Today’s Agenda

- Summary of the e2e arguments
  - Pros
  - Cons
- Some common assumptions in networking
- Problems faced by today’s Internet
  - Social
  - Technical
- Specific examples of new requirements
- Some initial proposals for re-engineering the net

E2E arguments: Summary

- Specific application-level functions usually cannot and preferably should not be built into the lower levels of the system (the network core)

- Sometimes an *incomplete* version of the function provided by the communication system may be useful as a *performance enhancement*

- This philosophy is central in today Internet's design:
  - Functionalities are moved up and out of the core
  - Lead to KISS principle in system design (or vice versa)
  - Lead to Simplicity principle (a kind of Occam's razor)

E2E arguments: Pros

- Evolvability:
  - No central authority imposing what kinds of applications can be developed
  - Easier to maintain backward and forward compatibility
  - “Simple” network layer makes it easier for IP to spread
- Cost befenit:
  - Applications that don’t need a particular feature do not have to pay the price (Turn this argument around?)
- Flexibility, Adaptability, Simplicity
  - Is it really?
- Easier to model, describe, and predict
- Philosophically pleasing (liberalism, e.g.)

E2E arguments: Cons (only a subset)

- It has been difficult to follow the philosophy
  - NAT, Firewall, Web caching
  - Design decisions (which are the ends, which is "completely and correctly implemented", ...) are sometimes based on trust, responsibility or performance instead of E2E [*e.g., why is reliable transport not in app. layer? We can also do source routing, or congestion control*]
  - Performance implications are not justified by E2E
- New applications have been flourishing, but mostly those sensitive to the E2E design approach
- The "ends" may not be trust-worthy, and may be stupid (less sophisticated users)
  - Spams, DoS, Viruses, Worms, ...

Common Assumptions about the Internet

- IP dominates global communications
- Packet switching is more efficient than circuit switching
- Packet switching is robust
- IP (and PS) is simpler
- QoS can be realized over IP

IP Dominates Global Communications?

- [US-census 2002] **Revenues**: Satellite Telecom (5.7B), ISPs (18.7B), Radio/TV broadcast (48.5B), Cable Distribution (77.7B), Cellular & other wireless Telecom (96.5B), Wired telecom-carriers (237.6B).
- [Nielsen/NetRatings survey 2004 & others] **Percentage of US households having access**: Internet (75%), Cable/Pay TV (78%), TV (98%)
- [RHK Industry Reports 2002] **Public Telecom Infrastructure Expenditures**: Core routers (1.7B), Edge routers (2.4B), SONET/SDH/WDM (28.0B), Telecom Multi-Service Switches (4.5B)

PS is more efficient than CS?

- More efficient means better utilized (both in transmission lines and switching equipments)
- True for networks with scarce bandwidths
- However, does it really matter today?
  - Average utilization levels
    - ATT switched voice (33%), Internet backbones (15%)
    - Private lines networks (3-5%), LANs (1%)
  - Various Reasons
    - Internet traffic is asymmetric and bursty, links are symmetric
    - Operators tend to over-provision because PS networks behave very badly once congested (oscillation, routing loops, black holes, disconnections, etc)
    - Over-provision to ensure low delay (satisfy customers), it's more economical to add capacity in large increments

PS is more robust than CS?

- Downtime per year:
  - Internet: 471min [Labovitz et al. 2000]
  - Phone networks: 5min [Kuhn 1997]
- Recover time
  - Internet: median 3min, frequently > 15min (due to slow BGP convergence time)
  - SONET/SDH rings: < 50ms (via pre-computed backup paths)
- Routing in the Internet
  - Routing info affected by user traffic, suffering from congestion (in-band routing)
  - Routing computation complex → overload processors
  - Probability of mis-configuring a router is high, one router's error affect the whole network

IP (and PS) is simpler?

- Number of lines of codes in
  - Typical Tel. Switches: 3 millions, extremely complex switch: 16M
  - Cisco's IOS: 8 millions [more susceptible to attacks]
- Router crashes frequently, takes long time to reboot
- Hardware
  - A line card of a router: OC192 POS has 30M gates + 1 CPU + 300MB packet buffers + 2MB forwarding table + 10MB other state memory
  - Current trend makes routers more complex (multicast, QoS, access control, security, VPN, etc) – violation of E2E
  - A line card of a typical transport switch: ¼ number of gates, no CPU, no forwarding table, one on-chip state memory
- Density: highest transport switch capacity = 4 x highest router capacity, at 1/3 the price
  - WDM, DWDM push the difference further
- IP's "simplicity" does not scale!

QoS can be realized over IP?

- Belief: over-provisioning allows low e2e delay → guaranteeing QoS is possible
- After > 10 years of research, IntServ and DiffServ are still not good enough.

- Few financial incentive to provide QoS over IP
  - Watch out for VoIP, however.
  - On the other hand, current phone services are much better with very low price

Other measures

- Scalability
  - CS scales more or less linearly
  - When data rates increase, routers can’t keep up
- Flexibility
  - IP is more flexible
  - Lead to high costs of end-systems
  - Need more sophisticated users [large organizations need a room of sys admin, just 1 phone operator

Problems Faced by Today's Internet (1)

1. Untrustworthy world
   - End points can't be trusted
   - Spam, viruses, worms, DoS, ...
2. More demanding applications
   - Best effort can't support MM apps
   - Might be possible (IntServ, DiffServ) but ISPs won't cooperate
3. ISP service differentiation
   - ISPs do not want to collaborate to allow E2E implementation, they want ISP-specific services
   - Lead to closed islands of enhanced services

Problems Faced by Today's Internet (2)

3. Rise of third-party involvement
   - Officials of organizations (corporate networks, ISPs, ...)
   - Officials of governments (China, Vietnam, ...): law enforcement, political censorship, public safety, ...
4. Less sophisticated users
   - Installation, configuration, upgrades, maintenance of complex end-system softwares require experts
   - End users want ease of use
   - Other dumb devices join the net (PDAs, sensors, watches, refrigerators, ...)
5. Many more network types
   - Sensors, PDAs, other devices
   - Inter-planetary networks (DTN, e.g.)

Examples of new requirements

- Users communicate but don’t trust each other
  - Two parties want to negotiate a binding contract
  - Authentication
  - Communication with anonymity
- End parties do not trust their own hardware, softwares
- The ends vs. the middle
  - Third party gets in the way of communications
  - E.g, should “traffic analysis” be allowed? How about firewalls? How about government reading your emails?
- Solving problems of spam, worms, phishing, ...
- Multiway communications

Some Technical/Non-technical Solutions

- More functionalities in the end nodes
  - Personal firewalls, filtering softwares
  - E2E smart MM applications (Real, WMP)
  - Use trusted third parties, more cryptographic communications (PGP and others)
- Adding functions to the core (deeply violate E2E)
  - Firewalls & other traffic filters
  - NAT elements
- Laws in cyberspace

Proposals for re-engineering the Internet

- Add a knowledge plane [Clark et al, 2003]
- Plutarch: network pluralism [Crowcroft et al, 2003]
- Role-based architecture [Braden et al, 2002]
- Triad Project [Stanford]
- ...

- Your proposal?