

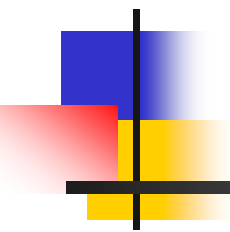
CS 5229

ADVANCED COMPUTER

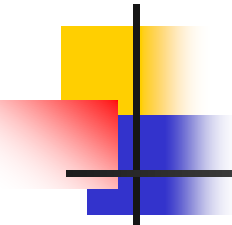
NETWORKS

2010/2011 Semester 1

Dr. Chan Mun Choon
School of Computing
National University of Singapore



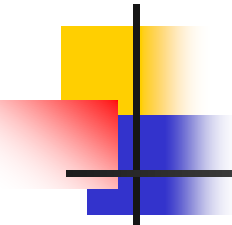
Network Architecture / Design



Reading Lists

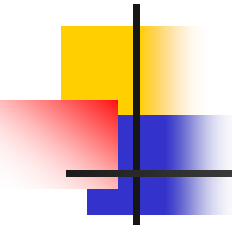
- [1] D. D. Clark, "The Design Philosophy of the DARPA Internet Protocols," ACM CCR Vol. 18, No, 4, Aug 1988, pp. 106-114.
- [2] J.H.Saltzer, et.al, "End-to-end arguments in system design," ACM Transactions on Computer Systems (TOCS), Volume 2 , Issue 4, Pages: 277 – 288, 1984

- D. Isenberg, [The Rise of the Stupid Network](#), Computer Telephony, August 1997, pg 16-26.



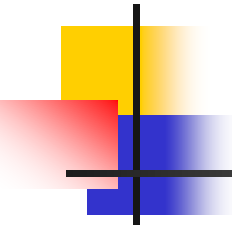
Before the Internet ...

- PSTN or the Public Switched Telephone Network, is a circuit-switched network that is used primarily for voice communications
- For much of the 20th century, the PSTN was the only bearer network available for telephony, until the arrival of VoIP
- Implications:
 - Voice traffic generates most of the traffic, so the network is designed to be optimized for voice
- PSTN implementation:
 - Circuit switching
 - Dumb device, smart network



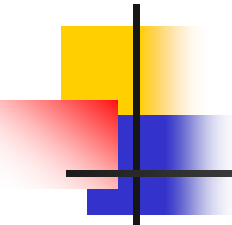
The “Smart” PSTN

- PSTN’s Intelligent Network
 - Customization in the network
 - Allow equipment from different vendors to inter-work
 - Focus on (voice) call completion, automation, and billing, e.g. call forwarding, calling card, etc.
 - Any change (new service) needs to go through a very LONG process before it can be deployed



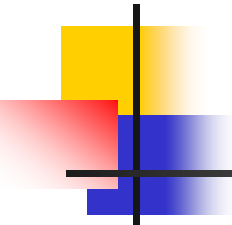
An example of the problem with “Smart” PSTN

- How to improve voice quality in PSTN
 - “Easy” way: increase voice sampling rate and/or change the coding algorithm
 - But the 64Kbps rate is designed into the network and many components need to be modified
 - Even small changes are difficult because many of the “intelligence” built into the system relies on certain assumptions how other components work
 - changing one component can cause some others to fail
- A design where the network/transport does not make any assumption on the traffic will make changes much easier to implement



With the Internet ...

- Data traffic volume has overtaken voice traffic ...
- Emergence of packet switching and availability of different access network technologies, e.g. cable modem, ethernet, WiFi etc.



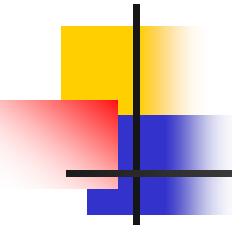
The “Stupid” Network

- Model: smart device, dumb network
- The best example: Internet
 - Infrastructure/bandwidth is (relatively) cheap and plentiful
 - Always on
 - Network simply deliver the bits, no processing
 - May allow a small amount of customization like different classes of delivery (lower delay, lower loss etc.)
 - Most, if not all customization/application-specific processing are performed at the end-points



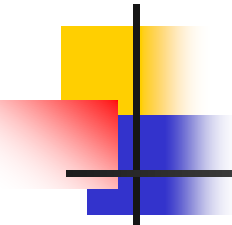
Design Philosophy of Internet Protocols

- Fundamental Goal:
 - Sharing of resources inter-connected network efficiently
- Second Level Goals (in order of importance)
 1. Communication must continue despite loss of networks/gateways
 2. Support multiple types of communication services
 3. Accommodate a variety of networks
 4. Permit distributed management of its resources
 5. Cost effective
 6. Allow host attachment with minimum effort
 7. Resources must be accountable



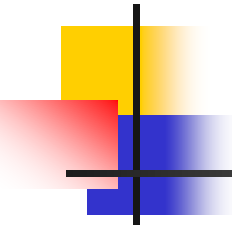
1. Survivability

- Two entities can continue to communicate unless there was no physical path over which any sort of communication could be achieved
- To achieve this goal, state information must be kept somewhere in the network
 1. Stored state in the network and uses replication for recovery
 2. Stored state in the endpoints
- Which one is preferred?



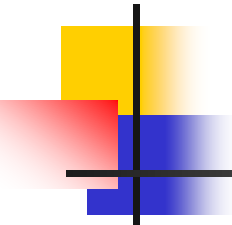
Cont'd

- Option 2, also called fate-sharing because if the state and the endpoint fail at the same time, is preferred because
 1. More robust against intermediate failures
 2. Much easier to engineer than replication
- Fate-Sharing approach
 - Network is stateless, or datagram network
 - More trust is placed on the endpoint machines than the network to ensure reliability



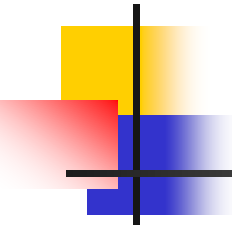
2. Types of Service

- The Internet protocol (IP) is based on a connectionless or datagram mode of service
- The reliability associated with the delivery of a datagram was not guaranteed, but "best effort"
- Type of service is supported at the transport level
 - Distinguished by differing requirements on throughput, delay and loss
- TCP was the first service provided in the Internet and was supposed to be general enough to support any type of service
 - TCP ensures reliable, in-order delivery



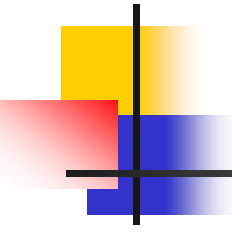
Cont'd

- First protocol outside the range of TCP was XNET
 - A cross-Internet debugger
 - The network is not working “correctly”, so XNET must work with unreliable delivery
- Another application is the real time delivery of digitized speech
 - One of major source of delay is the error recovery mechanism
 - Need to split TCP and IP, which was originally a single protocol, into 2 separate protocols. UDP was added
- Supporting multiple services proves more difficult than initially thought as new requirements are considered
 - Luckily, **reliable delivery is not designed into the network**



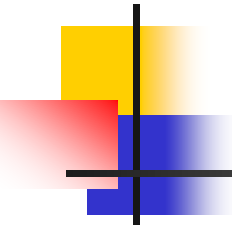
3. Varieties of Networks

- In order to allow the flexibility of connecting of various networks, only a minimum set of assumptions is made about the networks to be connected
 - Network can transport datagram
 - Packet size must be within some reasonable range
 - Requires reasonable low error rate but not perfect reliability
 - Understanding some suitable form of addressing



Cont'd

- What is missing?
 - Reliable or in-order delivery
 - Broadcast/multicast
 - Priority/differentiation (or in modern terms QoS)
 - Network state information like failure, delay, link utilization etc.
- Customization of service at the transport level
 - Is it possible to make up for these missing features at the end-points?



Other Goals

5. Cost effective

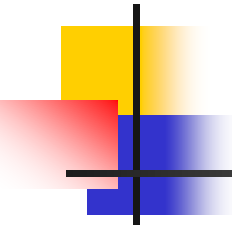
- Not as cost effective as a more specialized environment designed for a specific service
- Header is large (40 bytes for TCP)
- End-to-end retransmission can be expensive

6. Allow host attachment with minimum effort

- Requires end host to follow certain rules (think socket programming)

7. Resources must be accountable

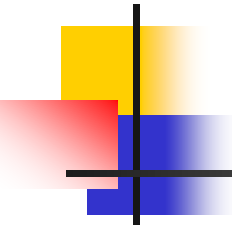
- Few tools/mechanisms



End-to-end Argument

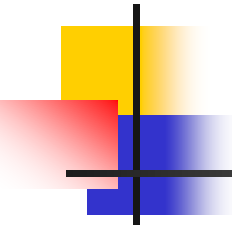
- A function or service should be carried out within a (network) layer only if it is needed by all clients of that layer, and it can be completely implemented in that layer
- This is one of the design principles guiding the “**stupid**” network approach of the Internet
 - Minimize lower layer functions, support the widest possible variety of services, and let the application adapt

- Walter Willinger and John Doyle, [Robustness and the Internet: Design and evolution](#), Robust design: a repertoire of biological, ecological, and engineering case, Oxford University Press, 2005.
 - Some interesting milestones and issues on how the Internet and protocols used have evolved



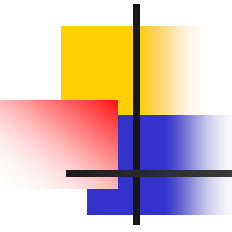
Improvements to TCP

- Congestion collapse observed in Oct 1986
- RFC-1122 lists the requirements for a conformant implementation of TCP in 1989



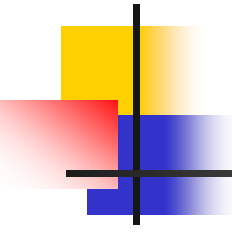
HTTP/1.0 and TCP

- Interaction between HTTP/1.0 and TCP
 - Known in 1994, addressed by HTTP/1.1 (RFC2616, 1999)
- The “World Wide Wait” phenomenon



BGP Scaling

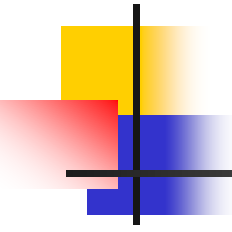
- A number of scaling problems with BGP
- One example: size of routing table
- Solution: CIDR



IP Addresses

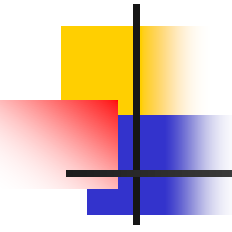
- IPv6?
- NAT?

- D. D. Clark, J. Wroclawski, K. R. Sollins, R. Braden, [Tussle in Cyberspace: Defining Tomorrow's Internet](#), ACM SIGCOMM, Aug 2002, pp. 347-356.



Tussle in Cyberspace

- The need of Internet has evolved, how best to adapt to the new (competing) requirements ?
- Many more parties involved
 - Users
 - Commercial ISPs
 - Private business user
 - Governments
 - Content Provider



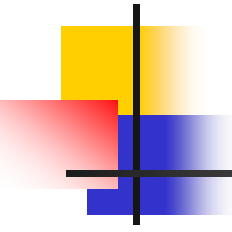
Design Principles

- Modularity

- Solutions that less efficient may provide better isolation
- BAD: DNS used to name machines and trademark. Should provide only location independent names for machines
- BETTER: Use explicit ToS bits to select QoS rather than say bind the decision to well known ports

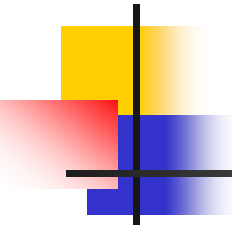
- Design for Choice

- Protocols should be designed such that all the parties to an interaction have the ability to express preferences without breaking the application
- Example: E-Mail
 - Choice of email reader
 - Choice of SMTP or POP protocols
 - Choice of email (SMTP/POP) servers



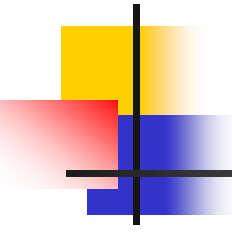
Control

- Who should be in control of what? Example
 - Provider owns the IP addresses
 - Users cannot keep the same IP address when they change ISP – “no number portability”
 - User not allow to provide network-related service, e.g. host web server



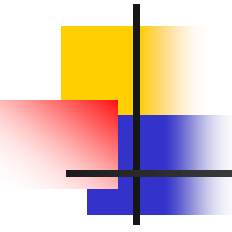
Trust

- One of the most profound and irreversible changes in the Internet is that users do not or cannot trust each other
 - Lost of trust calls for less transparency, and we get firewalls, NAT etc.
 - Desire by ISP for control and application customization produces application level filtering, connection redirection, etc.
 - Desire for performance improvement leads to deployment of caches, mirrors, hacks to DNS etc.



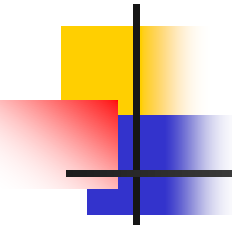
Openness

- Openness of the network is key to innovation and success of the Internet
 - Openness “encourages” competition
 - Vertical integration, bundling of many services together, requires the removal of certain forms of openness, but may be a useful service to have
 - The ability to deploy a new protocol without having to modify the inside of the network is important



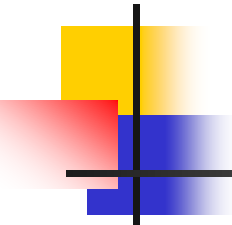
Future of end-to-end arguments

- End-to-end argument has work well so far
 - Mechanism should not be placed in the network if it can be placed at the end node
 - Core should provide general service, not one tailored to a specific application
- Innovations
 - If the core of the network must be modified to deployed a new application, innovation will be hindered
- Reliability and Robustness
 - If some application states resides in the network, the number of failure points increases. Thus, the simpler the core network, the more reliable is the application



What might happen?

- Evolution and “enhancement” of existing, mature applications is inevitable
- Failure of transparency will occur
 - Network may not be that “dumb” anymore
- Peeking is irresistible
 - The network can do something good/better if it has more information about the packet/application requirements
 - End-to-end encryption is needed to protect your privacy
- FIND?
- GENI?



HW1 Q1

- Read New Times Headlines: **Google and Verizon Near Deal on Web Pay Tiers**
- **Question:**
 - a. **What is net neutrality?**
 - b. **Do you think net neutrality is a good thing? Explain your point of view.**