ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

QOS – Quality Of Service

Michael Schär

Seminar in Distributed Computing

Outline

- Definition QOS
- Attempts and problems in the past (2 Papers)
- A possible solution for the future:
  Overlay networks (2 Papers)
- PlanetLab
- Personal opinion

Outline

- Definition QOS
- Attempts and problems in the past (2 Papers)
- A possible solution for the future:
  Overlay networks (2 Papers)
- PlanetLab
- Personal opinion

Definition of QOS

- „ A network that supports quality of service (QoS) is a network that presents its capabilities to the user and allows them to make choices as to the service they receive. Choices can be made in a number of dimensions:
Bandwith, Availability, Latency, Loss“
- „The effort to engineer an end-to-end alternative to best-effort packet delivery on the Internet”

Outline

- Definition QOS
- ☞ Attempts and problems in the past (2 Papers)
- A possible solution for the future:
  Overlay networks (2 Papers)
- PlanetLab
- Personal opinion

What was done in the past

- A huge amount of QOS research was done over the years for providing choices for different QOS guarantees in a variety of network
- Almost none of this research has had any impact, and certainly not in any way proportional to the expended time and effort

Failure to Thrive

Failure to thrive is a medical term which denotes poor weight gain and physical growth failure over an extended period of time in infancy.

(Source: Wikipedia)

Problem 1: Complexity (1)

- Complexity of an architecture is proportional to the contained components:
  - Protocol path
  - Software path
  - Physical path
- Problems for Scalability of a complex network:
  - Amplification principle: Local fluctuations can produce large-scale effects
  - Coupling principle: Unexpected interactions can happen between seemingly-isolated features and components

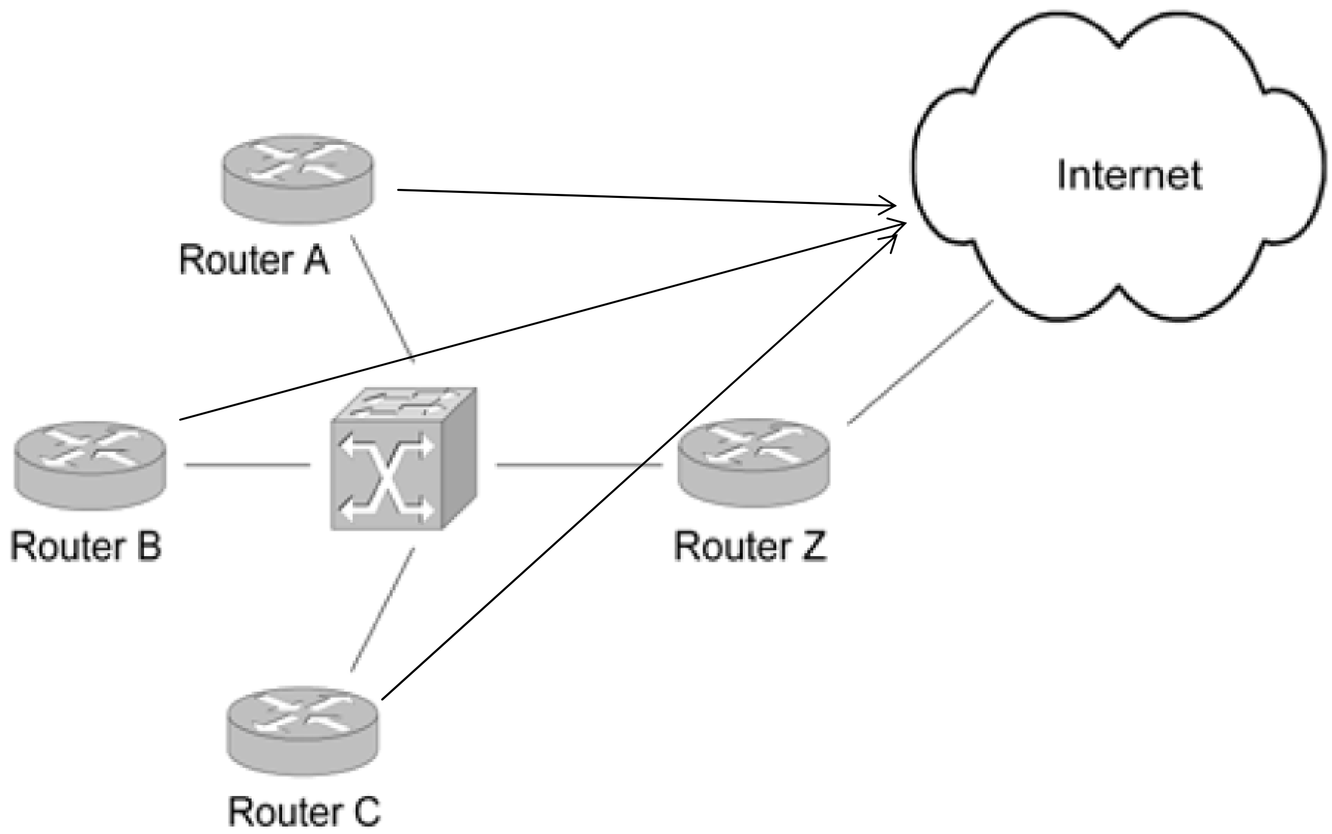
Problem 1: Complexity (2)

- Researches totally underestimate this problem
  - The protocols are designed to function
- Network engineers who manager router and switches have to deal with it
  - Find bugs
  - Report these bugs to vendors
  - Upgrade code

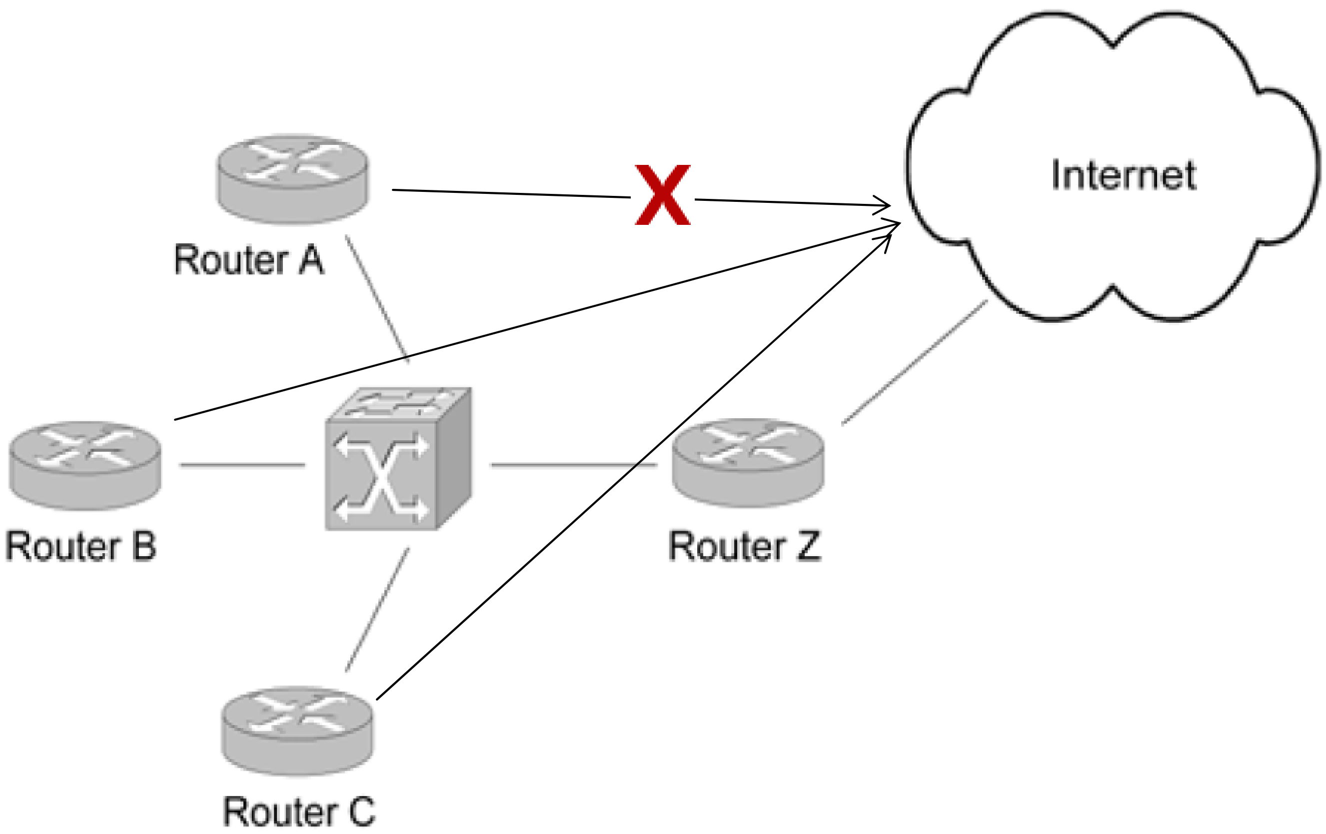
Complexity: Example

- Lawrence Berkeley National Laboratory (LBNL)
    - Scientific Research Institution
- 80 subnets
- 10000 connected devices

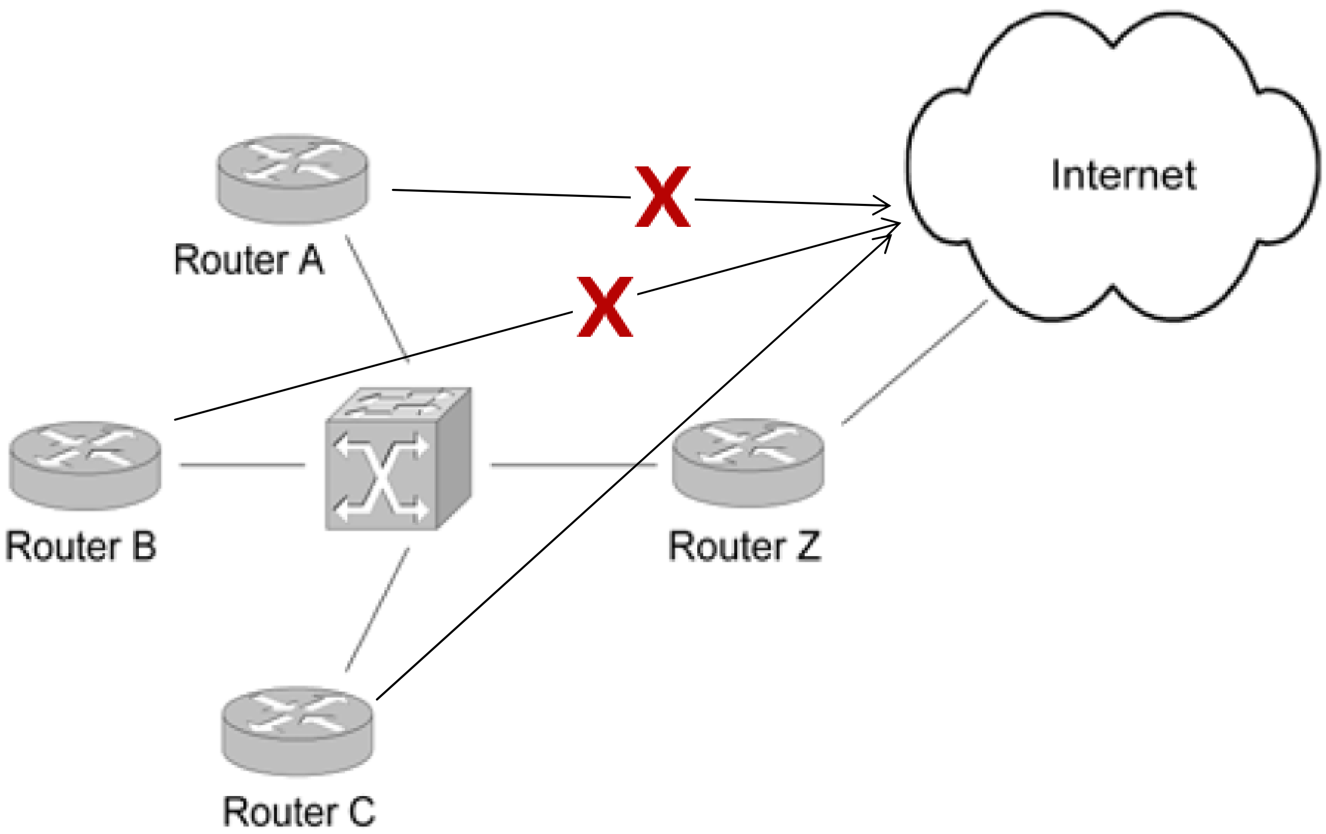
Complexity: Example



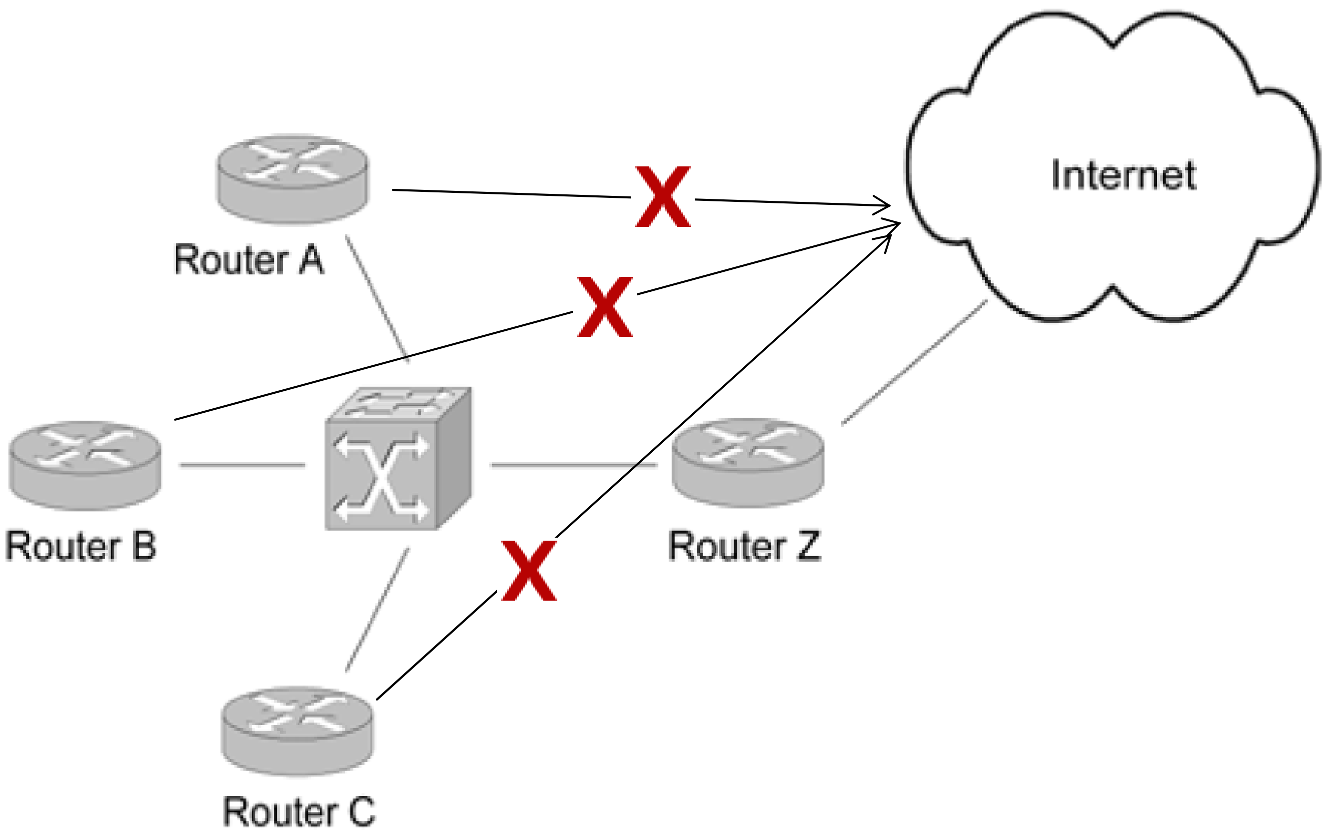
Complexity: Example



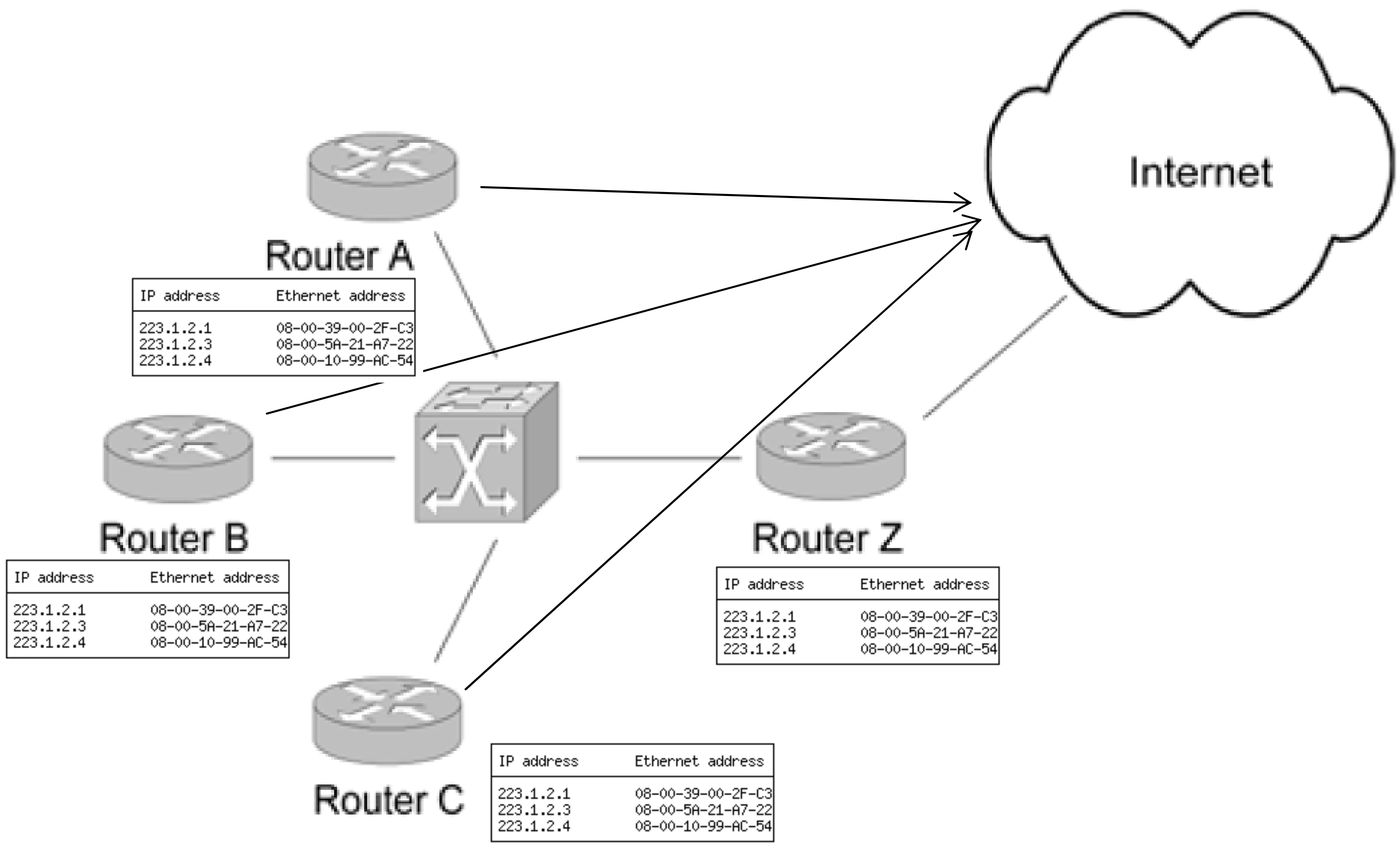
Complexity: Example



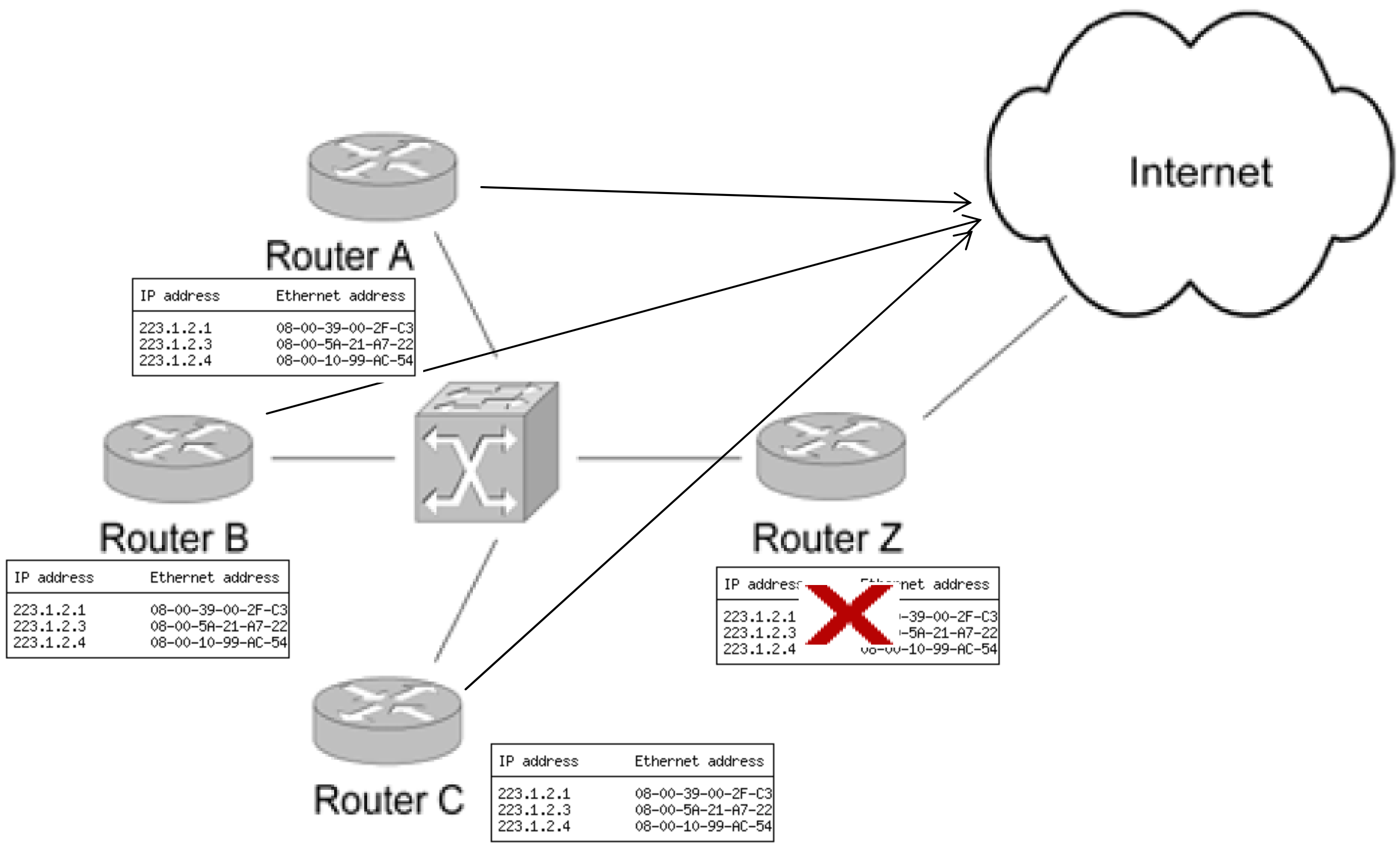
Complexity: Example



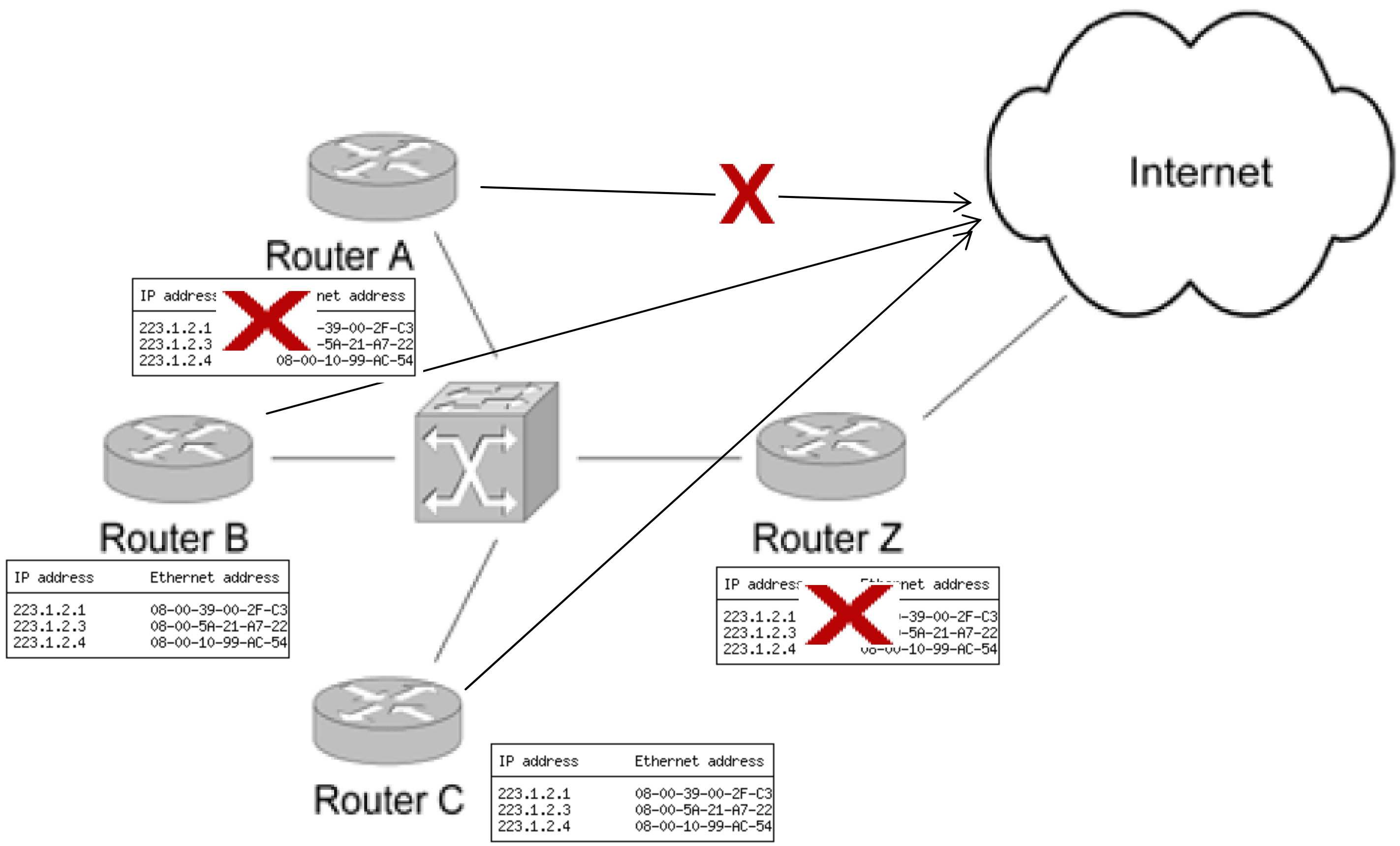
Complexity: Example



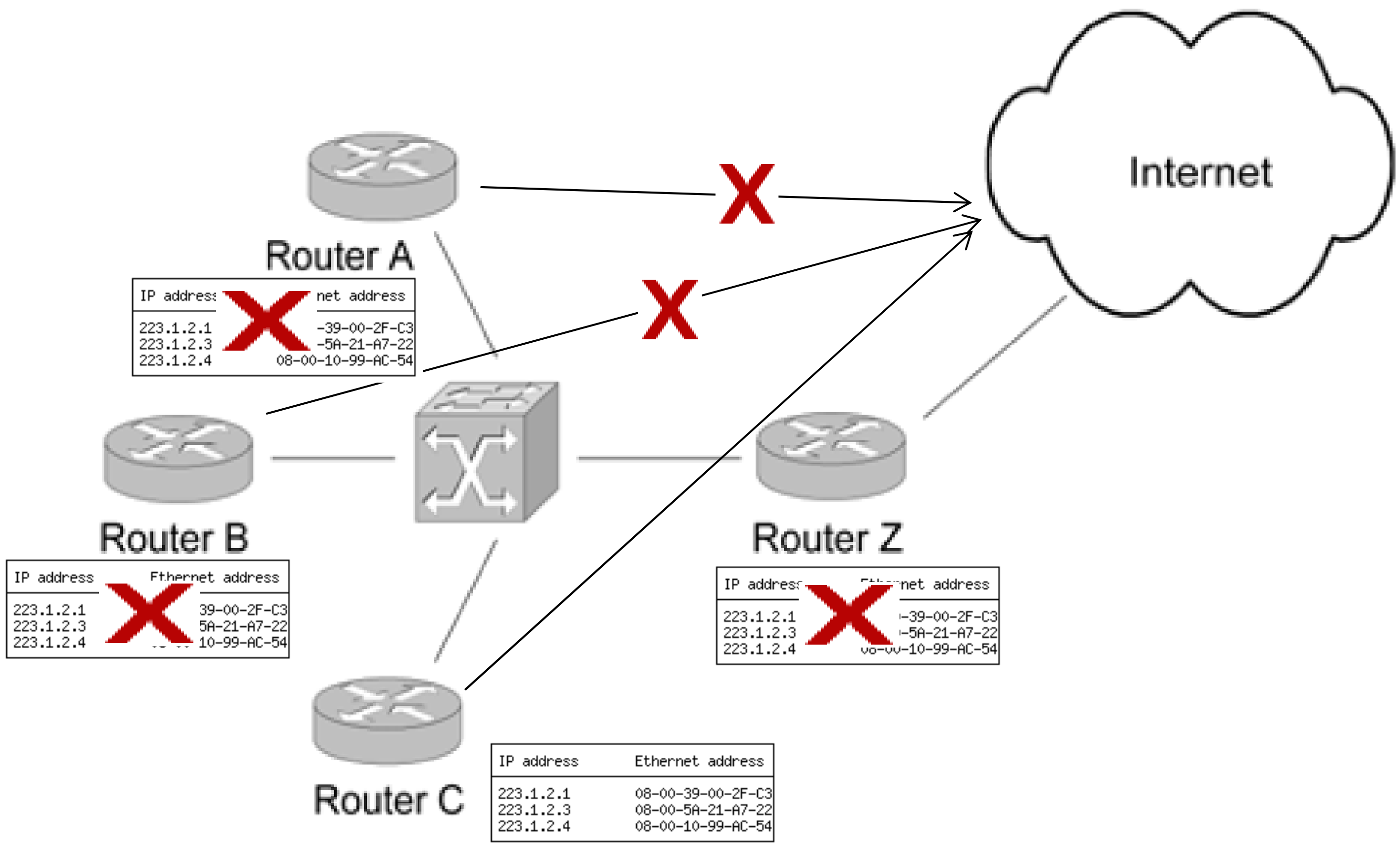
Complexity: Example



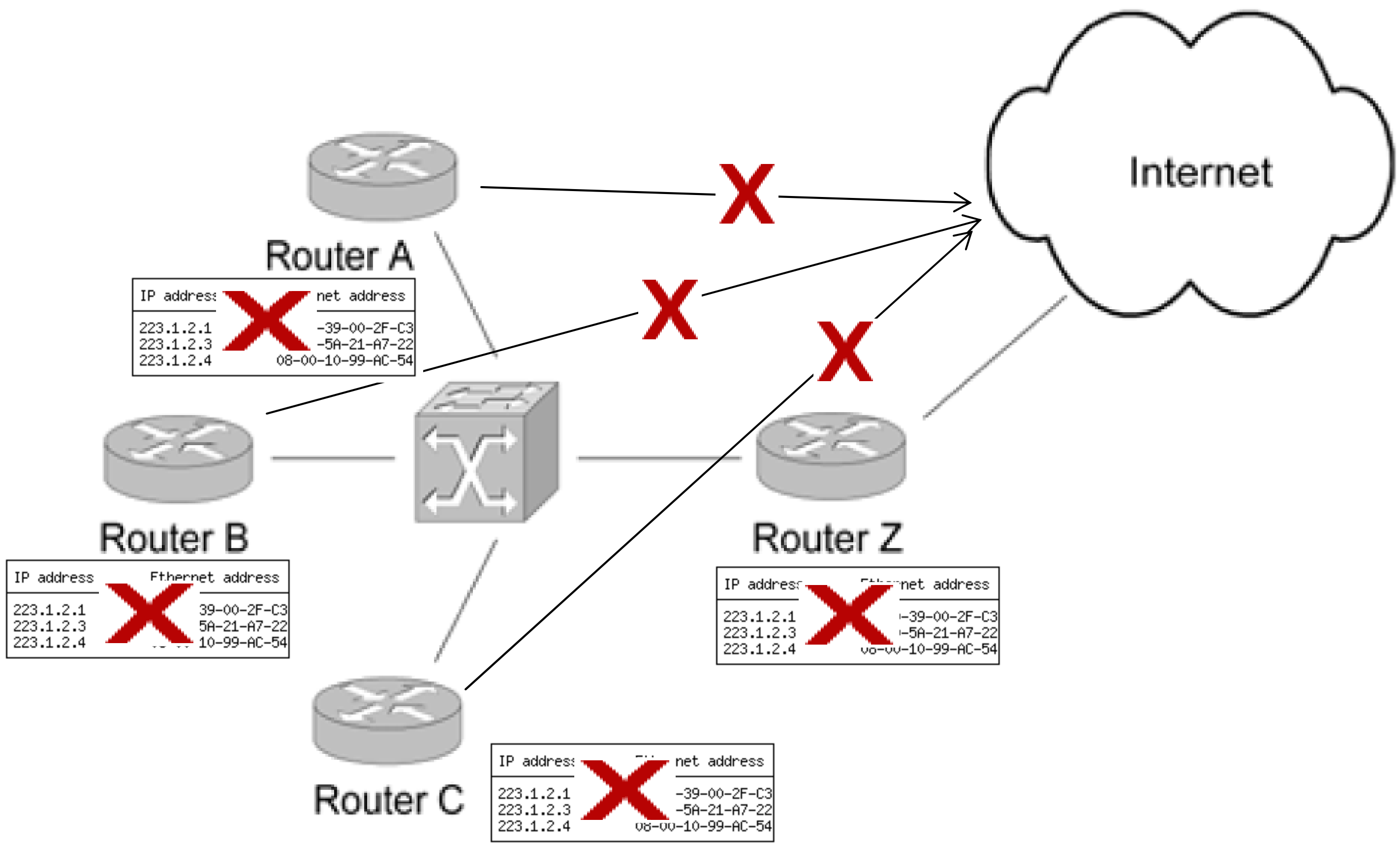
Complexity: Example



Complexity: Example



Complexity: Example



Complexity: Example

- But 2 months later:
  - The same problem
  - Conclusion: ARP failure due to a route processor crash triggered by a bug in the code which handles multicast packets
- The next 12 months:
  - 10 serious bugs on 5 hardware platforms
  - Took engineer weeks to solve them

Complexity: Example

- Final conclusions:
  - Implementing IP Multicast requires a substantial commitment of engineering resources
  - It impairs the stability of unicast routing because frequent OS upgrades and intrusive testing is necessary
  - „IP multicast defines a limit-case for deployable complexity in today‘s internet“

Problem 2: Extra-technological factors

- Finite staff time for troubleshooting
- Scarcity of debugging tools
- Limited skill-set of operational engineers
- Lack of trust between neighboring domains

What protocol designers do…

Problem 3: Ignored functional constraints

- Economic forces
- Historical forces
- Institutional forces

- Questions to be asked:
  - What does my currently stable network have to gain from enabling the new technology?
  - Can I debug without impacting best-effort service?
  - Are there benefits sufficiently compelling to compensate for the potential pain?
  - When it breaks will I be blamed?

Problem 4: Timeliness

- Network researchers only take practical steps when a problem is already there
- Then it takes too much time
- Not enough cost is spent in QOS if there is no obvious problem

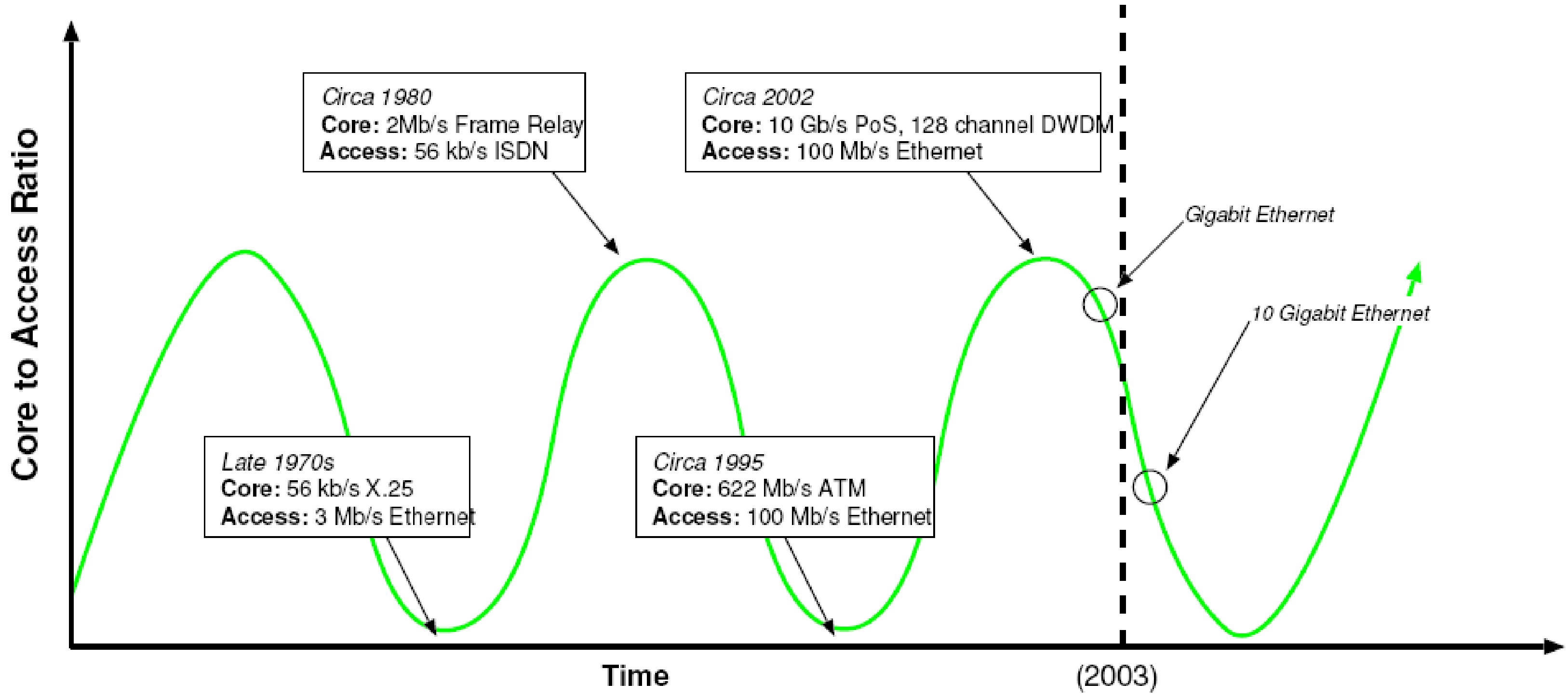
Problem 5: Inherence to the network

- The mechanisms that shall provide the guarantees have to be researched and engineered before the network is deployed
- Even if they are not needed at the moment
- This is done in security systems, so why not here?

The solution which was always taken: „Throwing bandwith at the problem“

- It avoids introducing new failures → no risk!
- Can solve problems like:
  - Latency
  - Jitter
  - Loss
- More bandwith is a good thing anyway

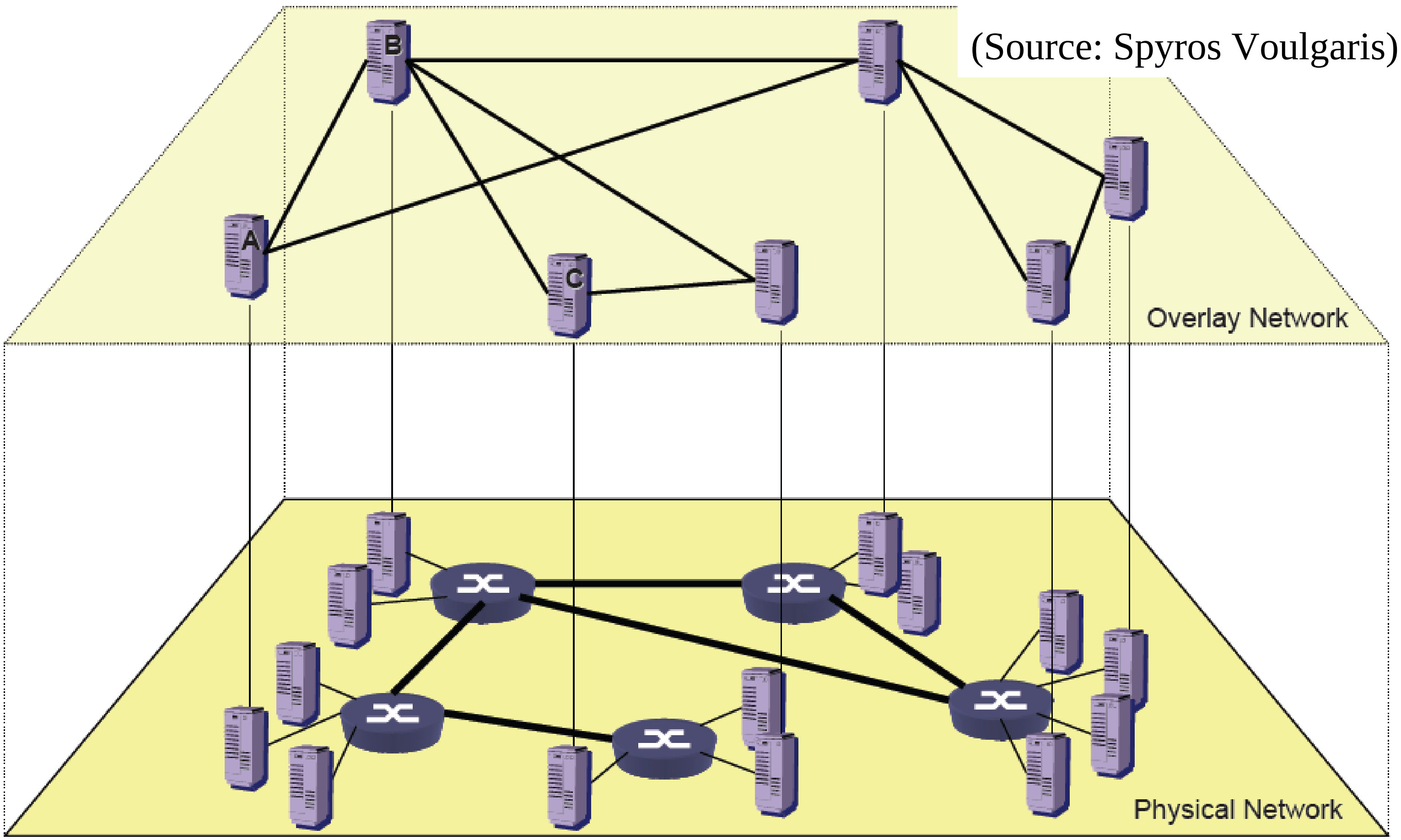
Relative core to access bandwith



Outline

- Definition QOS
- Attempts and problems in the past (2 Papers)
- A possible solution for the future:
  Overlay networks (2 Papers)
- PlanetLab
- Personal opinion

Structure of an overlay network



(Source: Spyros Voulgaris)

The idea of overlay networks is not new

This approach has drawbacks

- Works only if the application used by many clients
  - Impossible to do reasearch studies into managing overlay networks
- Modifications very difficult
  - Learning from experience does not help much
- Security
  - All defined in the individual application

Goal: Use the overlay as a research testbed AND as a deployment platform

- Research testbed
  - Researchers have access to a large set of geographically distributed machines
  - A realistic network substrate that experiences congestion, failures, and diverse link behaviors
  - The potential for a realistic client workload
- Deployment platform
  - Researchers have a direct technology transfer path for popular new services
  - Users have access to those new services

Testbeds already used: Physical testbeds

- Production testbeds
  - Example: Internet2
  - Support real traffic from real users
  - Problem: Have to be very conservative in experimentation: → only little incremental changes are possible
- Research testbeds
  - Problem: Lack of real user traffic

Testbeds already used: Overlays

- Mainly used for deploying fixes for specific problems
  - ABONE: Focused on supporting extensibility of the network forwarding direction
  - XBONE: Limited to IP-in-IP tunneling

Physical dimension

- Large amount of nodes (1000s)
- Most of the sites are single nodes connecting many clients to to the overlay
- Nodes should differ from each other
  - Different link behaviour
  - Geographically distributed
- About 100 sites should have much computing resources at network crossroads

Software components (1)

- Virtual Machine Monitor (VMM)
  - Runs on each node
  - Defines an interface to abstract resources for services distributed over the testbed

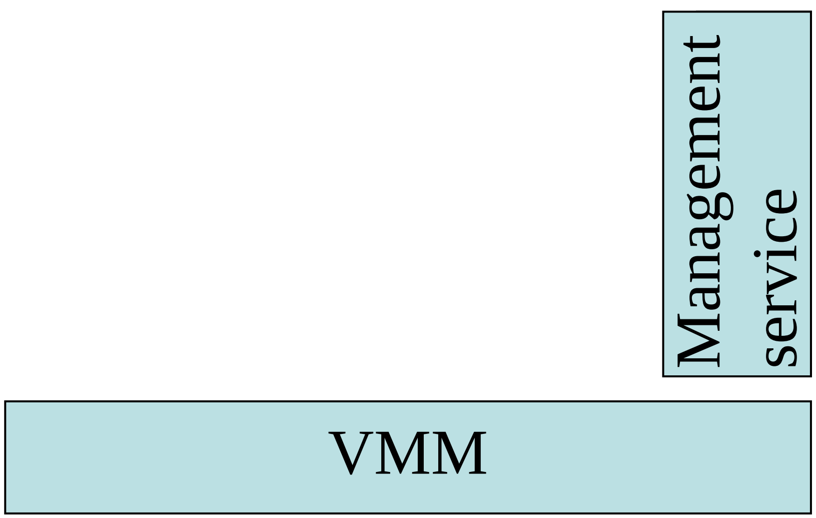
Software components (2)

- Management service
  - Controls the testbed

    - Discover the set of nodes in the overlay

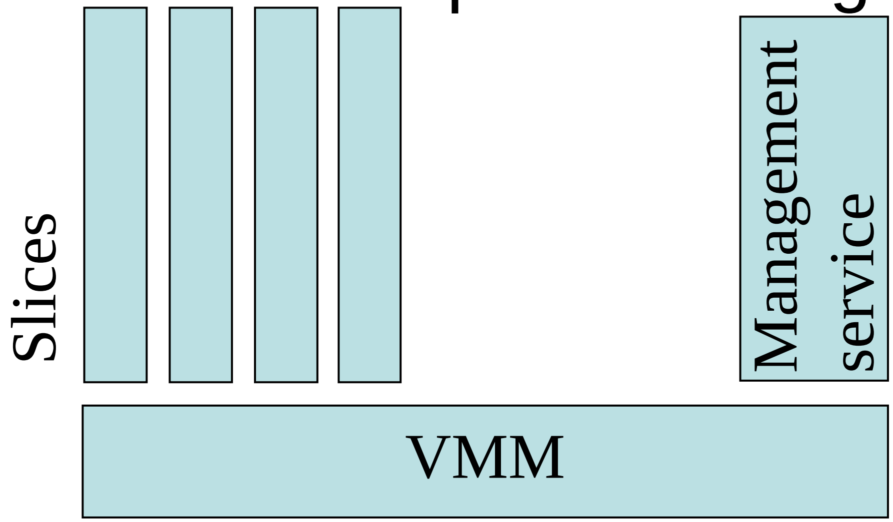
    - Monitor their health

    - Keep the software running on them up to date



Principle 1: Slice-ability

- Each application should run in a slice of the overlay
- Each node has to multiplex multiple services
- Slicing can be characterized on how these nodes are spread through the internet



Principle 2: Distributed control of the resources:

- Researchers
  - Install and evaluate new services
  - Decide how the service are deployed
- Clients
  - Access the services
  - Decide what services to run on their nodes
  - Should be required to allocate slices of their machines to experimentation
  - Be able to set policy on how resources are allocated to different services

Principle 3: Unbundled management of the overlay

- Several largely independent sub-services
- Running in an own slice of the internet
- For sub-services of the core system agreed-upon versions are necessary
- Other services can have different implementations where the better ones can replace older ones

Principle 4: API should promote application development

- Existing and widely adopted programming interface
- Easier access by clients
- The underlying platform can change over time the API shouldn‘t

Deployment of a service in an overlay

- A new-generation service provider performs the following steps
  - Choose a particular new architecture
  - Construct or use an overlay that supports the architecture
  - Distribute proxy-software to real users for accessing the overlay
- If the overlay is successful
  - Offer direct access to the customers
  - Offer access to the ISPs

The future

- Development of many different overlays with different characteristics at the same time possible
- This process can lead to two scenarios:
  - Uniformity
  - Synergy of dynamic diversity

Remaining problems

- Overlay builds on the underlying network
- The overlay cannot control the quality of service for packets traversing the virtual testbed
- When allocating slices on nodes it‘s not possible to ensure that a given application receives predictable network performance
- If several overlays shall coexist without an architectural chaos overlay designers must consider how to bring this union of overlays together to form a coherent framework

Outline

- Definition QOS
- Attempts and problems in the past (2 Papers)
- A possible solution for the future:
  Overlay networks (2 Papers)
- ☞ PlanetLab
- Personal opinion

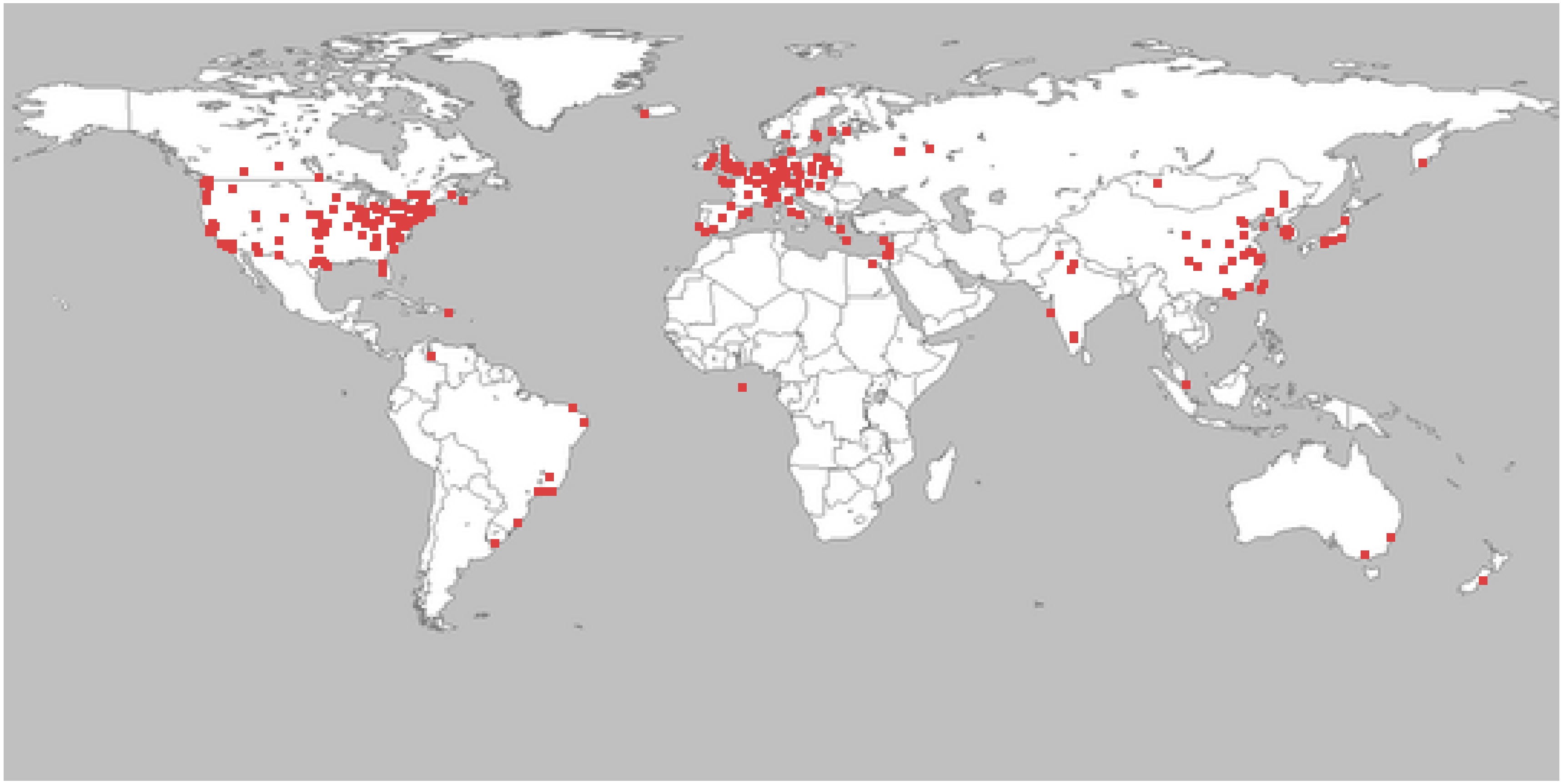
Phases

- Seed phase
  - 100 machines
  - Pure testbed
  - Functionality for a small known set of researchers
- Researcher as clients
  - Increasing the number of nodes up to 1000 sites
  - Users are primarily researchers experimenting with their services and other primitive services provided
- Attracting real clients
  - Spinning off of physically distinct copies of PlanetLab

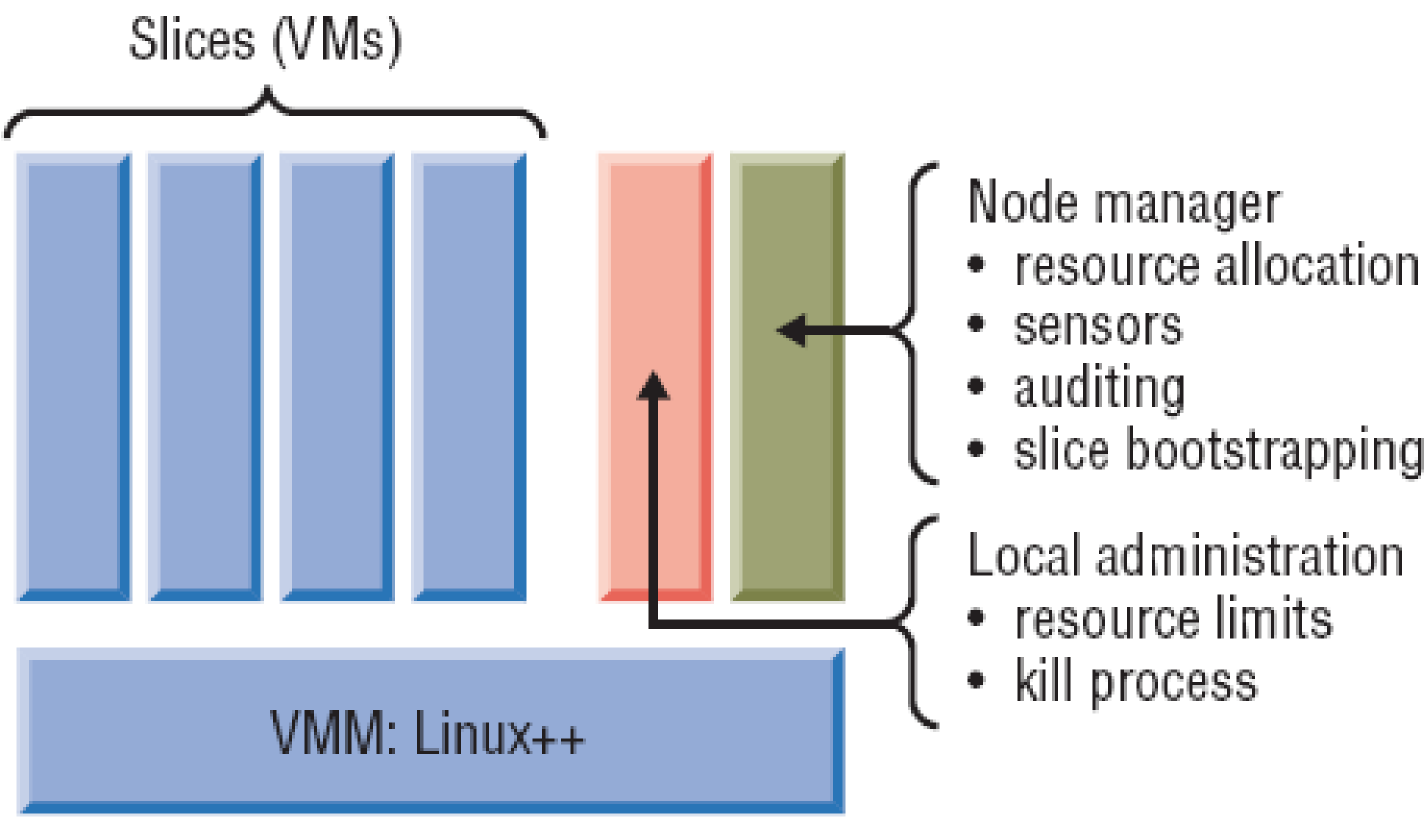
Current state

- Consists currently of 840 nodes
- Used by more than 1000 researchers
- Getting an account in not easy
- New technologies developed for
  - distributed storage
  - network mapping
  - peer-to-peer systems
  - distributed hash tables
  - query processing

Distribution of the nodes



Node architecture



Outline

- Definition QOS
- Attempts and problems in the past (2 Papers)
- A possible solution for the future:
  Overlay networks (2 Papers)
- PlanetLab
- Personal opinion

Personal opinion: Attempts and problems in the past

- Old approaches for enabling QOS have absolutely failed in most cases
  - I see the main problem in agreements of different ISPs and a conservative community
  - Complexity seems to me just to be an excuse

Personal opinion: Overlay networks

- Overlay networks like PlanetLab could really bring some changes to the internet because the services can evolve over time and get tested by real users
- Some services might really be successful and have a certain user community
- I doubt that the fundamental architecture of the internet will change
- It will still take a lot of time

Personal opinion: Do we need „something new"?

- If I would have been asked some weeks ago:
  - „I'm not unhappy with it because I think it „works""
- However there are for sure some advantages that could be achieved by QOS
- I guess if we had them already no one would like to miss them again

Where are we? Where can we go?

Questions?

References

- **Failure to thrive: QoS and the culture of operational networking**
  - Bell, G.,
  - August 2003
  - http://doi.acm.org/10.1145/944592.944595
- **QoS's downfall: at the bottom, or not at all!**
  - Crowcroft, J., Hand, S., Mortier, R., Roscoe, T., Warfield, A.
  - August 2003
  - http://doi.acm.org/10.1145/944592.944594
- **A blueprint for introducing disruptive technology into the Internet**
  - Peterson, L., Anderson, T., Culler, D., Roscoe, T.
  - January 2007
  - http://doi.acm.org/10.1145/774763.774772
- **Overcoming the Internet impasse through virtualization**
  - Anderson, T.; Peterson, L.; Shenker, S.; Turner
  - April 2005
  - http://ieeexplore.ieee.org/iel5/2/30759/01432642.pdf