Lecture : 3{Part I} (Date : 10/07/2002)

Mausam

9th October 2002

What were the ideas from the 90's which have the potential to change one's life in a significant way?

Ans: Some of such ideas are

- Self-similarity (#items people are interested in is very very large. The popularity-rank graph has a long tail. This has effect on caching. It will be discussed in some later lectures)
- GUI for Wide-Area-Networks. This is important for the web to be useful.
- Integrated DRAM and logic.
- Cooperative software like Linux.
- Pervasive Computing.

Does internet today have dynamic virtual circuits? What are the arguments for and against it?

Ans: Internet as is today, does not have dynamic virtual circuits. The gateways assume minimal support from the layers below which is the ability to forward packets, nothing more. We could instead think of Internet++, in which gateways would assume the functionality of DVC, which would be used to control the bandwidth, congestion and routing. Clark discusses the following arguments against DVC:

- Currently the routers are simple, and hence use less CPU. This would be affected if we have DVC.
- Interoperability requires the use of lowest common denominators for networking. It will be hard to incorporate all this functionality in all the existing networks.
- Currently internet uses fate-sharing principle. What happens if there is a failure in the middle of a DVC somewhere. Reliability could thus be affected.

- Clark would argue today, that it will be detrimental to freedom of application innovation. (Today, one doesn't assume anything from the network and everything gets done at the end host. Hence one can develop suitably as per one's requirements)

The arguments in favour of DVC's are :

- The end hosts become extremely complex. Overall efficiency would improve if some more work is shared by the network layer.
- Any application that requires more than a minimal support from the network (like QoS), can't be implemented in the existing scenario.
- Currently internet has an assumption that all hosts are trusted. If one wants to safeguard from the hosts then we would need to do something at the network level which is not possible because of the assumption of minimal functionality.
- It would be easier to measure/debug/diagnose errors in the network.

What are the important design principles for the internet? Give examples.

Ans: The important design principles are universality, robustness, scalability, security and flexibility. Some examples of how they are (or should be) manifested are:

- Universality:
  - In internet IP is run over everything (common protocol).
  - Internet is best effort. (No QoS guarantees)
  - Universally e2e principle is used.
- Robustness:
  - KISS (Keep it simple stupid!).
  - Soft state as seen in packet switching/dynamic routing. Hence one can recover from any failure in the middle, and get back to the previous state.
  - Liberal in what you accept, conservative in what you send.
- Scalability:
  - e2e principle. This keeps local decisions local (of course due to this one loses control). So there is an implicit hierarchy in the system.
  - In multicast system, Internet lets the receivers decide so there is no congestion on senders.

    - Shared learning is exploited like in BGP routing.

- Security:
    - Minimise the circle of trust.
    - Modularising on the tussle boundaries. (different organisations compete)

- Flexibility:
    - Applications have got significant decision power.
    - Network elements should allow for new functionality (upgradeable) like active networks etc.

What were the basic ideas in the design of internetworking?

Ans: Internetworking is the connection of different packet switching networks with each other. The following are the ideas that were used to successfully complete this process:

- Layer of interaction between different networks through the use of gateways.
- Global addressing scheme.
- Standard format of the packets.
- Fragmentation of packets if going into a network whose maximum allowed packet size is less than the size of current packet.
- Error detection done at the end host.

What are the characteristics of the gateways?

Ans: Gateways do the following things:

- They reformat packets to suit the next network. That is, they add MAC layer, and the local headers in the packet after removing the old local headers.
- They forward packets to the appropriate host.
- The fragment as and when necessary.
- They optionally buffer the packets.
- They do some part of failure detection being aware of the connectivity with the other gateways.

Gateways don't do the following things:

- They don't reassemble the different fragments.
- They don't do any error detection. (Actually more and more link layer level error detection has been happening these days. In fact, the interface on the input to the gateways do some error detection)

Can some gateway, which sends packet in some legacy network not understanding IP , remove the IP header from the packet?

Ans: No. For the local network, IP header+data is taken as the data and the network operates only with its MAC header. So the input gateway won't remove the IP header. It will only add the local MAC header and the data would be successfully transmitted further.