces: Paper-based information



1798, 1922 (DIN)



Interfaces: transportation



1435 mm

1830 (Stephenson)
1846 UK Gauge Act

About 60% of world railroad mileage

12'

What makes interfaces permanent?

- Widely distributed, uncoordinated participants
- Capital-intensive
  - depreciated over 5+ years
  - see Y2K problem
- Allocation of cost vs. savings
  - e.g., ISP saves money, end user pays
- Hard to have multiple at once
  - “natural monopoly”

Extrapolating from history

- IP now *the* data interface
- Unclear that any packet-based system can be
  - ≥ 10 times cheaper
  - ≥ 10 times more functionality
  - ≥ 10 times more secure
- Replacing phone system due to generality, not performance
  - IP offers general channel
- → We're stuck with IPv4/IPv6
  - except for niche applications (car networks, BlueTooth, USB, …)

Technology evolution

* Early technology stages:
  * make it work
  * make it cheap
  * make it fashionable
  * This happened in the auto industry. Early cars barely worked at all, every journey was an adventure. In the 1920s Ford broke the automobile patent and built a car for the common man, a car that did not need the skills of a mechanic to drive. Reliability improved gradually until the 1970s when there was a sudden realization that consumers would pay more for a car that was not designed to rust. Today most cars will go 10,000 miles between services and not need major repairs beyond a clutch plate for 50,000 or even 100,000 miles
* Completion of conversion from analog to digital/ packet media
* Patterson: **S**ecurity, **P**rivacy, **U**sability, **R**eliability
  * phishing attacks, DDOS
  * cost of purchase vs. cost of ownership
  * dependability (crashes & reboots)

What defines the Internet?

Basic IP service model

- Unchanged since 1978
  - Send without signaling
  - Receive at provisioned address, without signaling
    - but: permission-based sending
  - Variable-sized packets < ≈ 1,500 bytes
  - Packets may be lost, duplicated, re-ordered

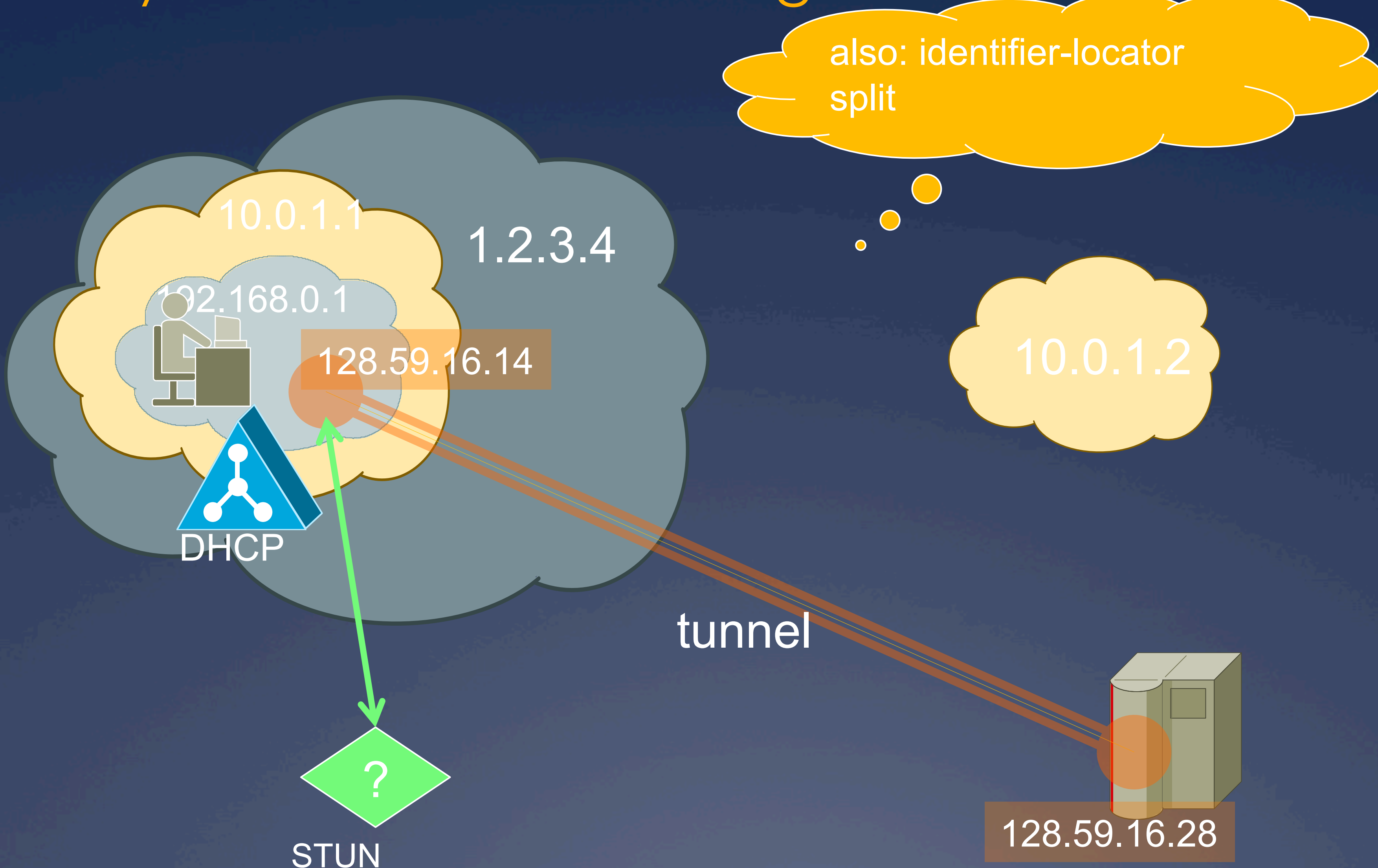
More than just Internet Classic

<table>
<tr><th>Network</th><th>wireless</th><th>mobility</th><th>path stability</th><th>data units</th></tr>
<tr><td>Internet “classic”</td><td>last hop</td><td>end systems</td><td>> hours</td><td rowspan="4">IP datagrams</td></tr>
<tr><td>mesh networks</td><td>all links</td><td>end systems</td><td>> hours</td></tr>
<tr><td>mobile ad-hoc</td><td>all links</td><td>all nodes, random</td><td>minutes</td></tr>
<tr><td>opportunistic</td><td>typical</td><td>single node</td><td>≈ minute</td></tr>
<tr><td>delay-tolerant</td><td>all links</td><td>some predictable</td><td>some predictable</td><td>bundles</td></tr>
<tr><td>store-carry-forward</td><td>all nodes</td><td>all nodes</td><td>no path</td><td>application data units</td></tr>
</table>

Addressing assumptions

* A host has only one address & one interface
  * apps resolve name and use first one returned
  * address used to identify users and machines
  * machine-wide DHCP options
* Failing
  * multi-homing on hosts (WiFi + Ethernet + BlueTooth + 3G)
* Attempts to restore
  * MIP: attachment-independent address
  * HIP: cryptographic host identify

Myth #1: Addresses are global & constant



Myth #2: Connectivity commutes, associates

* Referals, call-backs, redirects
* Assumptions:
  * A connects to B → B can connect to A
  * A connects to B, B to C → C can connect to A
* May be time-dependent

This is not your text book's Internet any more…

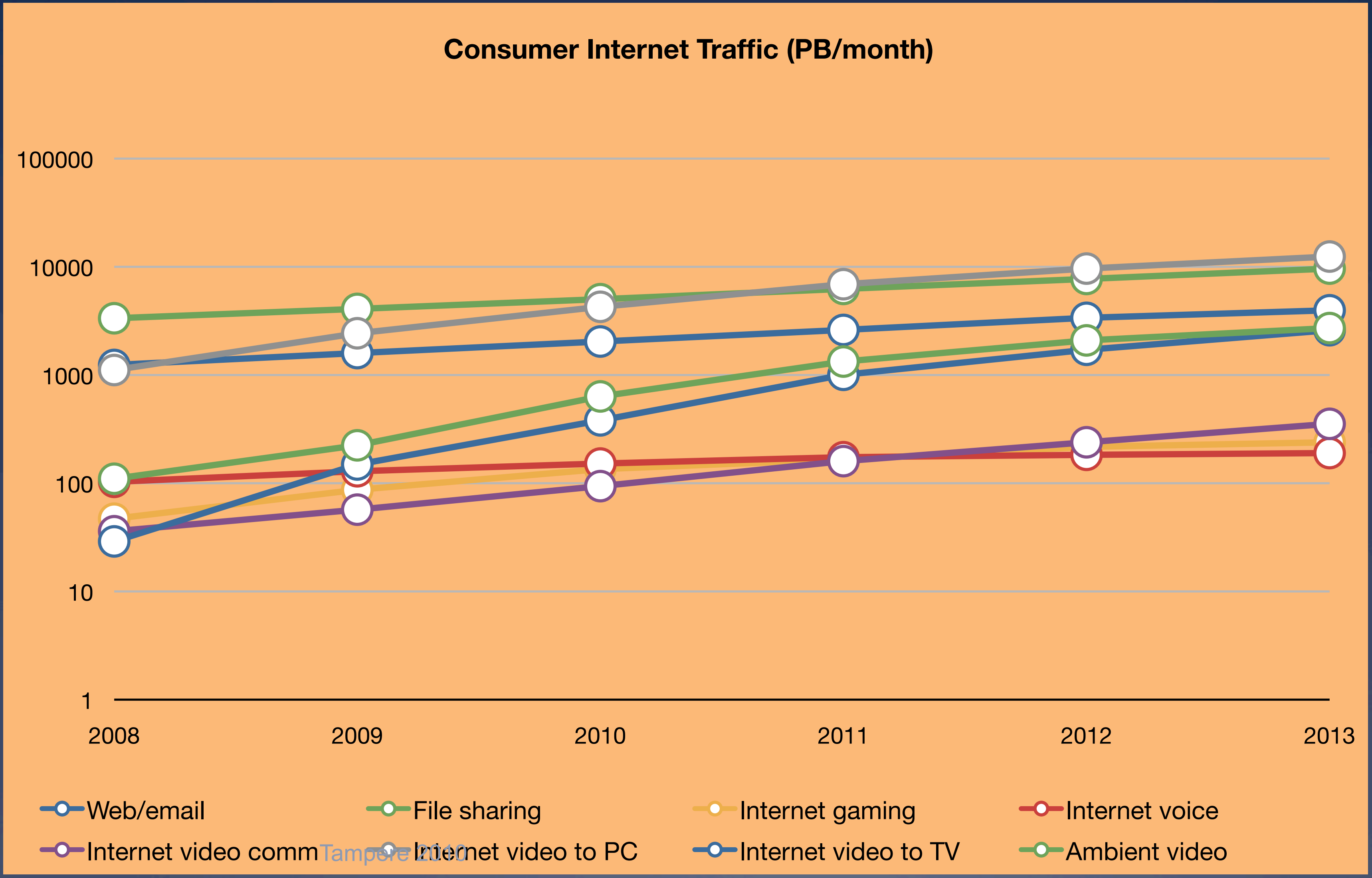
Cisco's traffic prediction

Table 3. Global Consumer Internet Traffic, 2008–2013

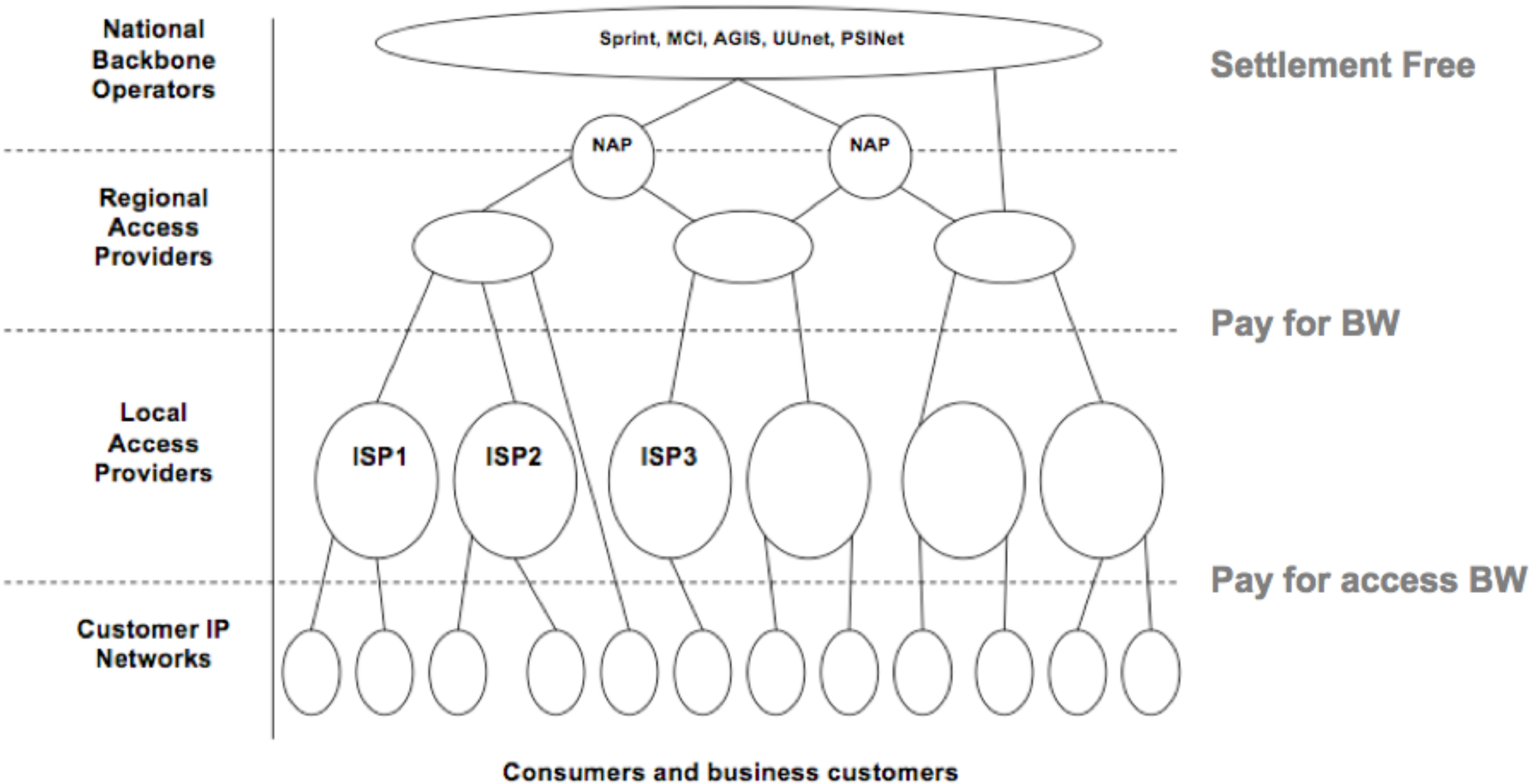
Consumer Internet Traffic, 2008–2013						
	2008	**2009**	**2010**	**2011**	**2012**	**2013**
By Sub-Segment (PB per month)						
Web/Email	1,239	1,595	2,040	2,610	3,377	3,965
File Sharing	3,345	4,083	5,022	6,248	7,722	9,629
Internet Gaming	47	87	135	166	217	239
Internet Voice	103	129	152	174	183	190
Internet Video Communications	36	57	94	160	239	354
Internet Video to PC	1,112	2,431	4,268	6,906	9,630	12,442
Internet Video to TV	29	149	381	1,004	1,711	2,594
Ambient Video	110	224	634	1,332	2,089	2,715

Ambient video = nannycams, petcams, home security cams, and other persistent video streams

Cisco traffic prediction

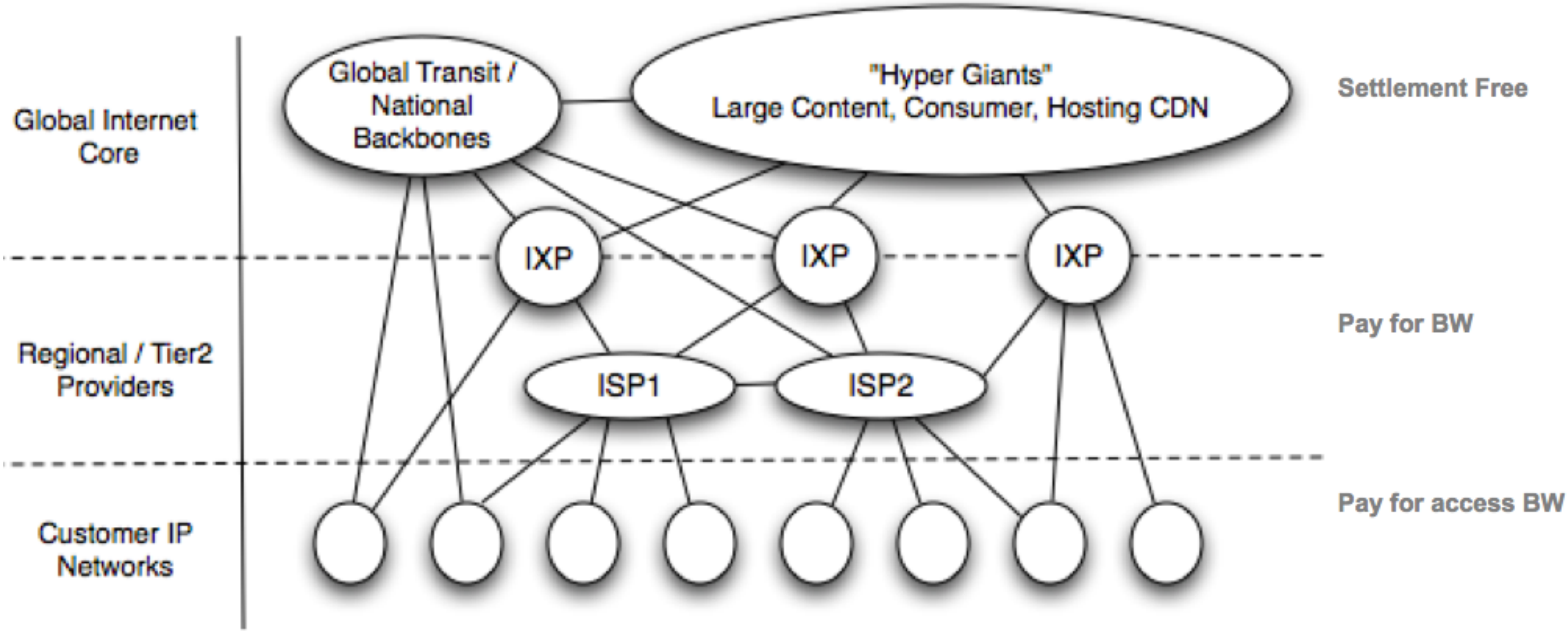


The old Internet



Craig Labovitz, "Internet Traffic and Content Consolidation", IETF March 2010.

A denser Internet



Craig Labovitz, “Internet Traffic and Content Consolidation”, IETF March 2010.

New network providers

Rank	2007 Top Ten	%
1	*ISP A*	5.77
2	*ISP B*	4.55
3	*ISP C*	3.35
4	*ISP D*	3.2
5	*ISP E*	2.77
6	*ISP F*	2.6
7	*ISP G*	2.24
8	*ISP H*	1.82
9	*ISP I*	1.35
10	*ISP J*	1.23

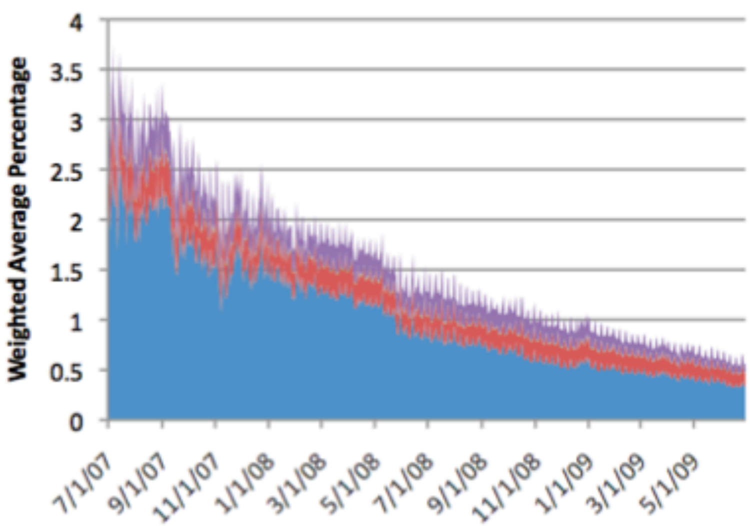
Rank	2009 Top Ten	%
1	*ISP A*	9.41
2	*ISP B*	5.7
3	Google	5.2
4	-	
5	-	
6	Comcast	3.12
7	-	
8	-	
9	-	
10	-	

Based on analysis of anonymous ASN (origin/transit) data (as a weighted average % of all Internet Traffic). Top ten has NO direct relationship to study participation.

Craig Labovitz, “Internet Traffic and Content Consolidation”, IETF March 2010.

P2P declining

P2P



Graph of weighted average traffic using well-known P2P ports

- **In 2006, P2P one of largest threats facing carriers**
  - Significant protocol, engineering and regulatory effort / debate
- **In 2010, P2P fastest declining application group**
  - Trend in both well-known ports and payload based analysis
  - Still significant volumes
  - Slight differences in rate of decline by region (i.e. Asia is slower)

Craig Labovitz, “Internet Traffic and Content Consolidation”, IETF March 2010.

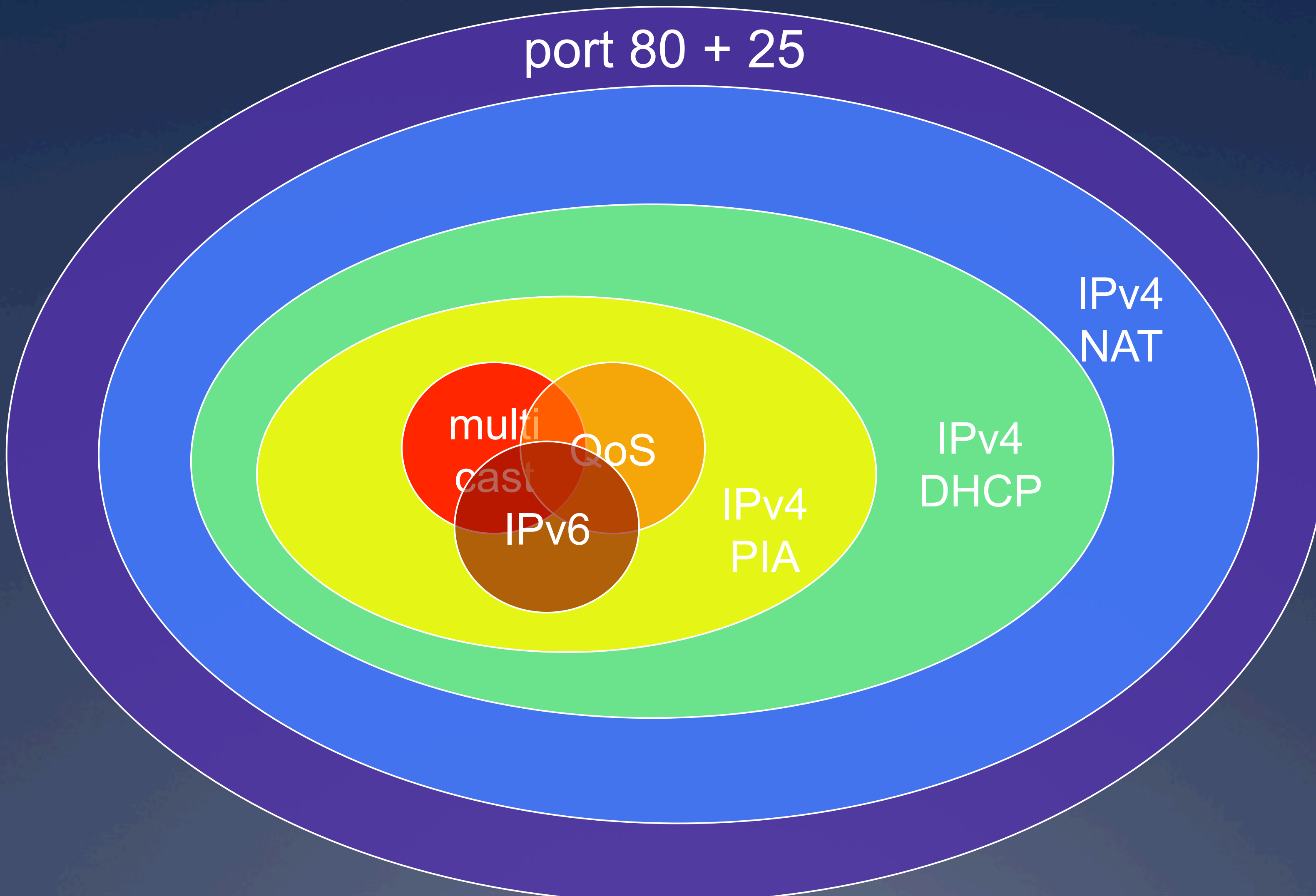
Network ossification

Challenges

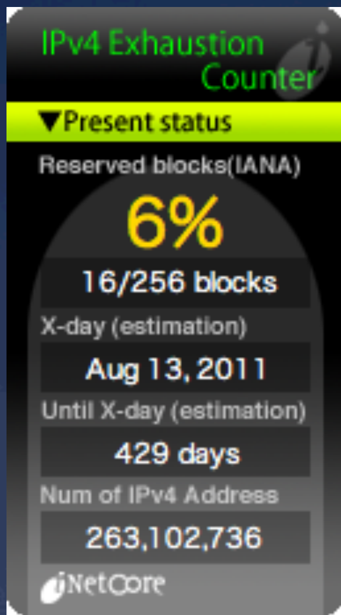
Why is the Internet ossifying?

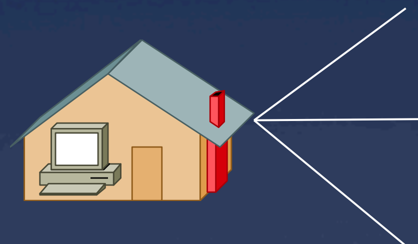
- Lack of network transparency
  - NATs
    - ➔ only UDP + TCP
    - ➔ only client-server
  - Firewalls
    - only HTTP
- Standardization delays
  - No major new application-layer protocol since 1998
  - Protocols routinely take 5+ years
- Deployed base
  - Major OS upgrade every 7-8 years
  - But: automatic software updates
    - encourages proprietary application protocols

Which Internet are you connected to?



Network challenges





multi-homing

routing table explosion

+2 years

+5 years

+8 years

99.9 → 99.999%

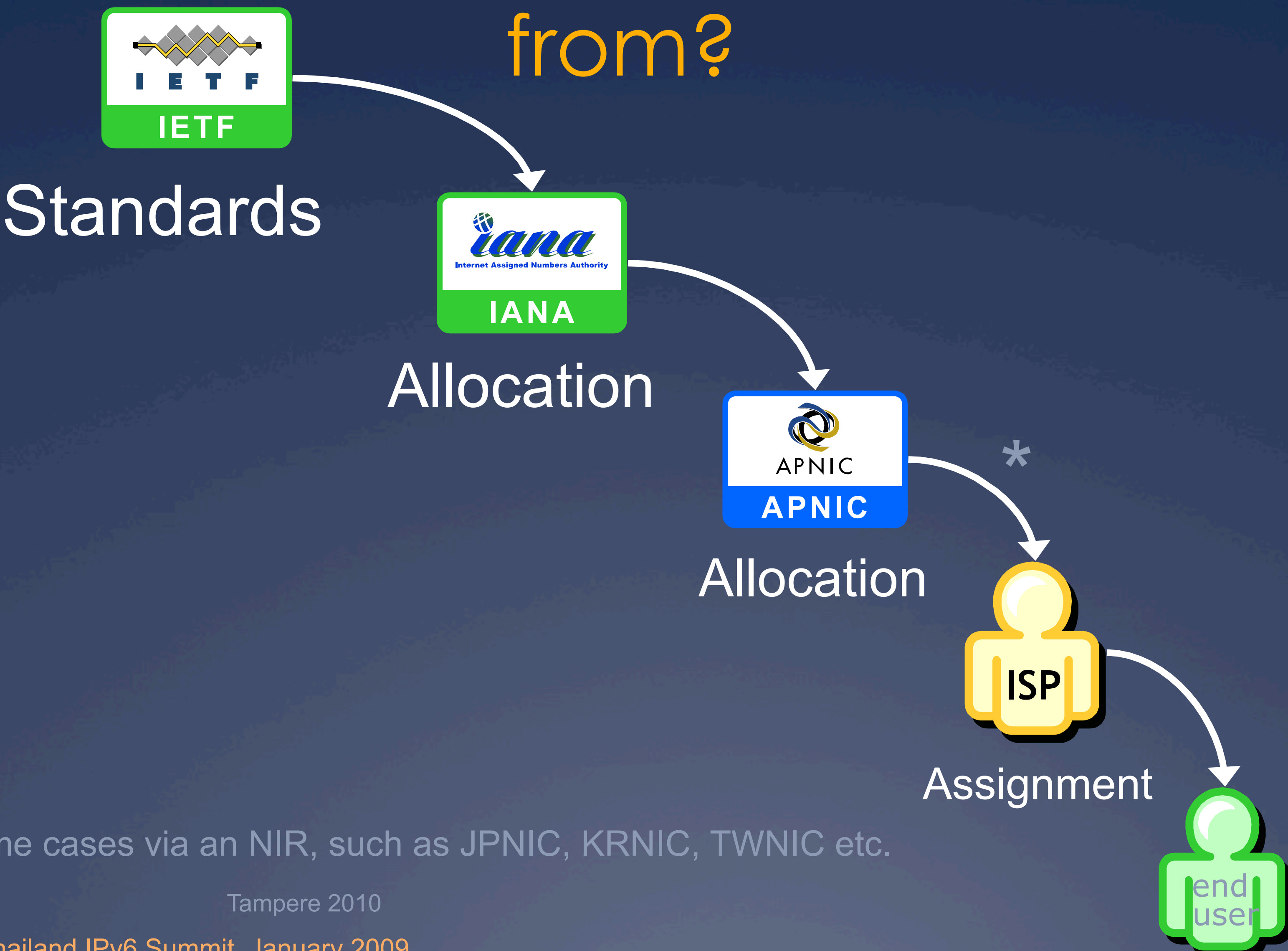
zero configuration

The end of IP(v4) as we know it

Challenges

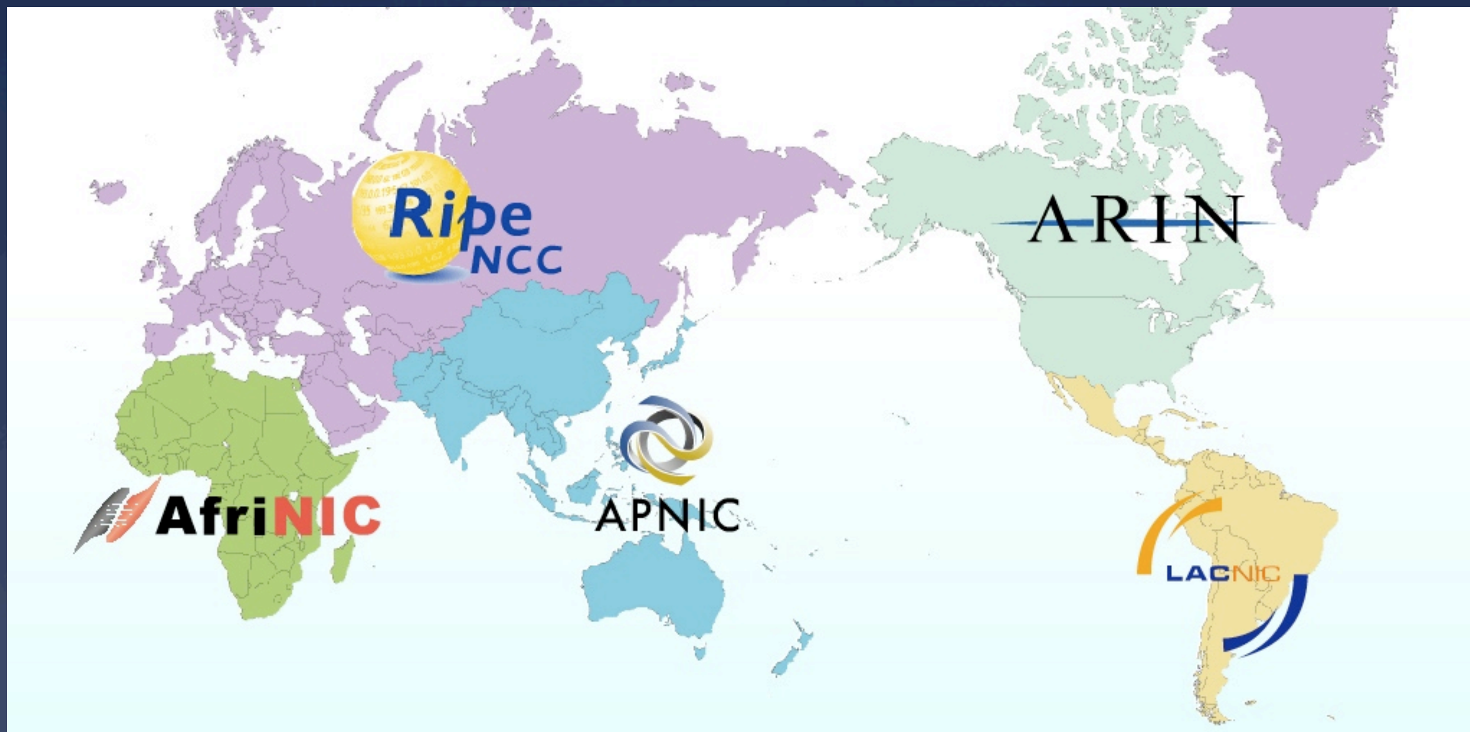
Tampere 2010

Where do IP addresses come from?



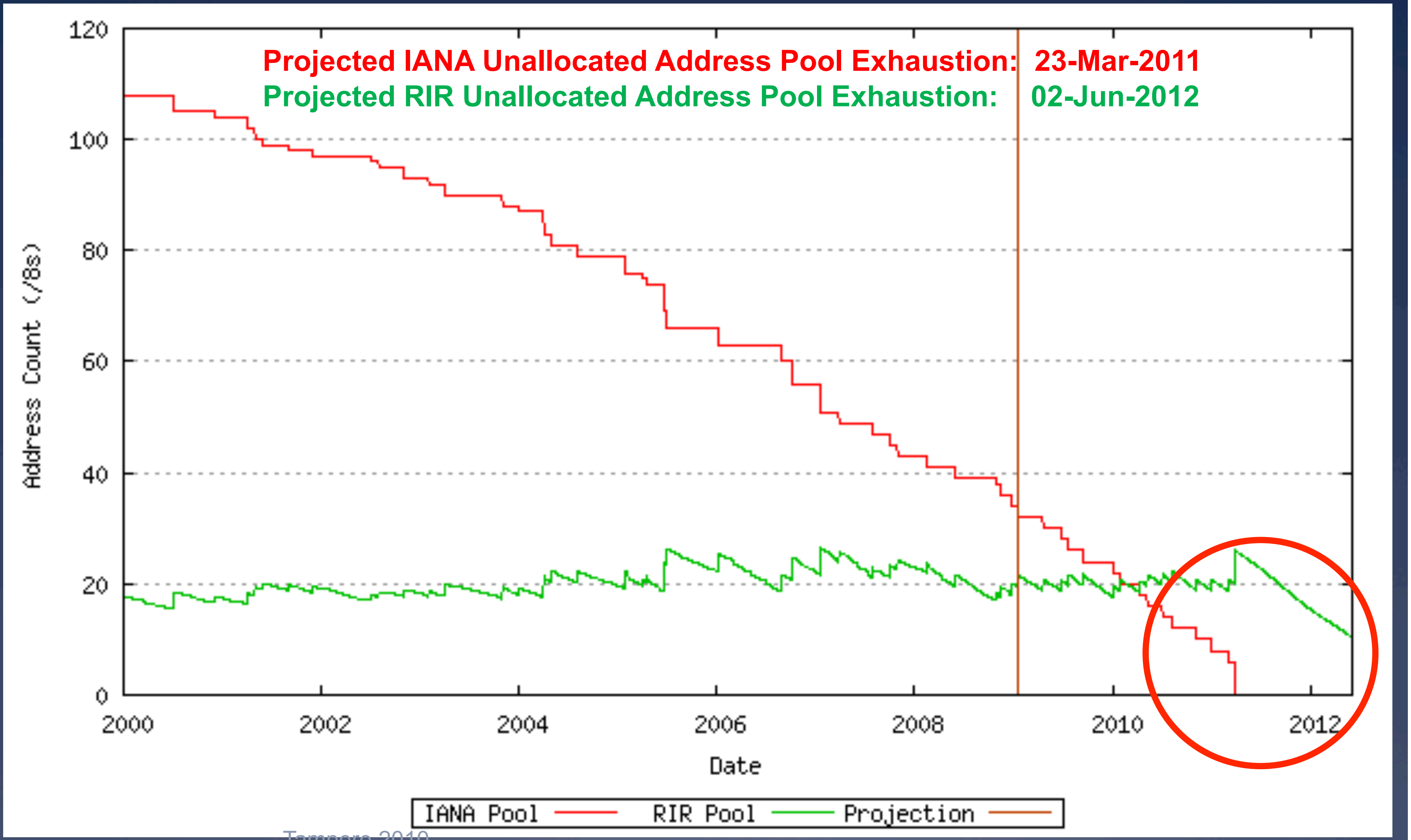
* In some cases via an NIR, such as JPNIC, KRNIC, TWNIC etc.

Regional Internet Registries



The RIRs are responsible for administering the fair distribution and responsible management of IP addresses and AS numbers in their respective regions,

IPv4 consumption – Projection

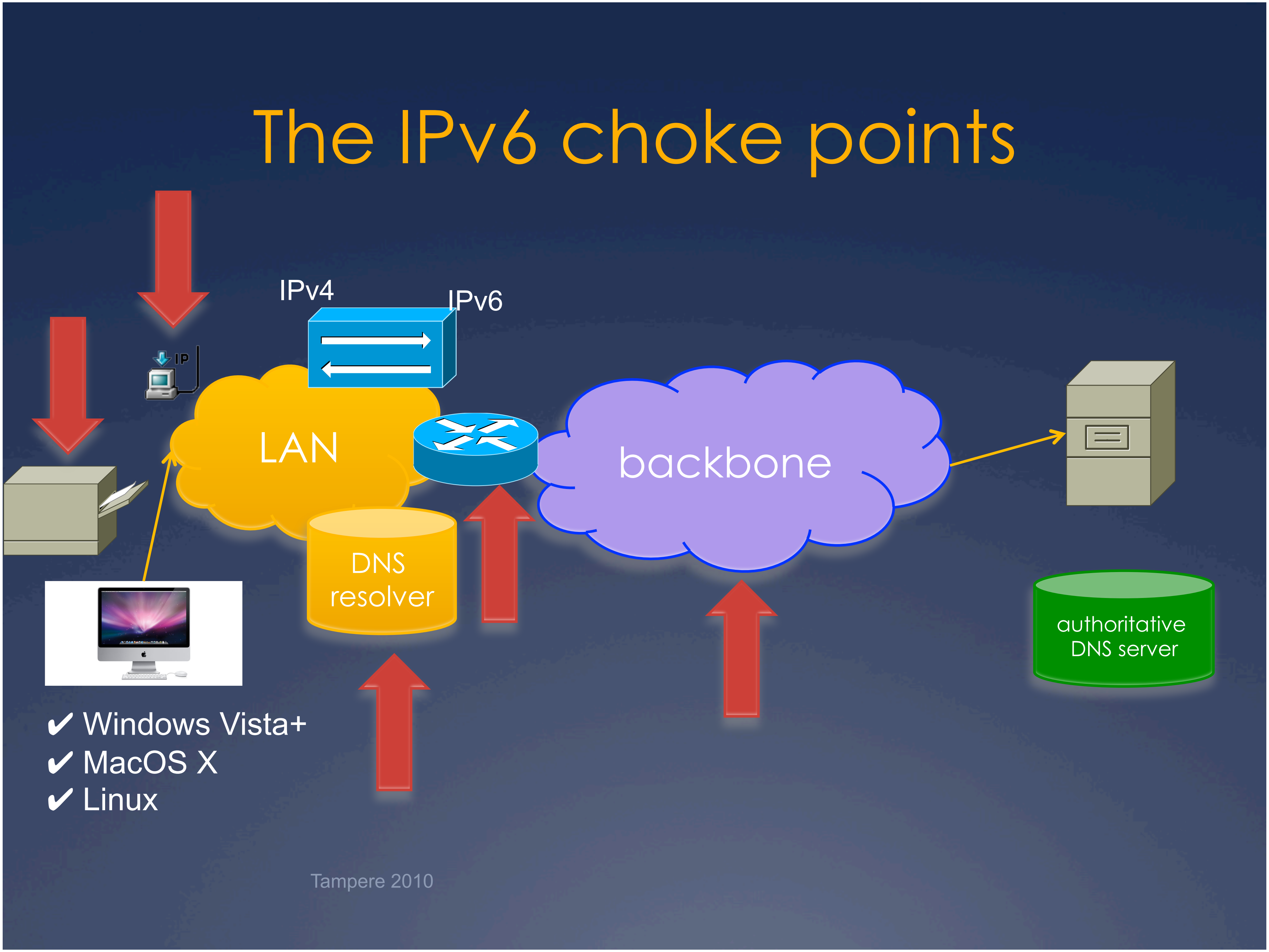


Tampere 2010

Miwa Fujii, Thailand IPv6 Summit, January 2009

The transition to IPv6

* IPv4 needed for at least a decade
* Dual stack transition
  * but IPv6 server + non-IPv6 network + dual-stack server fail annoyingly
* NAT IPv4 ↔ IPv6
  * longer term, RFC 1918 (192.168.*.*) + global IPv6 address
* Decreasing IPv4 address demand
  * multi-layer ("carrier-grade") NATs →
    * limited effectiveness (hundreds of ports for BitTorrent or web page)
    * reliability problems
* Increasing IPv4 address supply
  * recycle unused /8s → few months supply
  * address auctions → router table size ↗

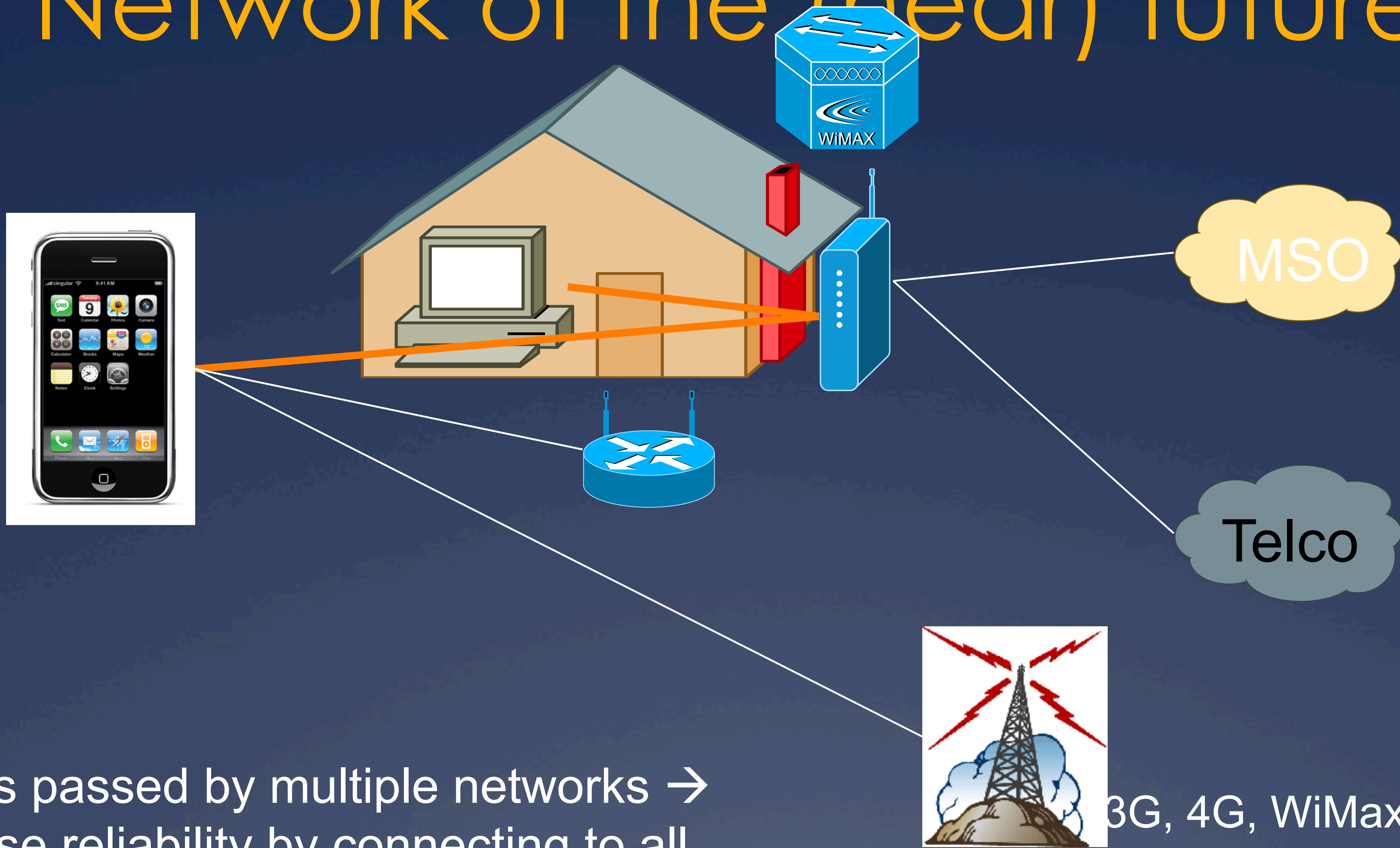

The IPv6 choke points
IPv4
IPv6
IP
LAN
backbone
DNS resolver
authoritative DNS server
✔ Windows Vista+
✔ MacOS X
✔ Linux
Tampere 2010

Pervasive multihoming

Challenges

Tampere 2010

Network of the (near) future

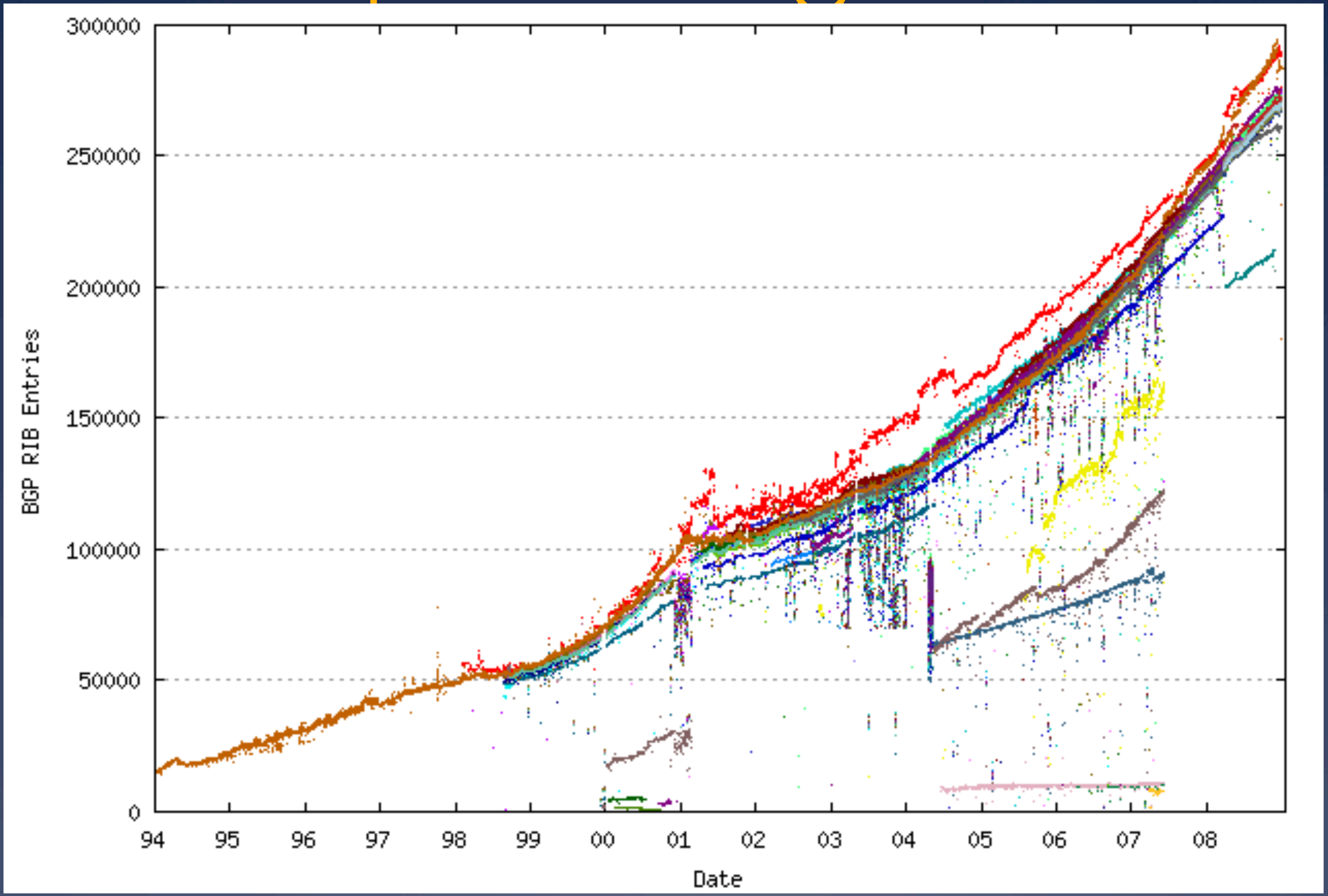


Homes passed by multiple networks → increase reliability by connecting to all ("reliable system out of unreliable components")

Multihoming (& mobility)

- Current IPv4 address ➔
  - identifier = unique host or **interface**
  - locator = network that serves host (provider)
- One system, multiple addresses:
  - multihoming: at the same time
  - mobility: sequentially
- Multihoming:
  - connections need to be aware of network path
  - socket interface makes it hard to program
- Solutions:
  - HIP: cryptographic host identifier
  - SHIM6
  - LISP: two network addresses
  - DNS: SRV, NAPTR

Example: BGP growth



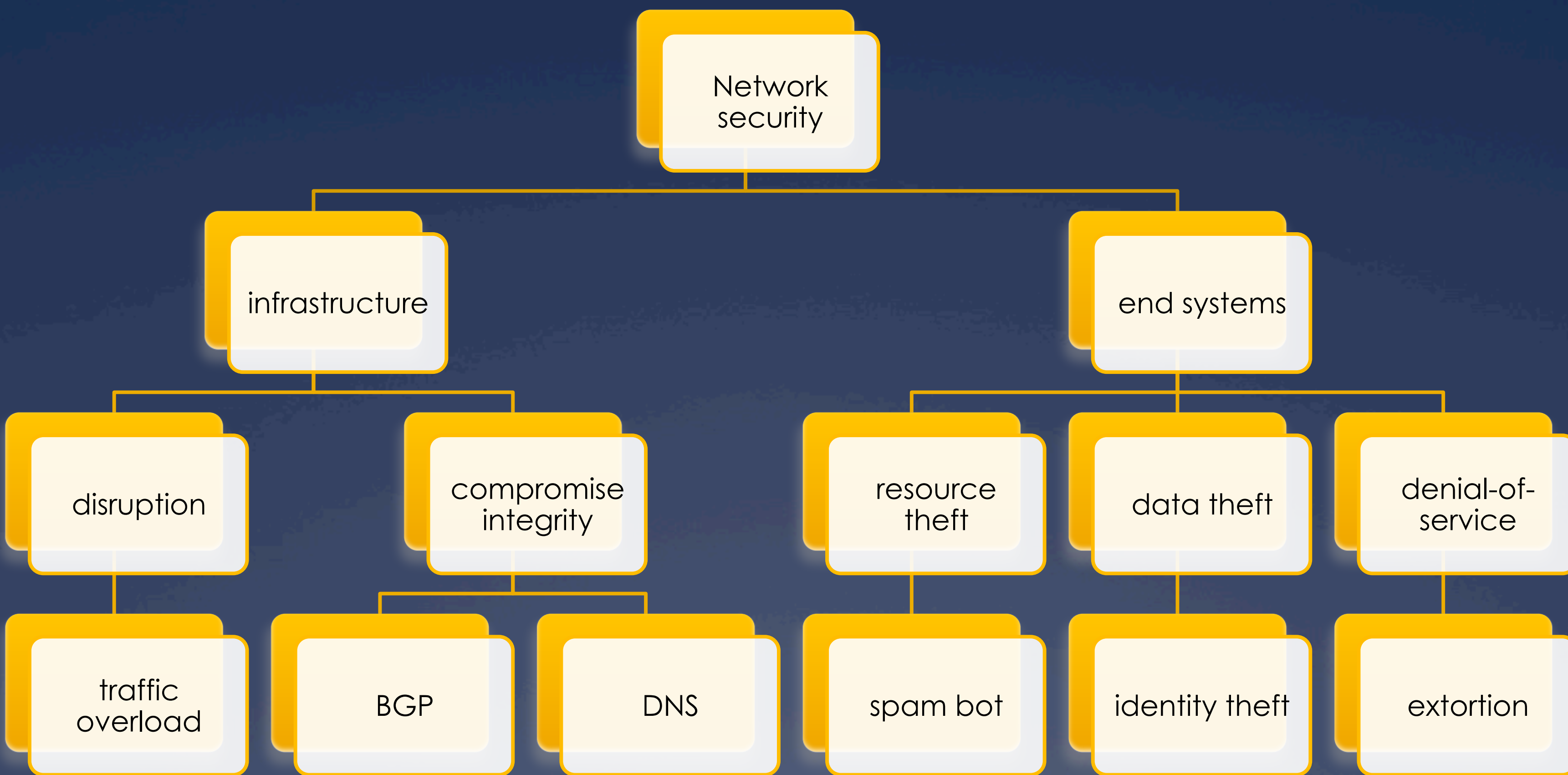
http://bgp.potaroo.net/

Security

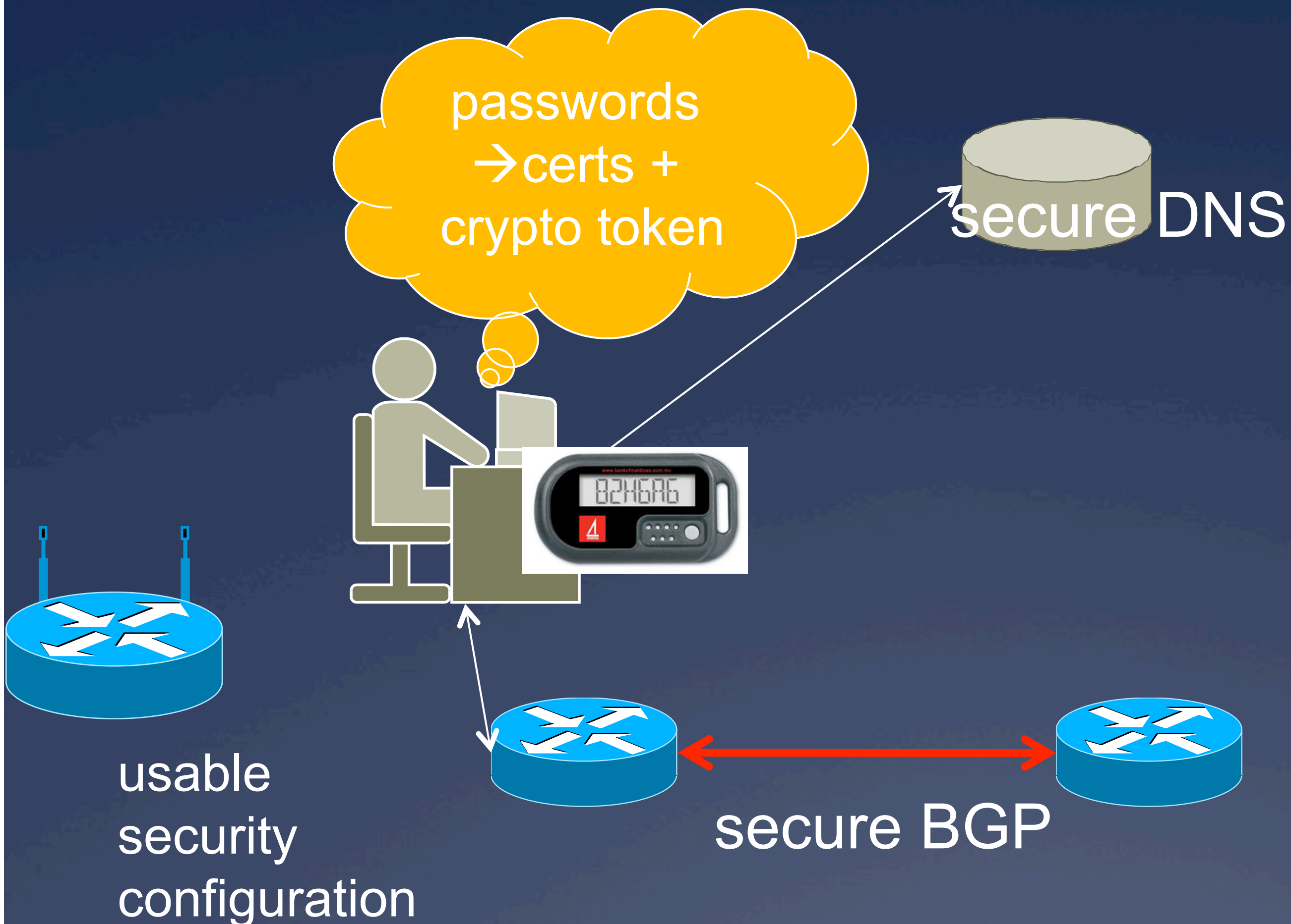
Challenges

Tampere 2010

Network security issues



What about security?



9: Political
8: Financial
Application
Presentation
Session
Transport
Network
Link
Physical

Technologies (mostly) available, but use & deployment hard

What about security?

- “The future Internet must be secure”
- Most security-related problems are **not** network problems
  - spam: identity and access, not SMTP
  - web: (mostly) not TLS, but distinguishing real bank from fake one
  - web: cross-domain scripting, code injection
  - browser vulnerabilities & keyboard sniffers
- Restrict generality
- Black list → white list
  - virus checker → app store
- Automated tools
  - better languages, taint tracking, automated input checking, stack protection, memory randomization, …
- Probably need more trust mediation

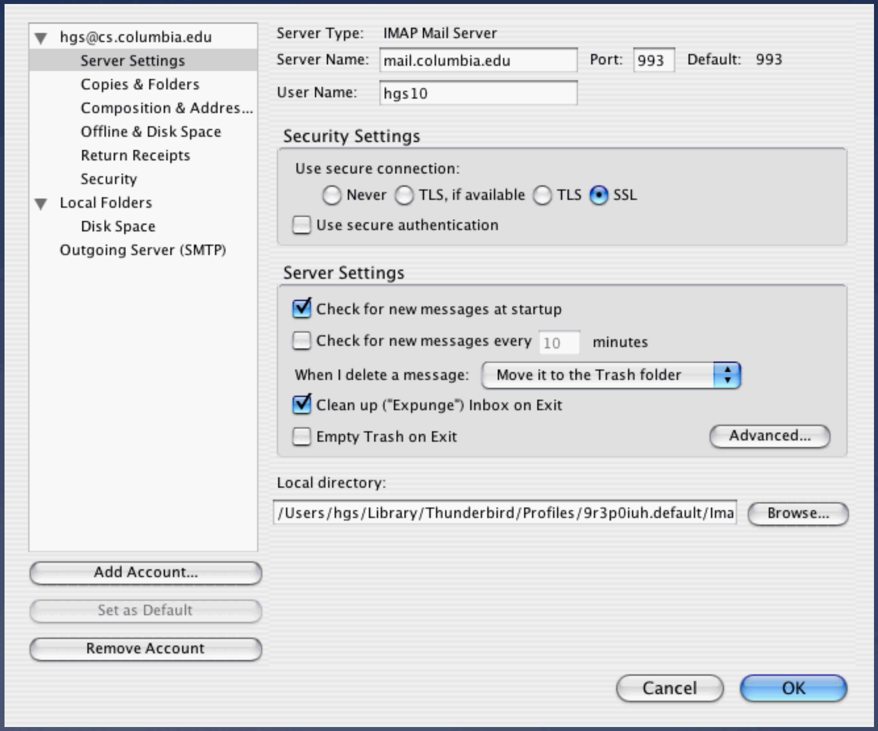
Usability

Challenges

Tampere 2010

Usability: Email configuration

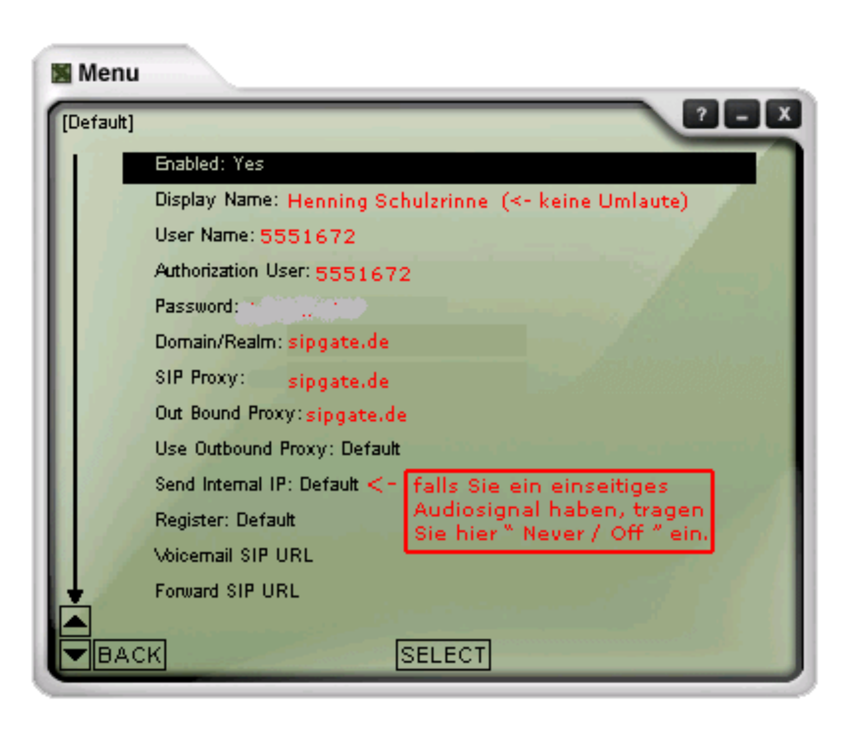
- Application configuration for (mobile) devices painful
- SMTP port 25 vs. 587
- IMAP vs. POP
- TLS vs. SSL vs. "secure authentication"
- Worse for SIP...



Usability: SIP configuration

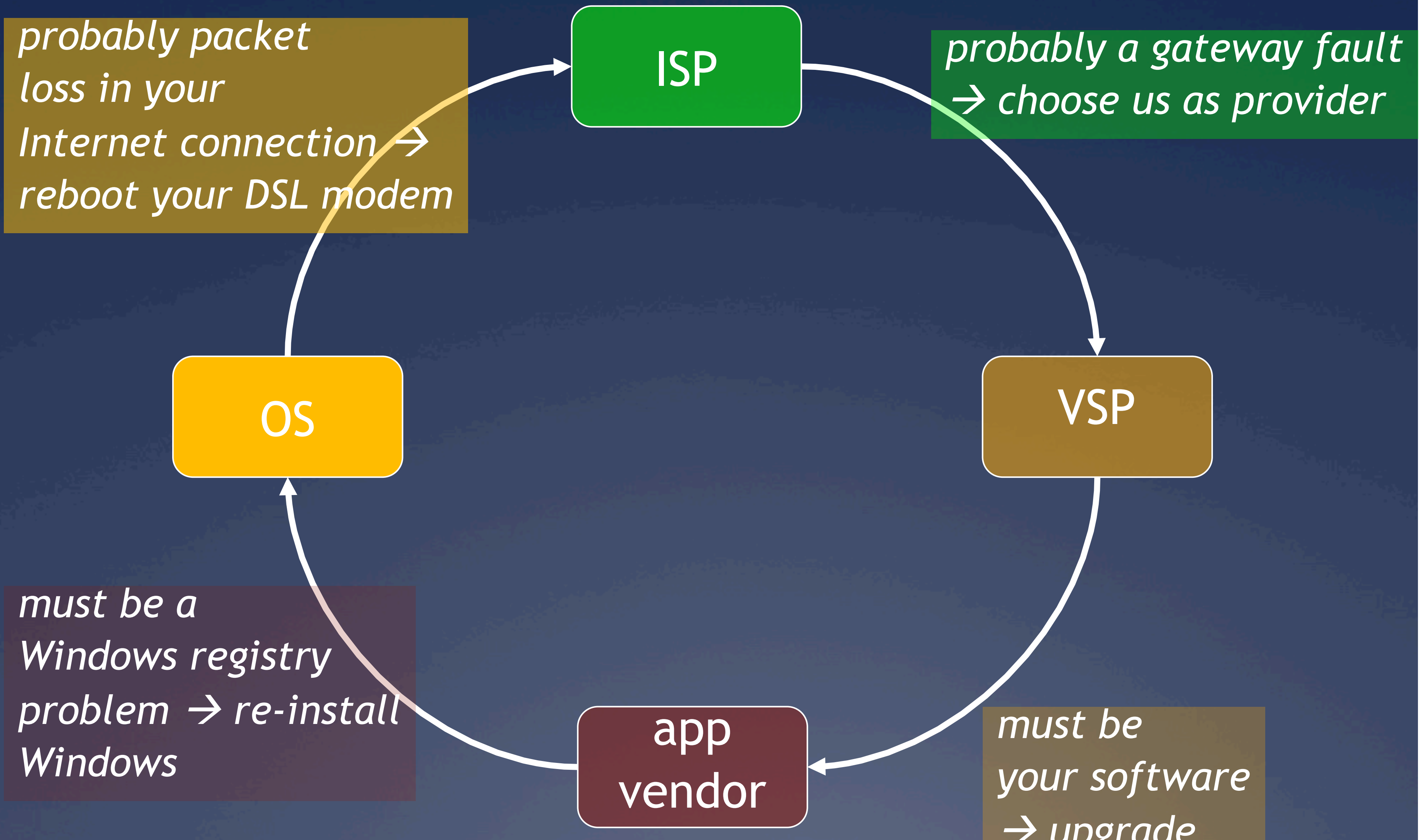
partially explains

* highly technical parameters, with differing names
* inconsistent conventions for user and realm
* made worse by limited end systems (configure by multi-tap)
* usually fails with some cryptic error message and no indication which parameter
* out-of-box experience not good

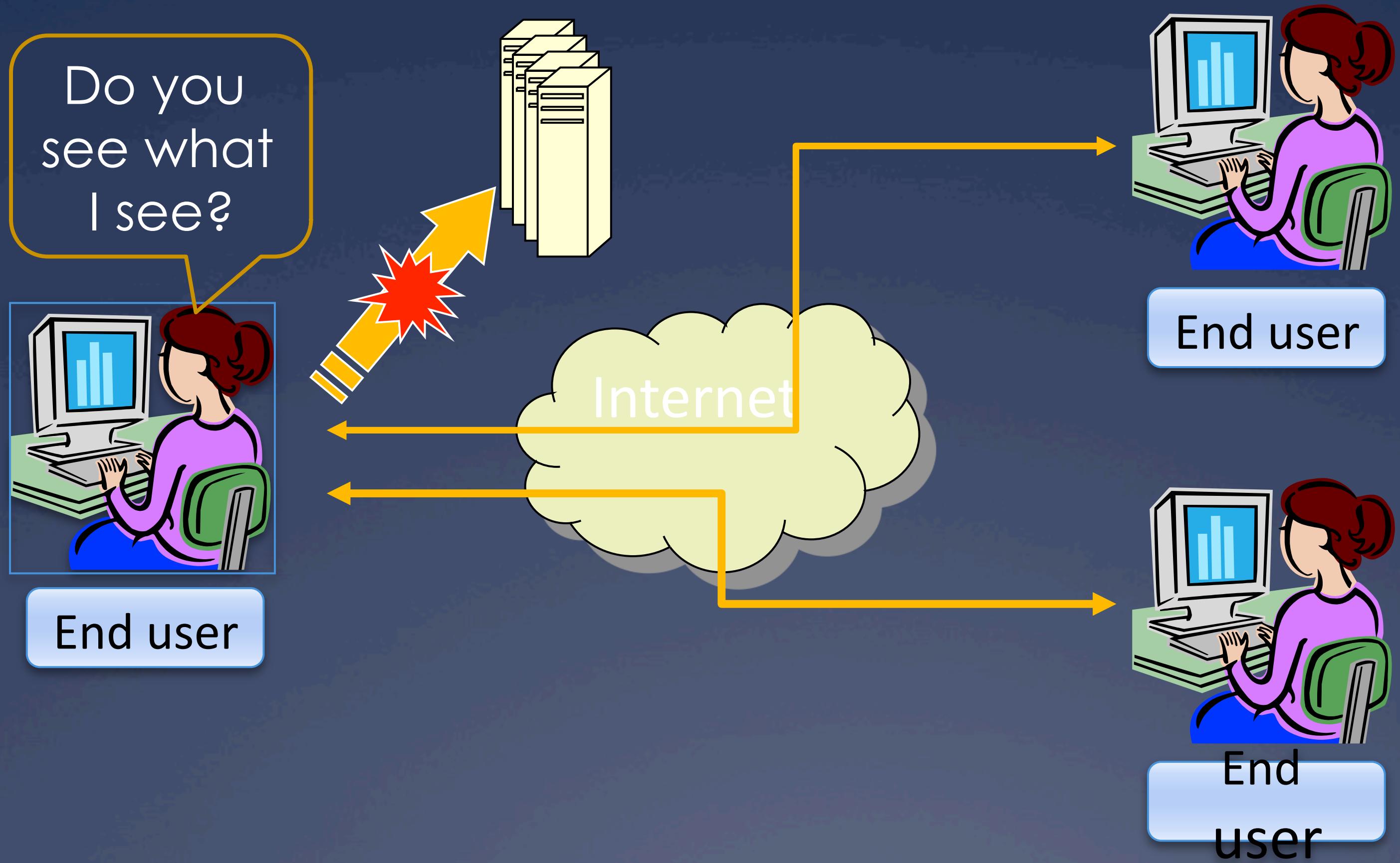


Usability: Interconnected devices

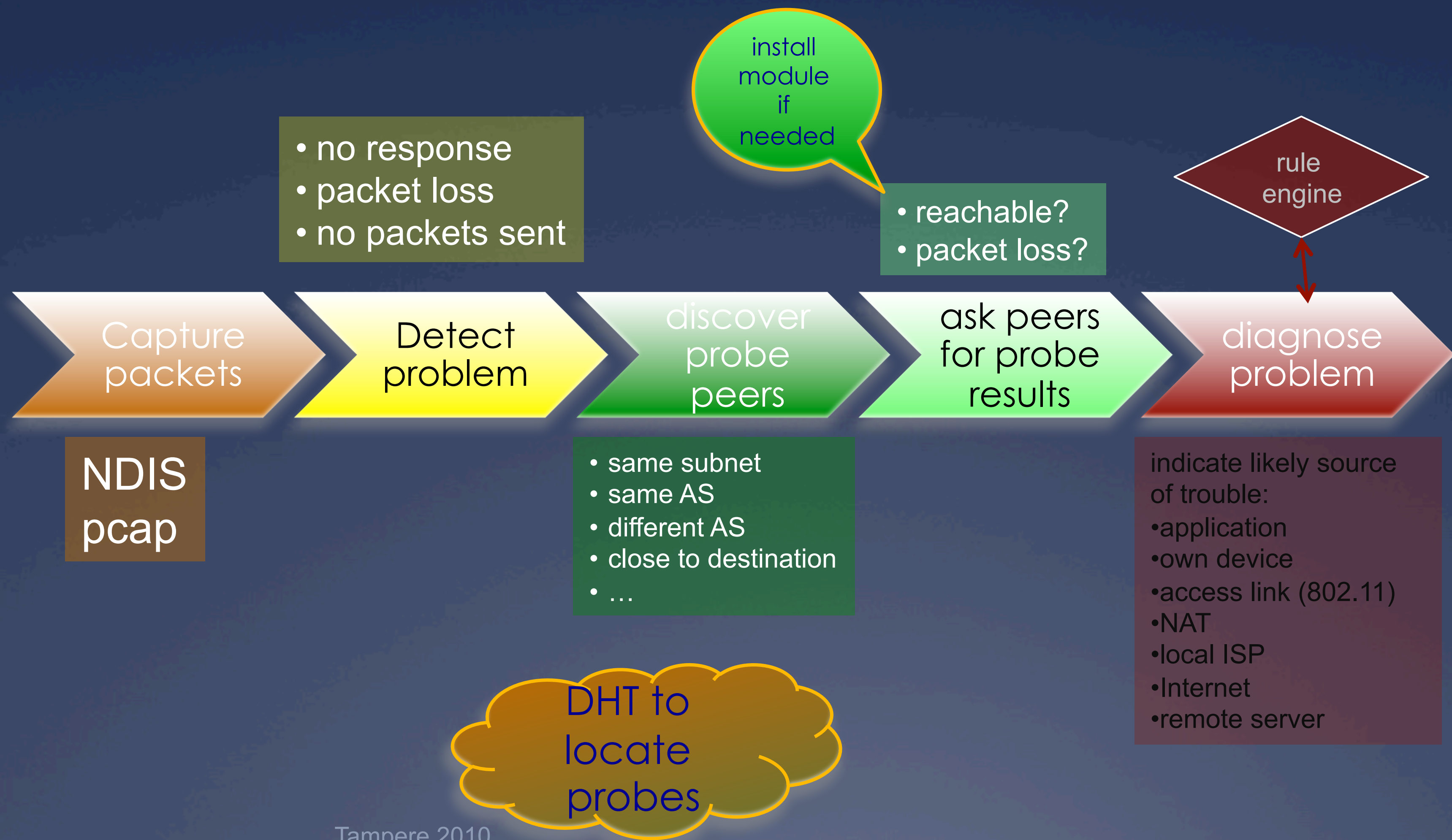
Circle of blame



DYSWIS = Do You See What I See?



DYSWIS



install module if needed
• no response
• packet loss
• no packets sent
• reachable?
• packet loss?
rule engine
Capture packets
Detect problem
discover probe peers
ask peers for probe results
diagnose problem
NDIS
pcap
• same subnet
• same AS
• different AS
• close to destination
• …
indicate likely source of trouble:
•application
•own device
•access link (802.11)
•NAT
•local ISP
•Internet
•remote server
DHT to locate probes

Mobility

Challenges

Tampere 2010

What about "the mobile Internet?"

- Same & different:
  - same
    - expect same services, applications (and speed)
      - fixed devices may acquire "app" model
        - task-focused, rather than file-focused
        - defined interfaces → easier to secure
    - reliability & predictability
  - Different
    - user interaction
      - secondary attention
    - context and sensing
    - disruption tolerant

What if?

I want to read WSJ, but no connection.

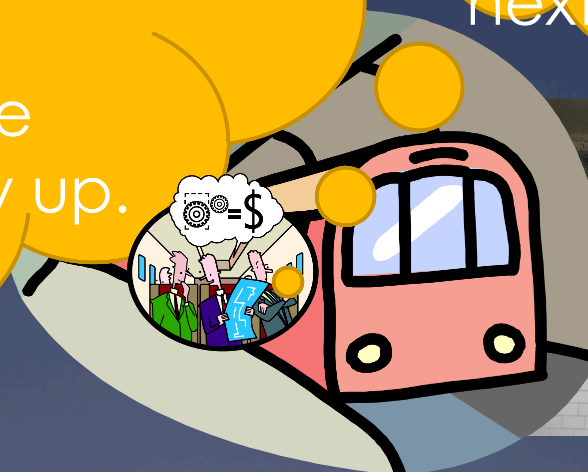
I want to read NY times, but no connection.

I want to send email to my boss, but no connection.

Disruption-tolerant networking

We have a connection. We only have 20 seconds to download the webpage. Hurry up.

Oops! I missed the chance. I will send my email next stop

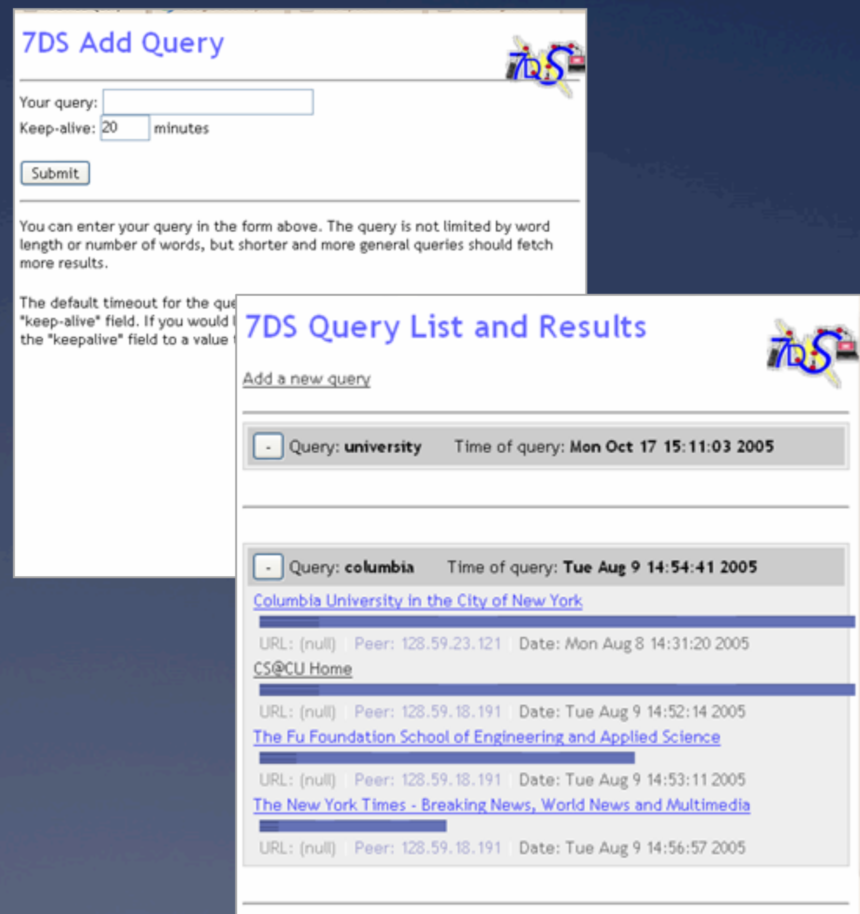


Opportunistic Networks

In the absence of the Internet, nodes can for an ad-hoc 802.11 wireless network and exchange data

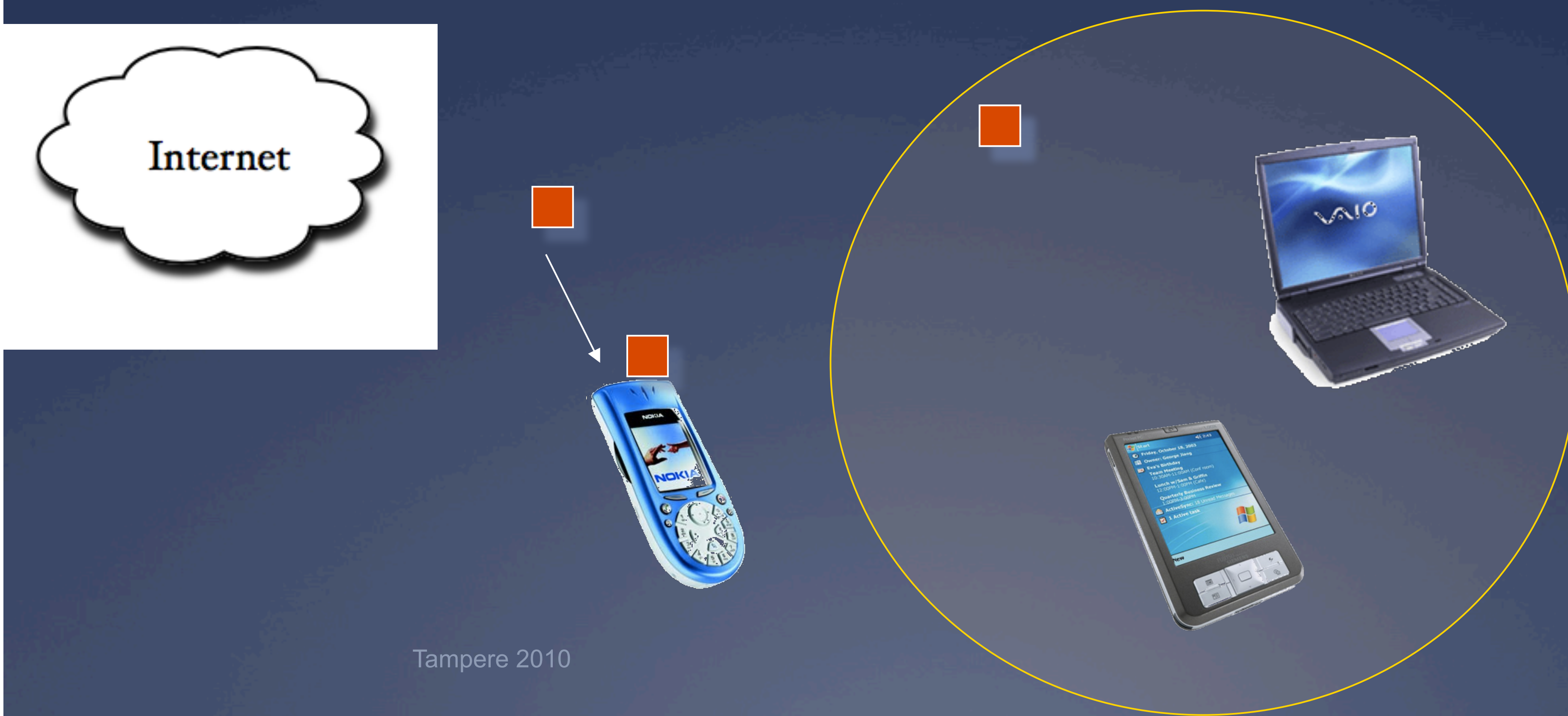
7DS

- Application suite: allows users to exchange data in disconnected networks
  - Distributed query and search
  - Mail Transfer Agent (MTA)
  - Automatic file synchronization



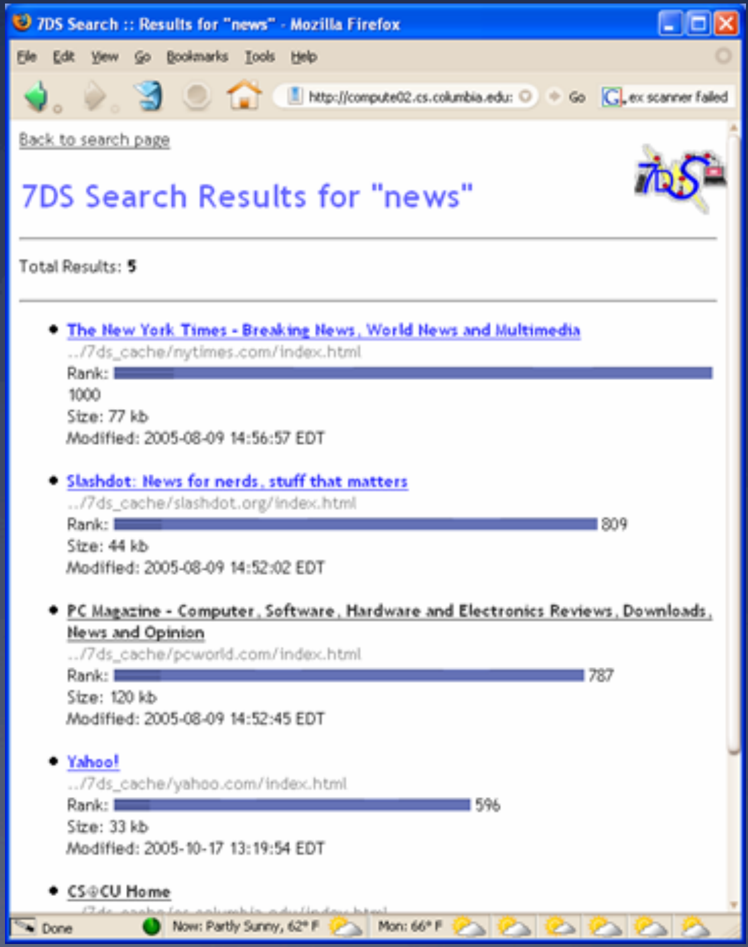
Web Delivery Model

* 7DS core functionality: Emulation of web content access and e-mail delivery

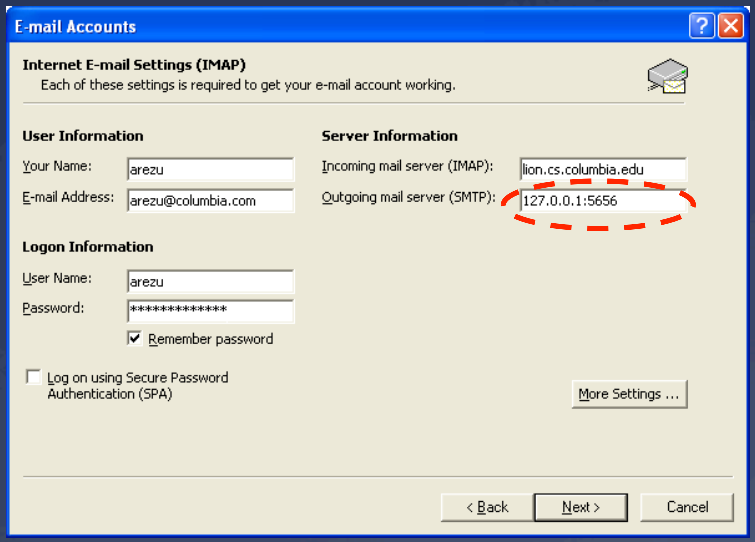


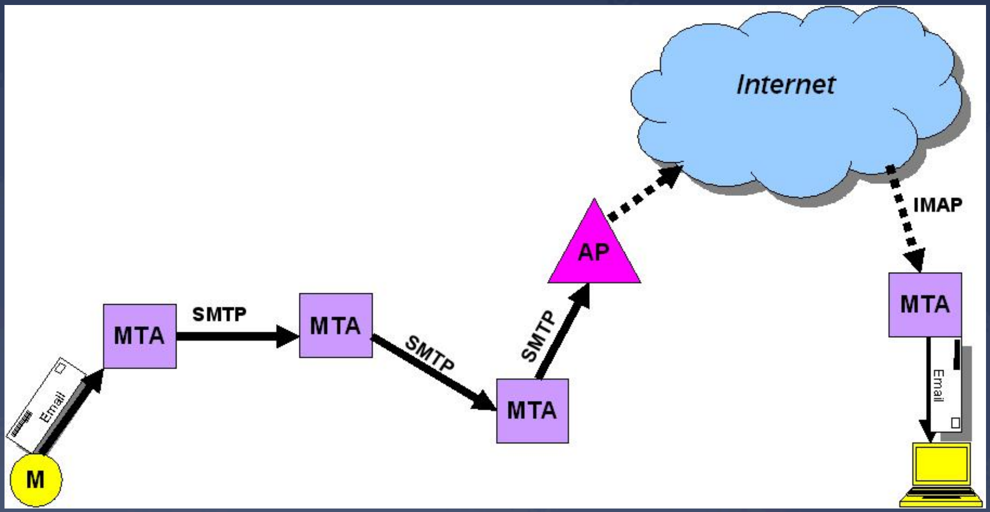
Search Engine

- Provides ability to query self for results
- Searches the cache index using Swish-e library
- Presents results in any of three formats: HTML, XML and plain text
- Similar in concept to Google Desktop

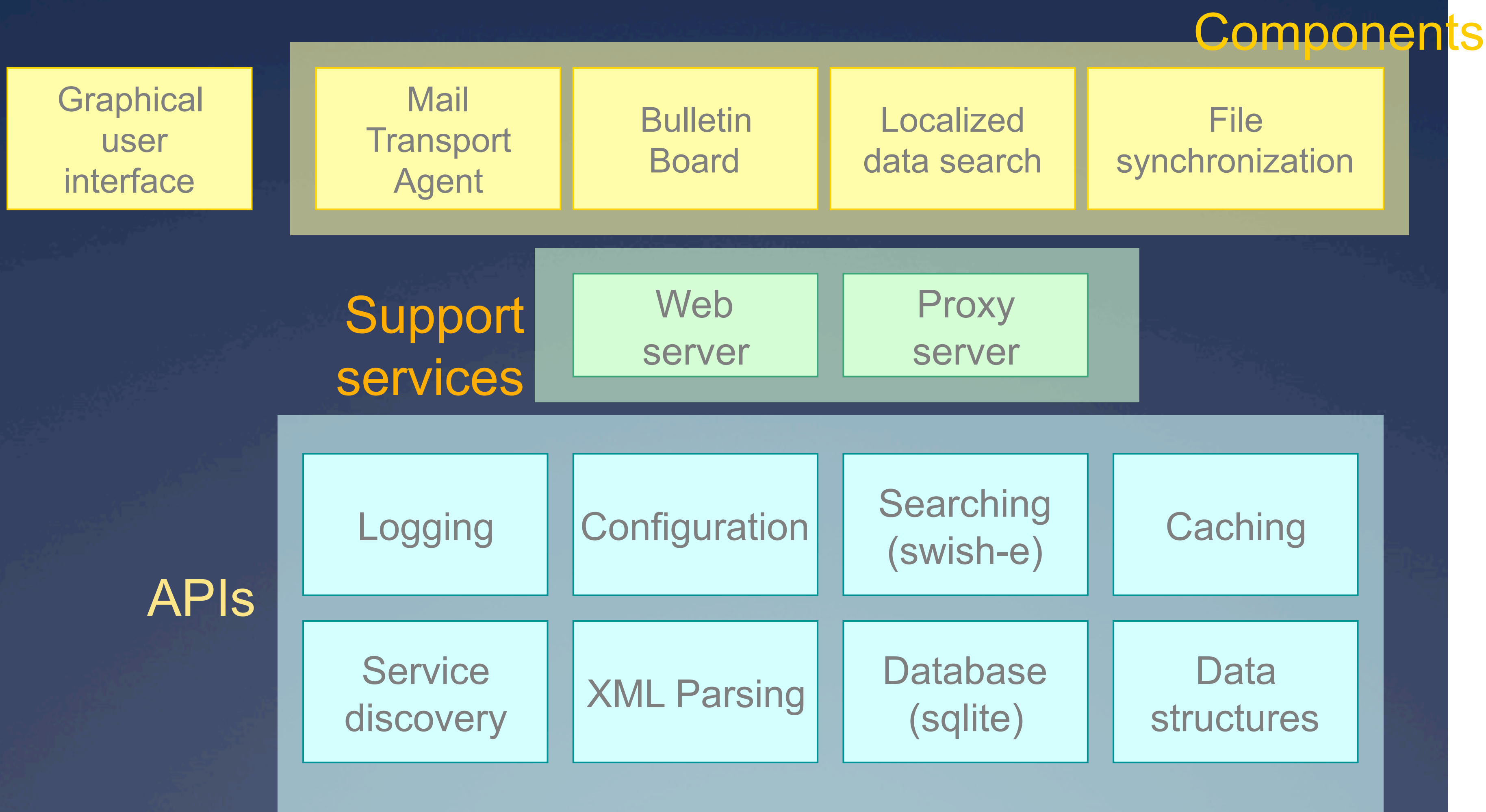


Email exchange



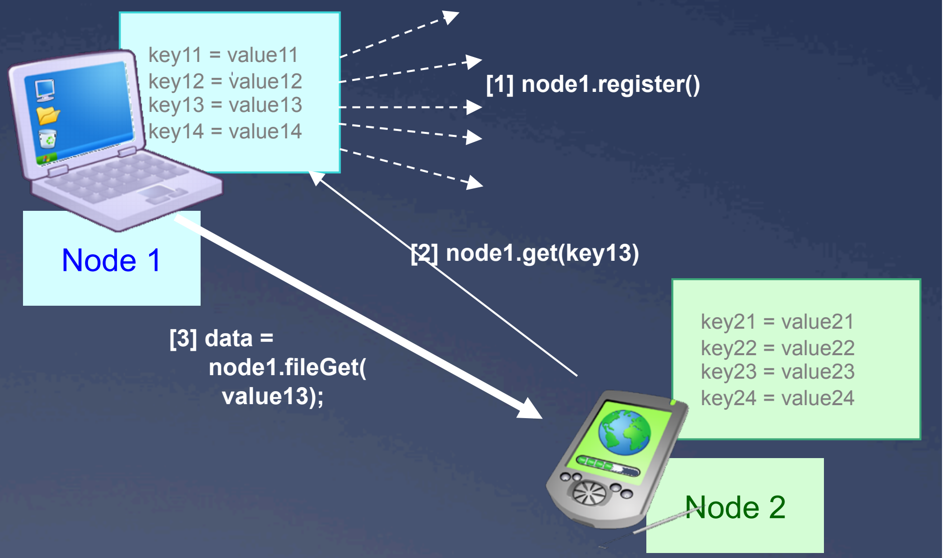


7DS architecture



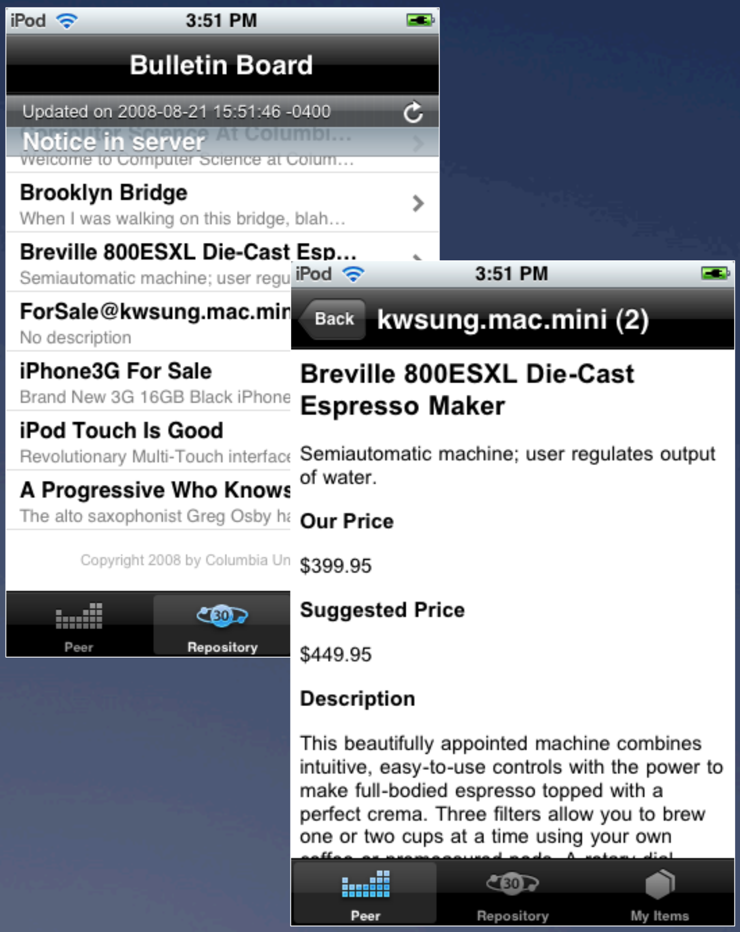
BonAHA service model

- Found a need for application framework for opportunistic networks: **BonAHA**
- Responds to events on network
  - serviceUpdated()
  - serviceExited()
- Access node properties
  - *node*.get()
  - *node*.set()



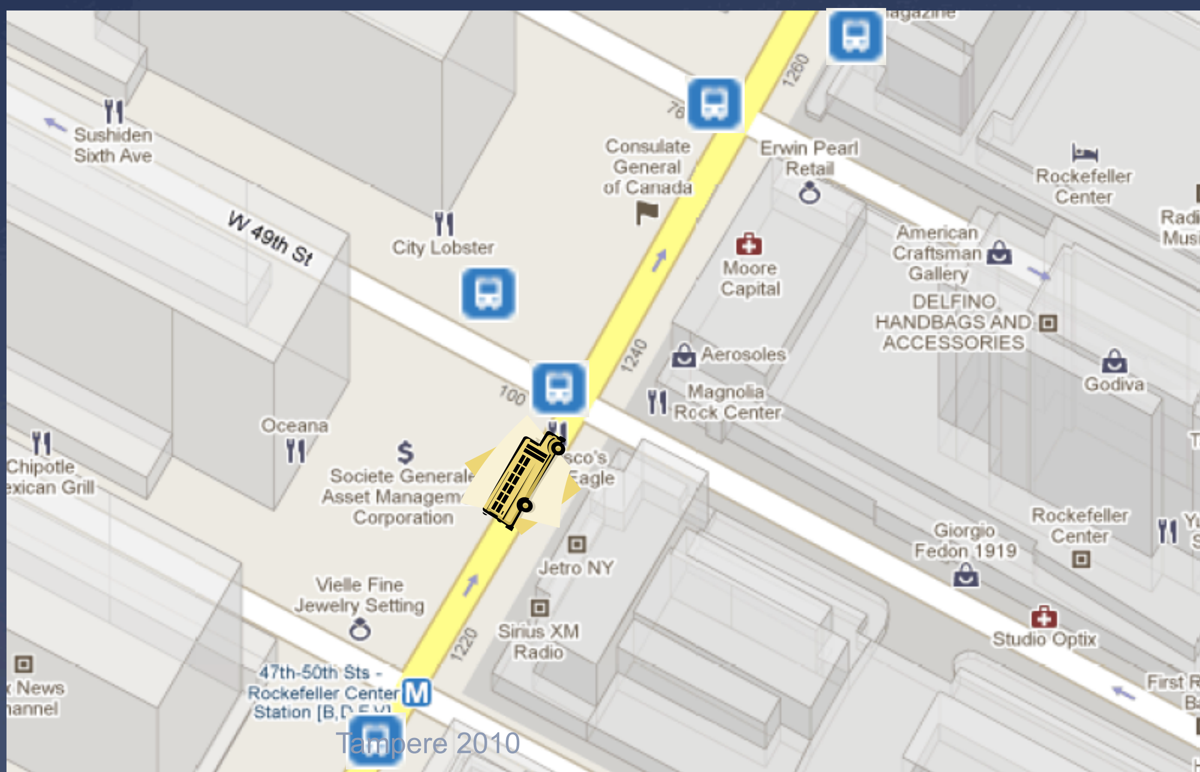
Applications: Bulletin Board System

- Can create and share posts
  - Other users can browse your posts
- Similar to a real-world paper bulletin board
- Create and share information in opportunistic networks
- iPhone platform



Bus model

* Public transportation (bus-stop) model
* Deterministic knowledge (temporal and spatial information)
  * Location of next bus stations (stops)
  * Expected next opportunity: (calculated by average speed of the bus)

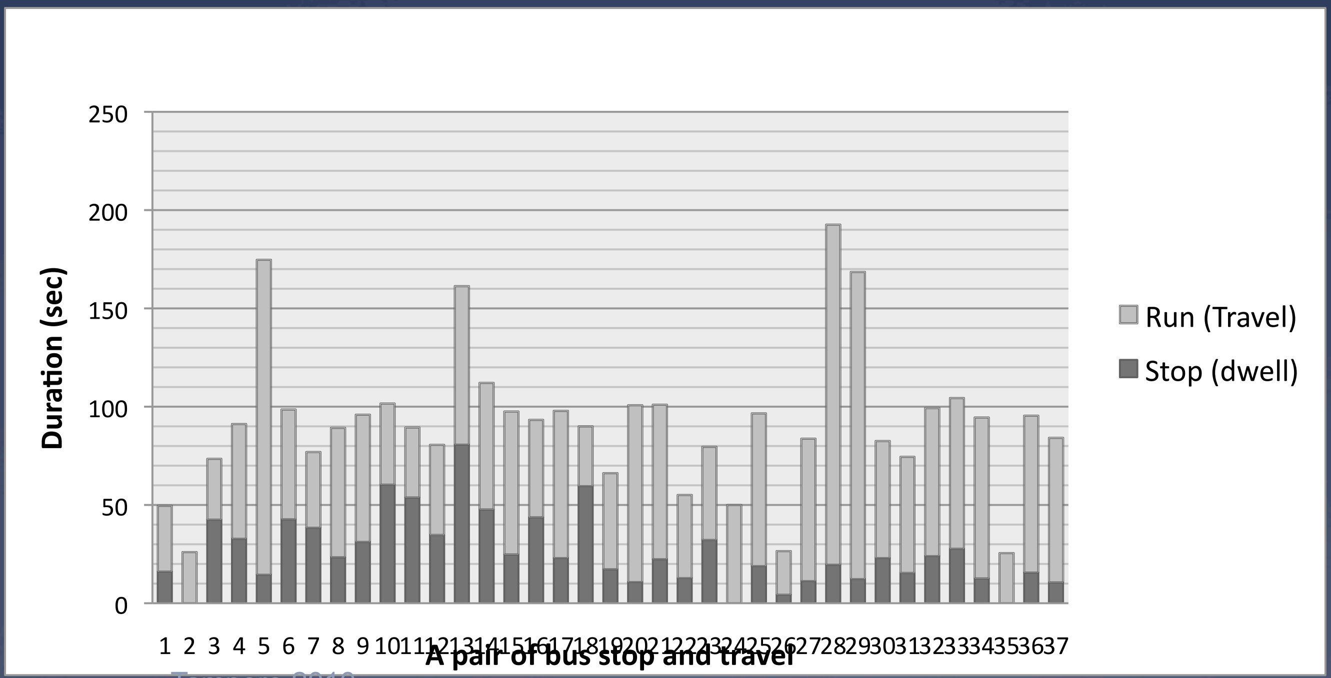


Manhattan
49^{th} St, 6^{th} Ave.

Bus station

Bus measurement

- Measurement of bus dwell time (stop time) and travel time in Manhattan
  - 2:30 PM – 3:30 PM, Jan, 2010
  - 116st, Broadway – 42st, 1 Ave
- Average bus dwell time is 26 sec; average bus travel time is 65.4 sec

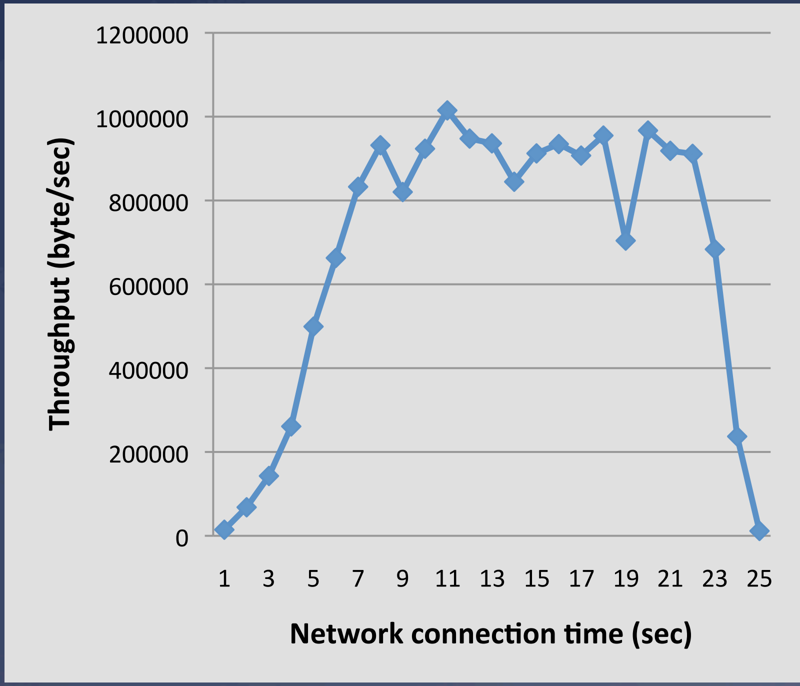


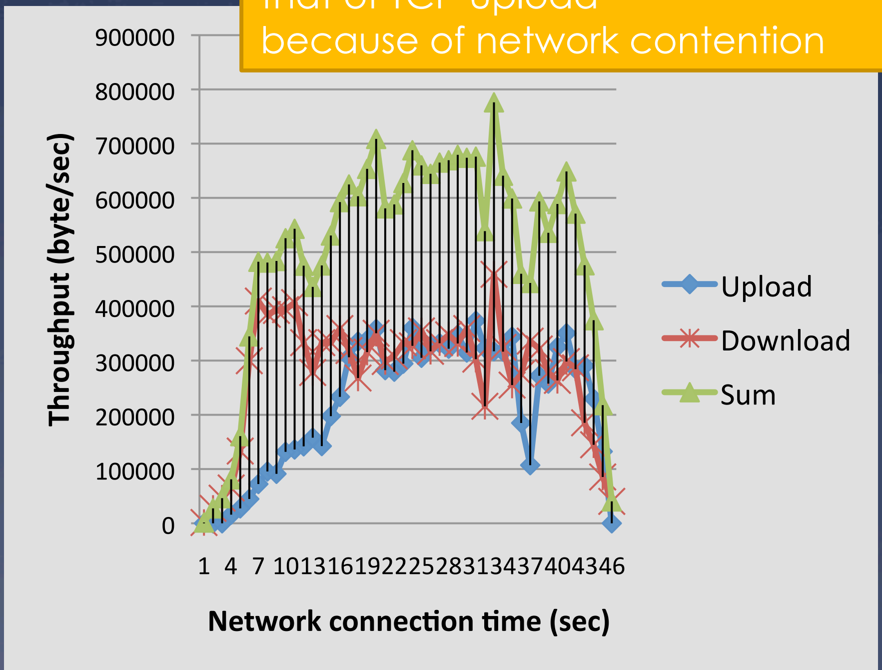
TCP goodput via IEEE 802.11g

- TCP-upload only
- Total network connection time: 25 sec
- Bus dwell time: 11 sec

- TCP-two-way (upload and download)
- Total network connection time: 46 sec
- Bus dwell time: 26.7 sec

Total throughput is smaller than that of TCP-upload because of network contention





Programmability

Challenge

Tampere 2010

Usage transition

Limited personal communication

- email
- static information retrieval (ftp → web)
- phone
- 3 core applications

Content-based

- large-scale distribution of popular content (entertainment video)

Personalized content and computation

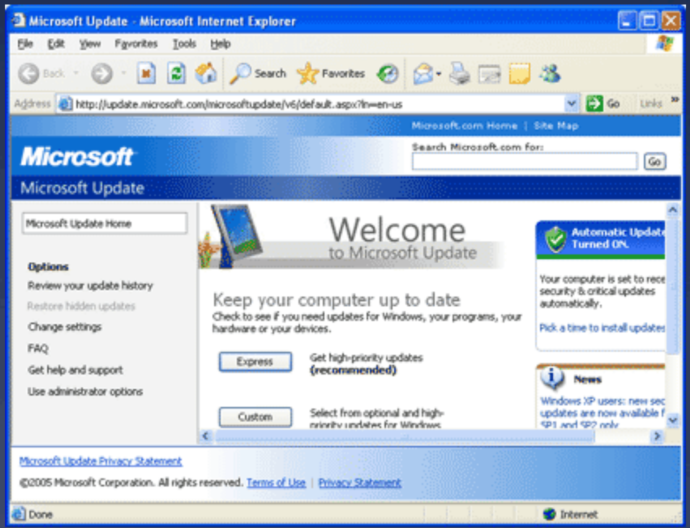
- social networks
- context-based information
- millions of tiny apps

Two worlds

10+ interfaces
0 GB disk
1 low-end processor

1 interface
TB disk
1-32 multi-core processors

Software: from floppy to autonomous



NetServ overview

Extensible architecture for core network services

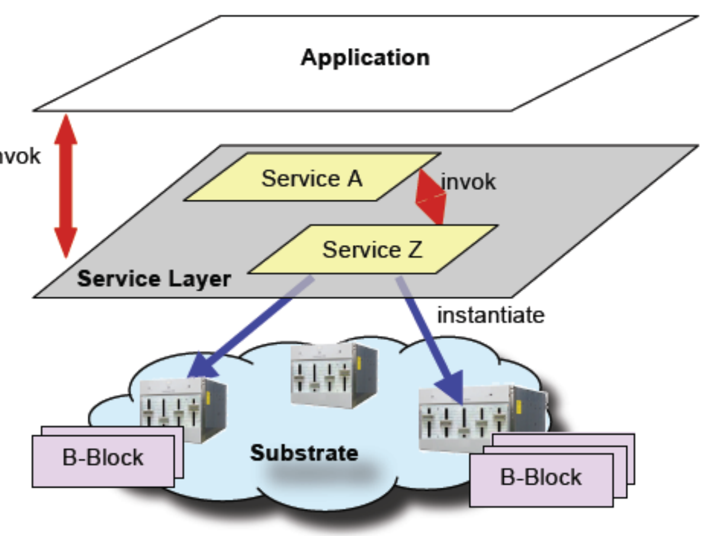


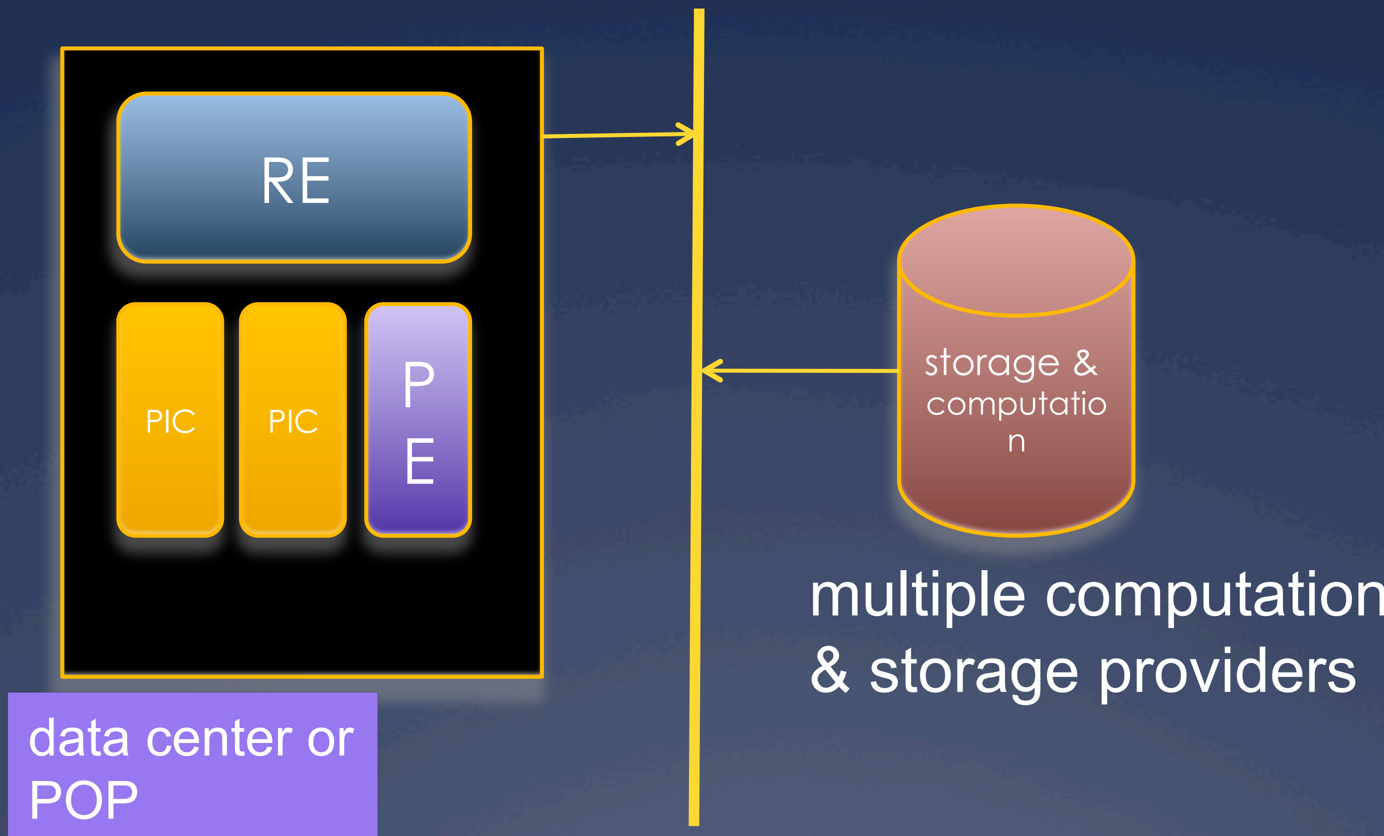
Figure 3: Instantiation of services over tunable building blocks.

Modularization
- Building Blocks
- Service Modules

Virtual services framework
- Security
- Portability

NSF FIND four-year project
- Columbia University
- Bell Labs
- Deutsche Telekom
- DOCOMO Euro-Labs

Network node example



Different from active networks?

* Active networks
  * Packet contains executable code or pre-installed capsules
    * Can modify router states and behavior
    * Mostly stateless
  * Not successful
    * Per-packet processing too expensive
    * Security concerns
    * No compelling *killer app* to warrant such a big shift
  * Notable work: ANTS, Janos, Switchware

* NetServ
  * Virtualized services on current, passive networks
    * Service invocation is signaling driven, not packet driven
    * Some flows & packets, not all of them
    * Emphasis on storage
  * Service modules are stand-alone, addressable entities
    * Separate from packet forwarding plane
    * Extensible plug-in architecture

How about GENI?

- GENI = global-scale test bed for networking research
  - parallel experiments in VMs
- ➔ long-term, "heavy" services
- We'll be demonstrating use of NetServ on GENI this June during GEC8

Service modules

- Full-fledged service implementations
  - Use building blocks and other service modules
  - Can be implemented across multiple nodes
  - Invoked by applications
- Examples:
  - Routing-related services
    - Multicast, anycast, QoS-based routing
  - Monitoring services
    - Link & system status, network topology
  - Identity services
    - Naming, security
  - Traffic engineering services
    - CDN, redundancy elimination, p2p network support

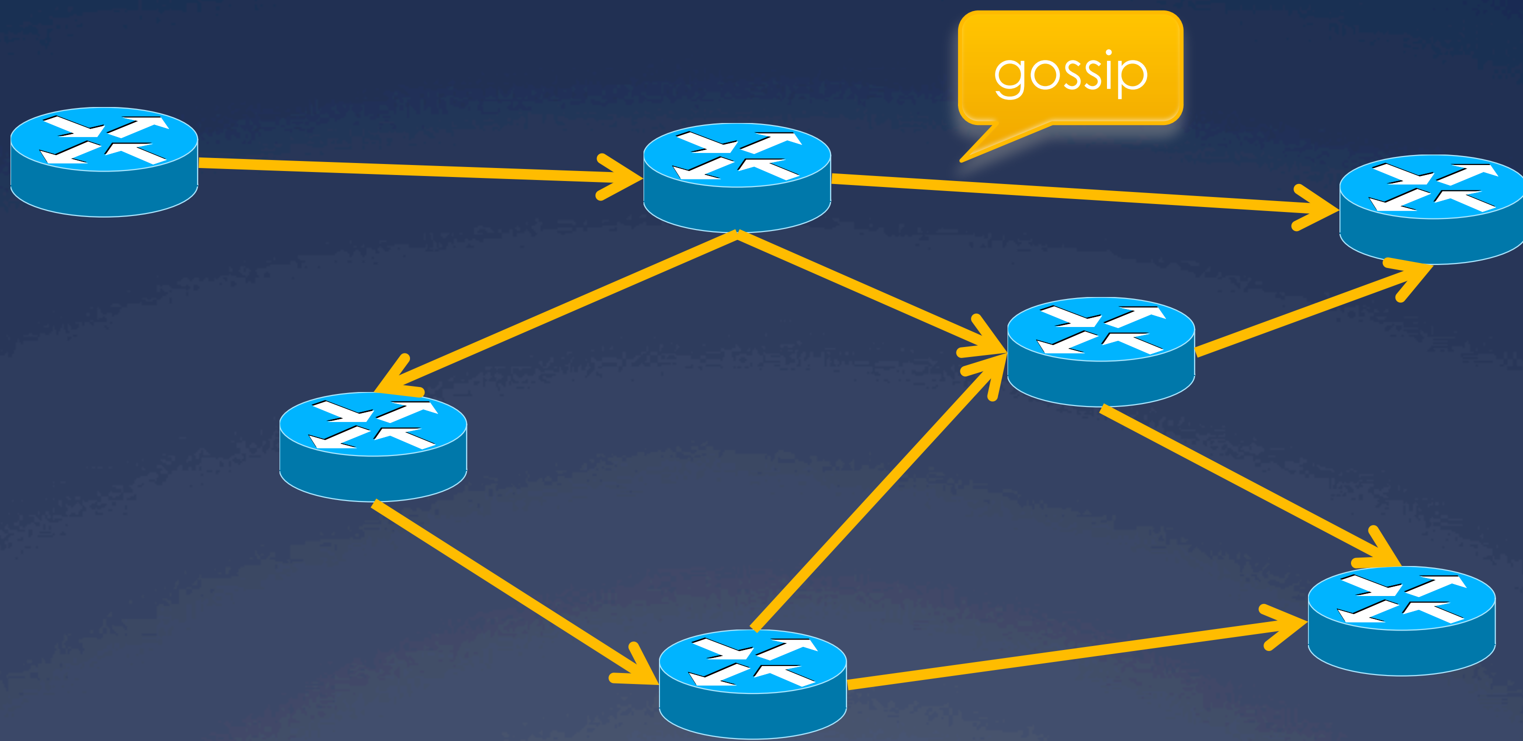
Deployment scenarios

- Three actors
  - Content publisher (e.g. youtube.com)
  - Service provider (e.g. ISP)
  - End user
- Model 1: Publisher-initiated deployment
  - Publisher rents router space from providers (or end users)
- Model 2: Provider-initiated deployment
  - Publisher writes NetServ module
  - Provider sees lots of traffic, fetches and installs module
  - Predetermined module location (similar to robots.txt)
- Model 3: User-initiated deployment
  - User installs NetServ module to own home router or PC
  - *or on willing routers along the data path*

Where does code run?

* All (or some?) nodes in a network
  * AS, enterprise LAN
* Some or all nodes along path
  * data path from source to destination
* Selected nodes by property
  * e.g., one in each AS

How does code get into nodes?



All nodes in (enterprise) network

OSGi

* Architecture
  * **Bundles**: JAR files with manifest
  * **Services**: Connects bundles
  * **Services Registry**: Management of services
  * **Modules**: Import/export interfaces for bundles

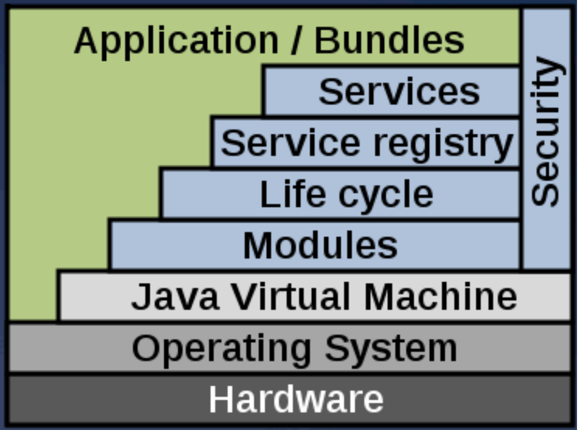
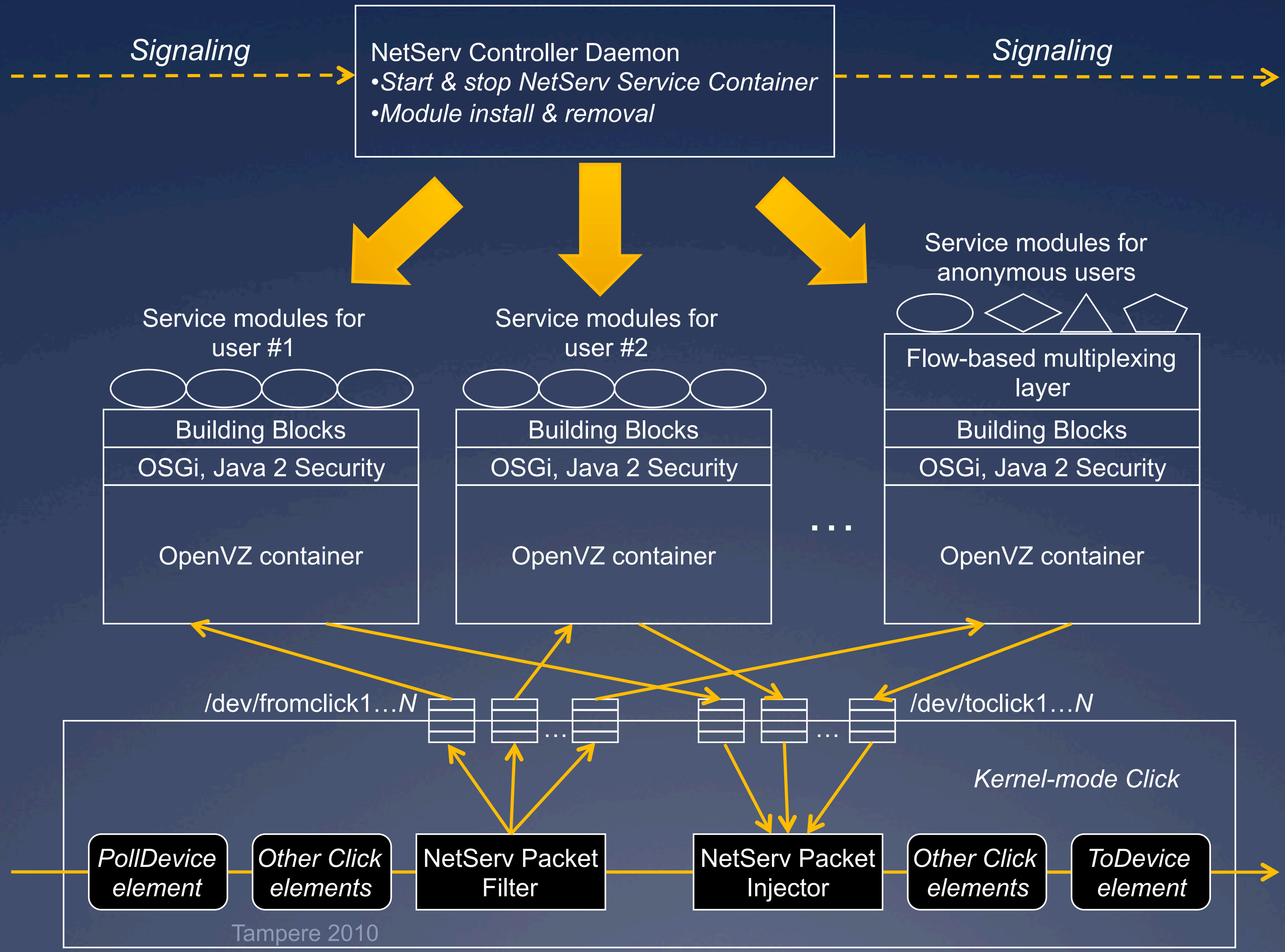
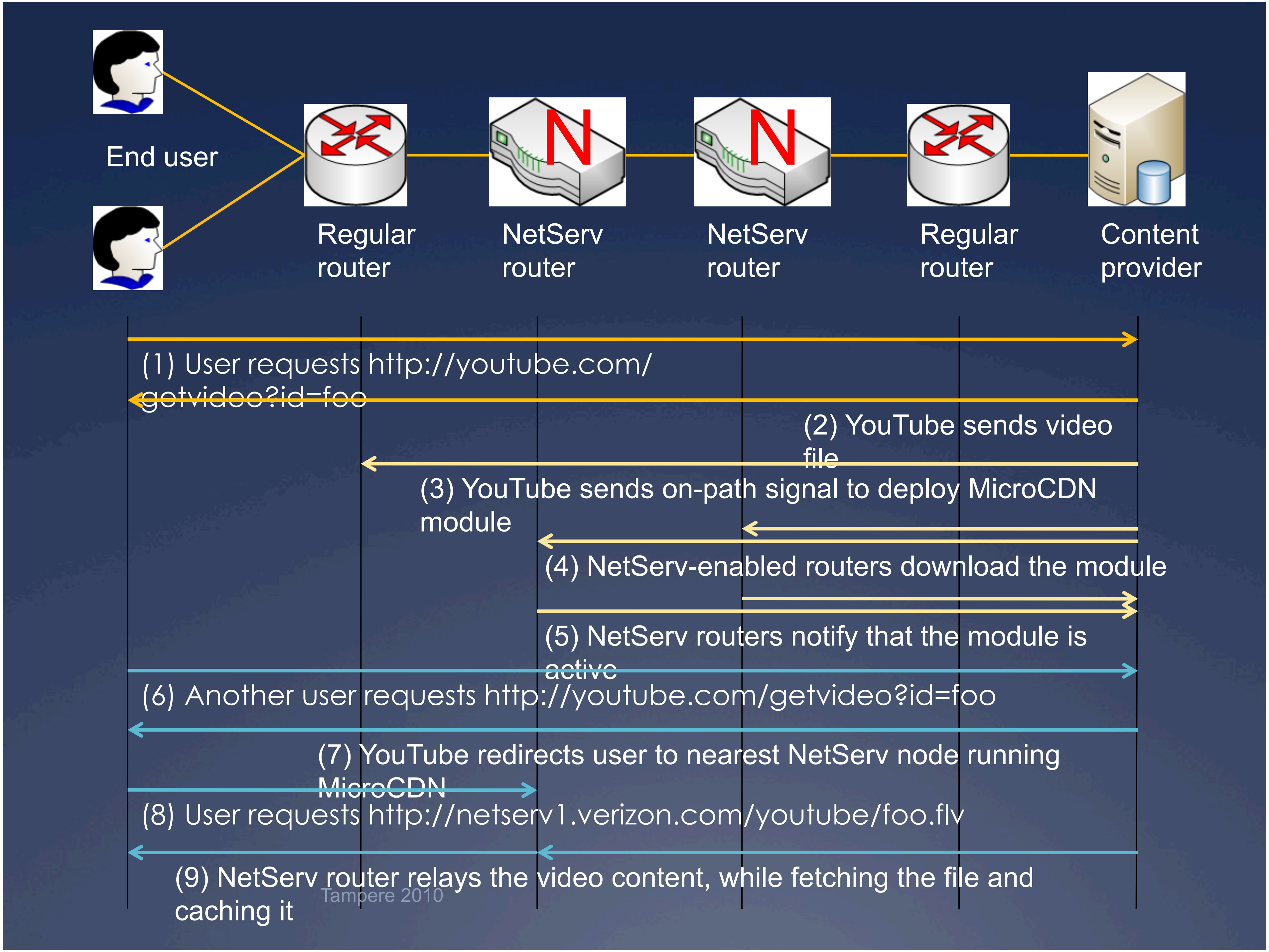


Image credit: Wikipedia

* Possible to "wrap" existing Java apps and JARs
  * Add additional manifest info to create OSGi bundle
  * E.g.: Jetty web server now ships with OSGi manifest; now extensively used with OSGi containers and custom bundles
  * For NetServ, we created a OSGi bundle for the Muffin HTTP proxy server

Current architecture




End user
Regular router
NetServ router
NetServ router
Regular router
Content provider
(1) User requests http://youtube.com/getvideo?id=foo
(2) YouTube sends video file
(3) YouTube sends on-path signal to deploy MicroCDN module
(4) NetServ-enabled routers download the module
(5) NetServ routers notify that the module is active
(6) Another user requests http://youtube.com/getvideo?id=foo
(7) YouTube redirects user to nearest NetServ node running MicroCDN
(8) User requests http://netserv1.verizon.com/youtube/foo.flv
(9) NetServ router relays the video content, while fetching the file and caching it
Tampere 2010

context →

SECE: Sense Everything, Control Everything

- SECE allows users to create services that combine
  - communication
  - calendaring
  - location
  - devices in the physical world
- SECE is an event-driven system
  - that uses a high-level language
  - to trigger action scripts, written in Tcl.

SECE: Examples of rules

```
every sunset {
    homelights on;
}

every week on WE at 6:00 PM{
    email irt_list “Pizza talk at 6:00 PM today.”;
}

if my stock.google > 14 {
    sms me "google stock:"+[stock google];
}
```

SECE: Event-triggered actions

Events	Actions
▪ Presence updates ▪ Incoming calls ▪ Email ▪ Calendar entries ▪ Sensor inputs ▪ Location updates	▪ Controlling the delivery of email ▪ Routing phone calls ▪ Updating social network status ▪ Controlling actuators such as lights ▪ Reminders (email, voice call, SMS) ▪ Interacting with Internet services

SECE: The glue for Internet applications

Towards a future Internet

- Long-term constant = *service model*
  - equivalent of railroad track & road width
- Identify core Internet functions we need
  - routing
  - packet scheduling
  - congestion control
  - name lookup
  - path state establishment
  - …
- Learn from history
  - why didn't these get done "right"?
  - which functions should be done as application
- Need engineering principles
- Requirement list doesn't help

Conclusion

* Abandon notion of a clean-slate next-generation Internet
  * that magically fixes all of our problems
* Need for good engineering solutions
  * with user needs, not (just) vendor needs
* Research driven by real, not imagined, problems
  * factor 10 problems: reliability & OpEx
  * more reliability and usability, less sensor networks
* Build a 5-nines network out of unreliable components
* Make network disruptions less visible
* Transition to “self-service” networks
  * support non-technical users, not just NOCs running HP OpenView or Tivoli