

CNT5106C Computer Networks, Fall 2009

Instructor: Prof. Ahmed Helmy

Homework #1

On the Internet Architecture, and Application Layer

[Date Assigned: Sept 21st, 2009. Due Date: October 7th, 2009 in class or to the TAs]

Full grade points: 70, Total Points: 87, Extra points: 17 points ~24%.

Q1. (8 points) In what ways is the Internet 'complex'? Give four different factors contributing to such complexity. Also show the impact of each complexity factor on the design aspects of the Internet protocols.

A1. (possible answer, students should expand upon each point as appropriate)

1- Scalability: number of users, servers, routers, websites, links, so on. The large number increases the complexity of manageability, especially for addressing, routing overhead, and load control. To manage addressing and routing the Internet addressing and routing architectures was designed in a hierarchical fashion, to allow for address prefix aggregation and dampening of dynamic/failure propagation. Congestion control is employed using TCP.

2- Heterogeneity of technologies and implementations at various layers: from the physical layer up to the application layer (depending on the type of link/medium used, routing, transport, security, etc.) there are a variety of technologies used, that use different protocols. Even for the same protocol there maybe different implementations. In order for the network to work properly these various technologies need to 'inter-operate' [i.e., be compatible with each other, and be able to communicate correctly]. Interoperability is an issue that is considered at the core of the network design and the layered protocol architecture in the Internet was devised to facilitate it.

3- Mix of applications, their requirements and their traffic leads to the need to provide for different types of services from the network. Different types of transport layers (UDP, TCP), and different types of queuing (FIFO, weighted Fair queuing), and traffic management architectures (differentiated services and integrated services) are proposed to provide different services to different application classes.

4- Administrative hierarchy, and differences in policies and configurations. These differences affect inter-domain routing and quality (and reachability) of paths in the Internet. The structure of the Internet into autonomous systems (AS) and the user of border router and BGP inter-domain routing allows for the definition of different policies and accommodates the administrative hierarchy.

Q2. (8 points) What is the difference between a hierarchical and a flat architecture? What is the architecture of the Internet in that sense? Mention at least three reasons for such Internet architectural design.

A2. A flat architecture is where all the nodes are at the same level, and there's no distinction or boundaries between nodes. A hierarchical architecture defines boundaries for each level of the hierarchy, and the nodes within each level have a different role in the

network. The Internet has a hierarchical architecture (whether based on the routing hierarchy or the administrative hierarchy).

A hierarchical architecture:

1- dampens dynamics and oscillations, where a failure or change in route in one level (or autonomous system) only needs to be detected and notified within that region. All other regions do not need to be aware of such failure.

2- allows various routing protocols (intra-domain routing) to be implemented within each AS, so long as they use BGP (inter-domain routing) as the border routers. This allows flexibility and individualized routing optimizations within each domain.

3- allows aggregation of routing prefixes, where each AS is assigned an IP address prefix. This allows the routing tables to scale by many orders of magnitude over an approach where each route needs to be stored individually, which may be needed without a hierarchy.

Q3. I. (4 points) What is a push mechanism or protocol? What is a pull mechanism? Give an example application of each.

II. (4 points) If you are designing a new networked application for the Internet, which factors would you consider to decide whether to use a mostly pull or push mechanism? Show how such factors would influence your design.

A3.

I. A push mechanism transfers the information (or messages) closer to potential recipients before an explicit request from the recipients is generated. A pull mechanism is one that requires an explicit request from the recipient(s) in order to transfer the information.

An example of a push protocol is: SMTP. An example of a pull protocol: http

II. The factors affecting the performance of a pull/push protocol include (but are not limited to): 1. access pattern: how often is this object cached and how often is it accessed (example: a push mechanism for a very popular video that is pushed closer to a large population that is going to frequently watch it, would be better than a pull mechanism), 2. delay: what is the delay to obtain the object, and 3. object dynamics: how often/soon does the information in the object expires (example: in a sensor network where the information sensed is constantly changing, but is queried once in a while would be better 'not' to push it, but to pull it when needed only).

Q4. I. (5 points) Can we provide absolute guarantees in the Internet based on its basic architecture? Why? [Contrast the Internet with another network to clarify your answer]

II. (5 points) How would you modify the design of the Internet to obtain better (or absolute) guarantees? Show what each modification would contribute to achieve your goal.

A4. I. - No we cannot provide absolute guarantees. The Internet design is based on statistical multiplexing and best effort service. In Statistical multiplexing the load on the network may occasionally exceed the capacity of the network (e.g., during congestion) leading to the need for queuing, which in turn leads to extra delays and/or losses. The load on the network is not controlled via admission control. By contrast, the telephone network uses admission control and reservation and uses TDM which pre-allocates time slots and resources to sources of traffic, constructing a dedicated circuit for each flow, and can provide service guarantees but wastes more bandwidth when the sources are idle.

I. - provide admission control (to control the load on the network and alleviate congestion, and subsequently queuing delays and losses), then provide reservations along the chosen path (which would require path setup phase before the flow is admitted into the network). This would create a 'virtual circuit' over which the performance may be guaranteed is the allocated resources withstand the maximum load. The utilization of the network may be low, however, since the statistical multiplexing gains (of on-demand resource allocation) may not be fully realized.

Q5. (8 points) Describe the two different ways in which the DNS queries can be propagated in the DNS hierarchy using figures to illustrate. Discuss the advantages and disadvantages of each.

A5. Borrow from the lecture notes

- recursive

- iterative

[One can also think of scenarios where potential attack is carried out on the root servers]

Q6. (10 points) One of the main principles of the Internet design, called the 'end-to-end argument', argues that a function should be pushed out of the network (to the edges) unless absolutely necessary. Discuss putting the following functionalities in the network vs. at the edges, using 'for' and 'against' arguments:

1- reliability

2- congestion control

A6. (10 points)

1. Reliability: (5 points)

In general, reliability requirements are specific to application needs. Some applications require 100% packet recovery, even with delays and jitters (such as TCP-based applications, http, ftp and telnet traffic). Other applications may be tolerant to loss but less tolerant to delays and jitter, such as real-time voice or video applications. Reliability (e.g., using re-transmissions and packet recovery) may add to the jitters and the delays and hence may not be desirable for real-time or voice applications. Hence it is not a good idea, in general, to include error recovery at the network layer (that is not

aware of application needs) and it is better to implement such functionality at the transport layer end-to-end.

In cases of lossy channels in the network (such as X.25 in the early networking days, or wireless links) it may be desirable to reduce the bit error rates on those links by including error recovery at the end points of those links. [In general, most links nowadays have very low BER, and for wireless links the MAC (such as IEEE 802.11) layer provides Ack'ed delivery]. Not including reliability functionality over lossy links could be expensive since any lost packets or frames would have to be retransmitted over many links again just because of this last lossy wireless link.

In the case of flows crossing wireless links in the last hop, reliability is implemented both end-to-end (using TCP) and at the data link layer (using 802.11) in the last link.

2. Congestion control: (5 points)

For congestion control, a similar argument may be given. That is, congestion reaction may be application specific and is better implemented end-to-end. Congestion *notification*, on the other hand, may provide useful information to the end points to react appropriately. Since losses in the network may be due to congestion or other factors, a signal from the network to the end point may help distinguish congestion errors from other errors. Only congestion errors should trigger 'back off' or rate cut at the end points. So, network assistance in congestion notification may help in some scenarios. [extra: In other scenarios network assistance may prevent synchronization effects of congestion control, e.g., RED, or may prevent/isolate misbehavior, e.g., WFQ.]. We shall study more about network congestion notification using ATM later in the semester.

Note that congestion control and reliability may be decoupled functionalities. Take real-time voice and video applications for example. Congestion control is needed since these applications tend to require high bandwidth and need steady flow of packets. They do not need strict reliability requirements and can tolerate loss up to a certain limit. In this case congestion control can be implemented end-to-end.

Q7. (6 points) 'Pee-to-peer communication is much slower than client-server communication' How would you argue for or against the above statement? Support your argument using expressions for transfer delays for large number of connections.

A7. Borrow from the lecture notes

Q8. (10 points) For a link with capacity of 1Mbps, what is the maximum number of users can be supported at the same time in the following situations:

I. (2 points) Circuit switching

II. (2 points) Packet switching such that the probability of exceeding the maximum capacity is less than 0.0004

III. (6 points) Packet switching such that the probability of exceeding the maximum capacity is less than 0.00015

Assume that on average each user is active 10% of the time, and when active has a rate of 100kbps. Show your solution steps and calculations.

A8. I. $1M/100k = 10$ users, since in circuit switching the resources are pre-allocated, we must allocate for the maximum rate

II. & III. In packet switching, the statistical multiplexing allows the instantaneous traffic (not the long term average) to potentially exceed the maximum capacity then more than 10 users can be admitted since, statistically, they will not always be on at the same time. Let's assume that N is the maximum number of users, then we can use the binomial distribution to calculate the probability of a certain number of users (say M) out of N will be active at any point in time.

To avoid exceeding the maximum capacity M should not exceed 10 users. So we use the binomial distribution with parameters N (unknown), $x > 10$, $p=0.1$ and $q=0.9$. To get N , some trial and error is needed. Perhaps we can write a program that calculates the probability of $x > 10$ for $N=10$ onward and stops when that probability exceeds 0.0004. Any other tool (binomial calculator, excel, etc.) can be used for this task.

The maximum number of users (or sources) that can satisfy the probability of 0.0004 is 35. Any number over 35 will yield a probability of over 0.0004.

III. For a probability of 0.00015 the maximum number of users is 31. Numbers over 31 result in a probability of exceeding 10 simultaneous 'on' exceeds 0.00015. (the steps should be explained clearly as in the write up for II above).

[Extra: other examples: for 0.0015 the max number is 40.

Check <http://www.stat.tamu.edu/~west/applets/binomialdemo.html> and try out different numbers. Note that due to various rounding and resolutions for the different tools the numbers you may get may not be 'exact'. If you show correct steps and correct calculations, even with different resolution, the answer is correct.]

Q9. (5 points) (Stateful vs. Stateless) Discuss one advantage and one disadvantage of having a 'stateful' protocol for applications.

A9. The protocol can now maintain state about (i.e., remembers) users preferences (e.g., shopping preferences as in browser cookies),

Disadvantage: when failure occurs the state needs to be reconciled (more complexity and overhead than stateless)

[other correct and reasonable answers are accepted]

Q10. (5 point) (Web Caching) Describe how Web caching can reduce the delay in receiving a requested object. Will Web caching reduce the delay for all objects requested by a user or for only some of the objects? Why?

A10. Web caching can bring the desired content "closer" to the user, perhaps to the same LAN to which the user's host is connected. Web caching can reduce the delay for all objects, even objects that are not cached, since caching reduces the traffic on links.

Q11. (9 points) One of the main problems in peer-to-peer networks is finding other peers and finding files/content, both of which relate to resource discovery problems. Describe three main approaches to resource discovery in peer-to-peer networks, discussing their advantages and disadvantages.

A11.

1. Centralized directory of resources/files, as in Napster. Advantage is that search for resources is simple with min overhead (just ask the centralized server). The disadvantages are: single point of failure, performance bottleneck and target of lawsuit.
2. Fully distributed, non-centralized architecture, as in Gnutella, where all peers and edges form a 'flat' overlay (without hierarchy). Advantages: robustness to failure, no performance bottleneck and no target for lawsuit. Disadvantages is that search is more involved and incurs high overhead with query flooding.
3. Hierarchical overlay, with some nodes acting as super nodes (or cluster heads), or nodes forming loose neighborhoods (sometimes referred to as loose hierarchy, as in BitTorrent). Advantages, robust (no single point of failure), avoids flooding to search for resources during queries. Disadvantages, needs to keep track of at least some nodes using the 'Tracker' server. In general, this architecture attempts to combine the best of the 2 other architectures.