中 山 大 学 软 件 学 院 **2012** 级 软 件 工 程 专 业 (2014 学年秋季学期)

《SE-301 计算机网络》期末试题答案(A)

1.

(a)

1) Electrical noise that corrupts the signal and

2) Buffer overflow in an intermediate node due to congestion.

(b) The four factors of packet delay are processing, transmission, propagation, and queuing. For (1) the predominant delay will be transmission in the 1ms range (processing is microseconds, propagation is nanoseconds, and queuing in non-existent). For (2) the propagation will be the largest delay (in the 100s of milliseconds), and for (3) the largest delay will likely be queuing due to the many hops between the Guangzhou and Shenzhen.

2.

(a) The answer is B. Disabling exponential backoff of timeouts will enable your TCP to recover from repeated loss (i.e., loss of retransmitted packets) more quickly. For the others:

- If you disable timeout retransmissions, then you will lose performance any time a lost packet is not followed by enough duplicate acknowledgments to trigger fast retransmission (you will never recover from the loss).
- If you disable fast retransmission, then you will lose performance any time you could have detected a loss quickly by observing 3 duplicate acknowledgments (you will only recover from the loss later, when the retransmission timeout finally expires).
- If you do not perform RTT estimation and do not adapt RTO based on it, then you're stuck with a fixed value for RTO. This may be too high, in which case you lose performance since you could have recovered from losses more quickly; or too low, in which case you will lose performance by retransmitting unnecessarily.

(b) **Answer:** if everyone's TCP failed to use exponential backoff for their retransmission timeouts, then during times of congestion there would be no mechanism in the network to drain the load presented to the network. This in turn could lead to congestion collapse. Note, full credit was allowed here when the answer in the previous part of the question differed from the correct one, but the explanation for this part of the question was consistent with that previous answer.

(c) **Answer**: there are a number of possibilities.

- Disable AIMD congestion control. When your TCP suffers a loss, leave the window alone. This allows you to transmit more data, potentially improving performance.
- Skip congestion avoidance. Never leave Slow Start, or always just send the entire window offered by the receiver.

- Change your TCP to send multiple ACKs for each incoming packet (the“ack splitting” attack).
- Increase your send and/or receive buffers (this one doesn’t violate congestion control in any fashion; it’s really just tuning).

3.

(a)

Step	N’	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0 (Initialization)	**A**	**5,A**	**8,A**	**13,A**	∞	**34,A**
1	**AB**		**7,B**	**13,A**	∞	**34,A**
2	**ABC**			**13,A**	**9,C**	**34,A**
3	**ABCE**			**11,E**		**18,E**
4	**ABCED**					**18,E**
5	**ABCEDF**					

(b) Here are some problems that would arise:

- Link-State (LS) routing requires flooding of link information to all nodes in the network. For a very large network such as the Internet, this would result in a huge number of messages. Distance-Vector (DV) does not need to flood information to the same degree.
- Dijkstra’s algorithm runs in time $O(N^2)$ for N links. Thus, the computational cost at each node would become prohibitive. DV has better computation cost, although each can take a long time to converge. DV is prone to forming temporary routing loops.
- LS routing exposes an ISP’s precise connectivity, which some ISPs prefer to keep private for competitive reasons. DV does not advertise how an ISP gets to a given location, so they are better in this regard.
- LS routing does not allow ISPs to express policies regarding what traffic they are willing to carry. DV also is not able to express such policies.

4.

(a) Answer: If all packets at 10,000 bits long, it takes 100usec to send the packet over a 100Mbps link, 400usec to send over a 25 Mbps link, and 10usec to send over a gigabit link. The sum of the three link-transmission times is thus 510usec. The sum of the propagation delays is 200+100=300 msec. **Thus the total end-end delay is 300.510 msec.**

(b) Answer: **25 Mbps**, the bottleneck link speed.

(c) Answer: We assume that requests are serially satisfied. 50 % of the requests can be delivered at 25 Mbps and 50% of the requests can be delivered at 1Gbps. **So the average rate is 512.5 Mbps.**

(d) Answer: the 25 Mbps remains the bottleneck link, which is not shared between clients. So the answer is the same as c) above. Note that we assume that the

100Mbps is shared at a fine grain, **so that each client can get up to 50Mbps over that link.**

5.

(a) The minimum packet size is necessary to ensure that a node is still transmitting when the first bit of a packet reaches the farthest node, so that a collision can be detected at both ends of the medium. The minimum packet size is determined by the
RTT for the maximum distance allowed.

(b) Wireless devices are not full duplex, so they can't check the channel state and transmit at the same time. Additionally, due the non-symmetric nature of the medium, it is not guaranteed that a collision can be detected at both ends.

6.

(a) We divide M by P and the remainder is the CRC. We use modulo-2 math with no borrows or carries. The quotient of the division is 110110 and the remainder is 10. Thus, the CRC is 010.

(b) Consider a message of at least two N-bit words. For a N-bit check sum if an odd numbered work has a value M added to it that an even numbered word has subtracted from it, then the checksum will be the same as the for the original message.

7. Any reasonable answer which contains the following terminologies: DNS, TCP/UDP, IP, MAC in **network and link layers** is acceptable.

0. 便携机上生成 DHCP 请求报文，DHCP 服务器收到请求后以 CIDR 块分配 IP 地址，并把 DHCP ACK 返回给便携机，便携机的 DHCP 客户记录下 IP 地址和 DNS 服务器的 IP 地址，安装好默认网关的地址，成功初始化网络组件.

1. 将 URL 放进 DNS 查询报文，放入以太网帧发送到网关路由器
2. 要想获得网关路由器的 MAC 地址，要通过 ARP 查询报文来获得
3. 于是将以太网帧发送到网关路由器（先发送到交换机，交换机再交付到网关路由器）
4. 网关路由器提取帧中的目的地址，并转发到最左边的路由器
5. 最左边路由器接收到该帧后，同样提取目的地址，根据转发表确定接口朝 DNS 服务器转发数据报
6. DNS 服务器得到了包含 DNS 查询的 IP 数据，根据 DNS 查询报文，在 DNS 数据库中找到 URL 对应的 IP 地址的 DNS 源记录（假设存在在缓存中，不存在则按照 DNS 的迭代递归等查询获得），制作成 DNS 回答报文，返回给用户主机
7. 用户从回答报文成功得到了 URL 所对应的 IP 地址，生成 TCP 套接字，三次握手之后成功进入连接状态
8. 借助套接字发送 HTTP GET 报文，相应服务器读取请求后返回 WEB 页内容，用户主机接收后成功得到主页！