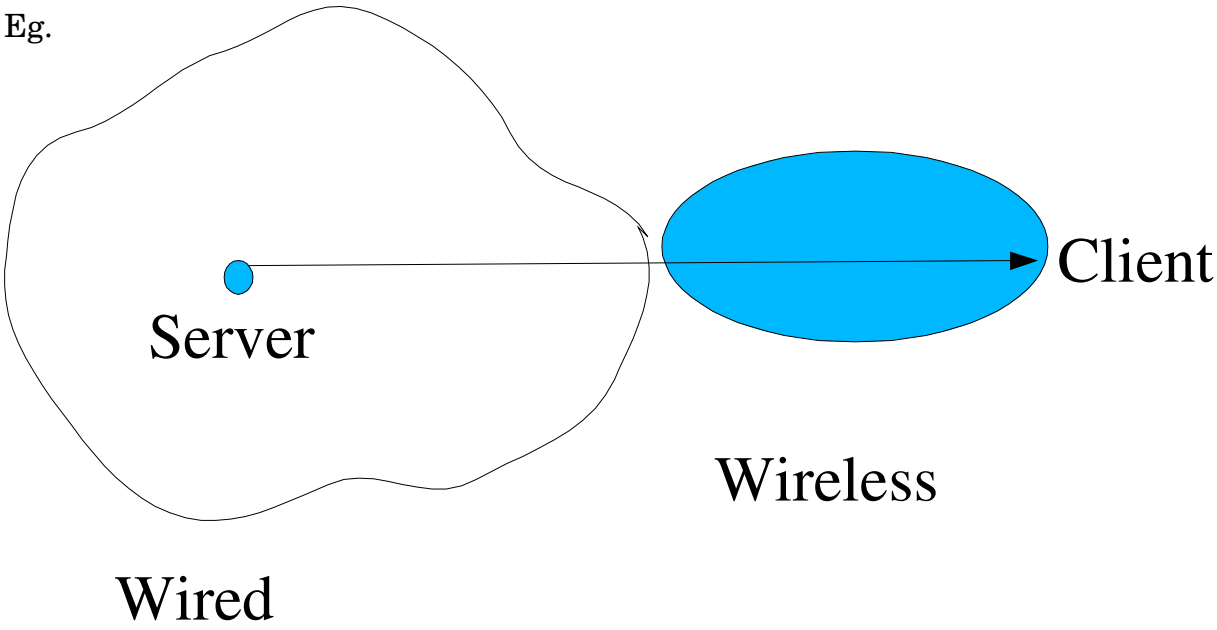
A LITTLE DIGRESSION INTO SOME INTERNET STATISTICS

Though there's a high rise in the number of cellphone users in India in the last five years, still India is way behind the advanced countries in terms of absolute growth.
All the growth in India has occured mainly in the metropolitan pockets, the rural areas are still devoid of any cellphone, Internet luxuries as yet. Most of the villages in India have only STD lines as a source of fast communication.As Census shows, 74% of Indian population inhabits villages hence it has obviously led to sparse distribution of cellphones in India.The main reason behind this lag is that most of the technologies used in India are not cost effective.To make a technology popular one has to really consider its economic aspect rather than spectral/bandwidth efficiency.
IIT Kanpur is working in this direction in the rural areas lying between Kanpur and Lucknow, using 802.11b. 802.11b can be made cost effective by mass production. IIT Chennai is also working on wireless in Local Loop technology to make
communications cost effective.

END TO END DESIGN PRINCIPLES

Question : What function in what layers?
When we design a system into layers built one on the top of other, we have to make a choice as to what function will be performed in what layer.
The End To End Design Principle is a guide to determine what fuctionality should go in what layer/module in the system.

Eg.



TCP doesn't perform well on wireless networks as it was designed basically for wired networks. To improve performance a large number of optimizations have been done in the network layer.

EXAMPLES OF LAYERING

Layering in an OS Kernel

SHELL
KERNEL
DEVICE DRIVERS
HARDWARE

Layering in a Microprocessor

COMPILER
ASSEMBLY LANGUAGE
INSTRUCTION SET
PROCESSOR

Examples of End to End Principle.

FILE TRANSFER APPLICATION

Assumptions -
1 File System reads the file with 100% reliability and copies it on the communication medium accurately.
2 Communication Network provides 100% gurantee of conveying it as it is to the other end.
3 The writing and storage of the file by the file system is again perfect.

If this is the case then we can safely assume that the file transferred will be an exact replica of the earlier one. But practically neither of the assumptions proves to be true.

Hence we have to trap some of the errors at the Application Level, so that we can ensure that the file has been transferred correctly. So we can perform a checksum of the contents of file.

[the paper in question has been written as a participating entry in the debate over enhancing the functionality of a network. It argues against it and suggests that functionality should be increased at the Application Level,

instead.]

The above example may not be a practical one presently owing to following reasons:

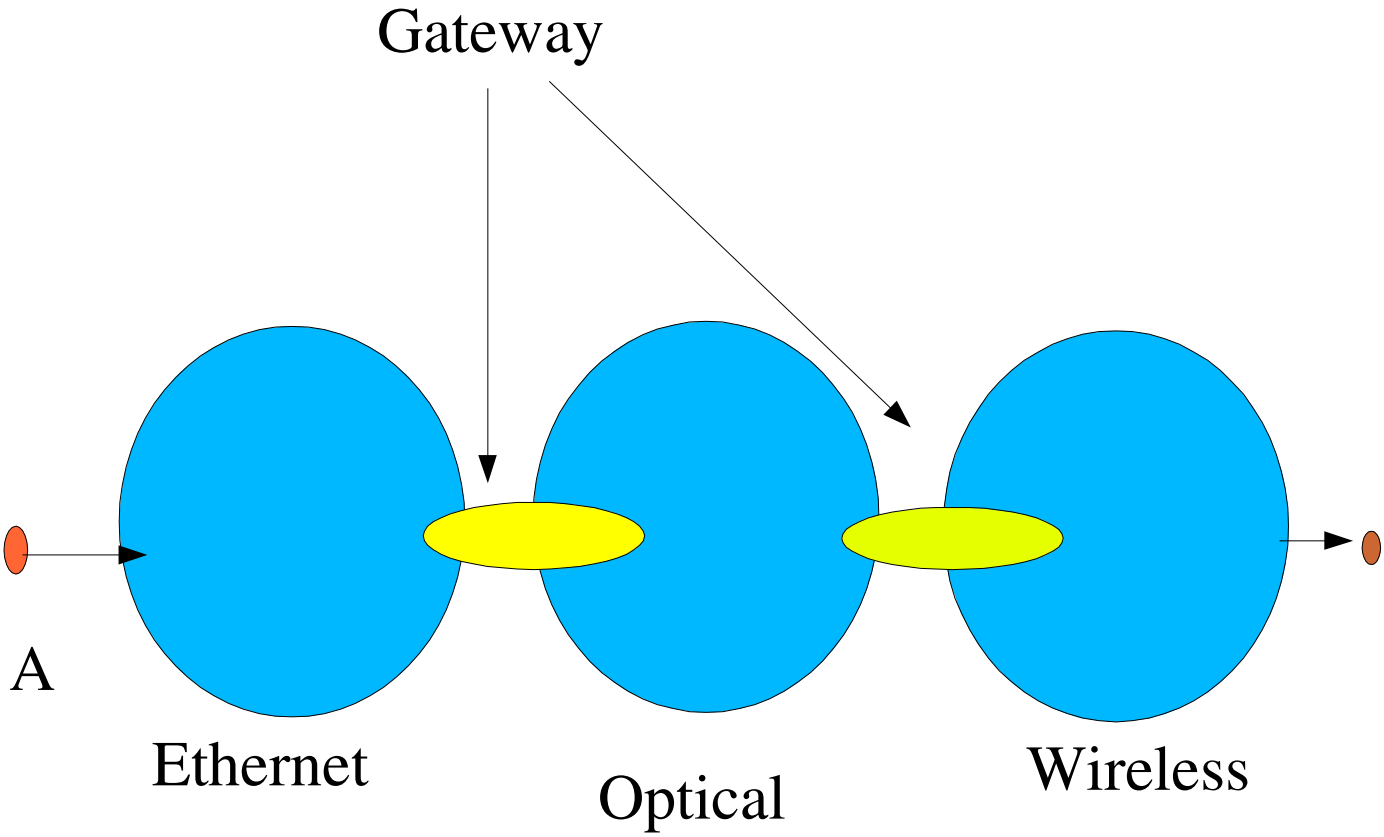
1. The Disk System is pretty reliable nowadays. It performs Checksum on the data by itself.
2. The other softwares are also reliable.

Practical Example

Since the network can't gurantee the authenticity of what one communicates through them one has to perform a checksum of a downloaded file, in order to ensure that it has the same data that he intended to download.
Eg. Linux System Files, Gnuchess.

Another Example

File Transfer Via Three Networks



Possible threats without End to End Design

1. Gateway introduces an error
2. Gateway might crash and cannot preserve the state information (information in Gateway Buffer)

Question:When to partially implement a function at a lower layer?
Answer:

1. When it is possible without much trouble.
2. When Performance improves by doing so.
3. The High Level Applications should have some use of the function in question.

CAVEATS IN LOWER LAYER IMPLEMENTATION

1. All applications have to use it even if they don't require it.(at least they have to pay for it)
2. Lower layer may not have enough information for efficient implementation.

Example:
VIDEO COMPRESSED AND SENT
Method

1. First frame contains the original picture and is referred as 'I' frame. High reliability is important for frame 'I'.
2. The frames following 'I' are differential frames referred as P/B. Not much reliability is needed for these frames.

The point is that if the 'I' frames get lost over the network or get corrupted, then all the following 'P/B' frames will become meaningless. Hence the 'I' frames should be sent with utmost care and all sorts of checks to ensure its accuracy, but the same is not required for differential frames. But the network layer has no means to know whether the data being sent is an 'I' frame or a 'P/B' frame, and hence can't adapt to these changing requirements. It is only the application which knows how to treat them differntly.

HTTP Proxy

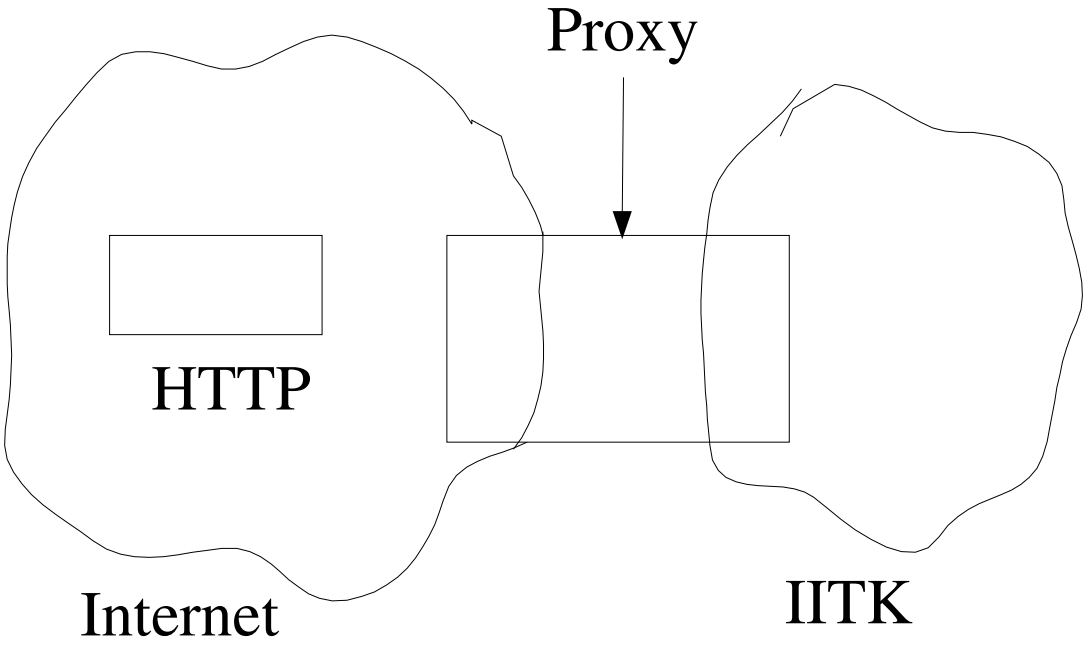


Diagram shows that a Proxy Server has been introduced between Internet and IITk network.
This Proxy Server does thye following jobs:
Provides security.
Caches frequently used data for faster access

Now suppose a user is performing a transaction thru this proxy, and after he commits the transaction, the Proxy receives acknowledgement and before sending it to the client the Proxy crashes. Now the server thinks that the transaction is complete. But since the client has not received the acknowledgement due to Proxy crashing, to him the transaction is still incomplete.
So, it can't be left over the network to ensure reliability. The applications have to develop an understanding between them to keep a log or checkpoint in such a case. The proxy server should send the client an acknowledgement after it's reovery.

RISC

Reduced Instruction Set Computers (RISC) are again example of End to End Design. In these computers a machine is made with a minimum instruction set. Now it's the job of application using it, to customize it according to their requirements. So a minimum functionality proves to be versatile and effective than the more customized ones.