Q1

There are five parts can cause delay in the structure of this question: source, 1,000km, router1, 5,000km, destination, and there are four kinds of delay: **nodal processing delay, queuing delay, transmission delay, propagation delay**. Also, as the router is using store-and-forward-transmission, so processing and sending cannot be done synchronously. Then we can build this table:

	Source	1,000km	Router1	5,000km	Destination
Nodal processing delay	0	0	0.001s	0	0
Queuing delay	0	0	0	0	0
Transmission delay	8000bit/1Mbps = 0.008s	0	8000bit/20Mbps = 0.0004s	0	0
Propagation delay	0	1000km/(2.5*10^8m/s) = 0.004s	0	5000km/(2.5*10^8m/s) = 0.02s	0

The delay = 0.008s + 0.004s + 0.001s + 0.0004s + 0.02s = 0.0334s

If there are other traffics in router, then a Queuing delay in router should be added in the whole delay.

Q2

Circuit-switching would be more efficient if all sources do not change the sending rate. For example, using TDM, Circuit-switching allocates 1 frame for each source, then the occupation of each frame will be 100% as no source will turn to silence or change frequency. Circuit-switching guarantees end-to-end connection by establishing a reserved link while packet-switching could not. Other than this, the cost of setting up a connection is ignorable as the lone period of transmission.

Q3

According to the question,
F = 10*10^9 bits
Us = 20 * 10^6bits/s
d = 10^6bits/s

Client-server: D = max{NF/Us, F/d}			
	N=10	N=100	N=1000
u=200*10^3bits/s	10,000	50,000	500,000
u=600*10^3bits/s	10,000	50,000	500,000
u=1*10^6bits/s	10,000	50,000	500,000

Peer to Peer: D = max{F/Us, F/d, NF/(Us+(u1+…+uN))}			
	N=10	N=100	N=1000
u=200*10^3bits/s	10,000	25,000	45,454
u=600*10^3bits/s	10,000	12,500	16,129
u=1*10^6bits/s	10,000	10,000	10,000

This approximation might be slower than bit torrent as users may leave the torrent in real circumstance.

Q4

The reason that there is a blink line is for indicating all meta-information for the message has been sent. It is a separator between header and body. HTTP could be designed without the blink line if there is an alternative separator does the same job.
The content-length header indicates the length of the body, it helps clients to determine where a response ends. It is helpful for reusing a connection. But it is not necessary, an alternative header "transfer-encoding" can also define the end of the header. Or the connection can be terminated after each response if we do not want to use "content-length".

Q5

1. The web page and images are cached in the web proxy cache and the domain is not cached in the local DNS
   a) The local DNS checks domain start from root DNS server
   b) The local DNS caches the domain
   c) Client requests the web page from proxy server, a connection is set up
   d) Proxy server checks the authority of this request, assuming it is ok
   e) Proxy server found the web page and send it to client
   f) Client read the page and then send requests to proxy server
   g) Proxy server found the resources and send them to client
   h) Connection close

2. The web page is not cached; but the domain is cached in the local DNS
   a) The local DNS returns ip to client
   b) Client requests the web page from proxy server, a connection is set up
   c) Proxy server checks the authority of this request, assuming it is ok
   d) Proxy does not find the cache, sends request to internet
   e) Proxy receives the webpage, caches it and sends to client
   f) Client read the page and then send requests to proxy server
   g) Proxy does not find the cache, sends request to internet
   h) Proxy receives the resource, caches it and sends to client

i) Connection close

3. Neither the web page nor the domain is cached
   a) The local DNS checks domain start from root DNS server
   b) The local DNS caches the domain
   c) Client requests the web page from proxy server, a connection is set up
   d) Proxy server checks the authority of this request, assuming it is ok
   e) Proxy does not find the cache, sends request to internet
   f) Proxy receives the webpage, caches it and sends to client
   g) Client read the page and then send requests to proxy server
   h) Proxy does not find the cache, sends request to internet
   i) Proxy receives the resource, caches it and sends to client
   j) Connection close

As we are using persistent HTTP, the TCP connection only need to be established once. We assume there is no queuing delay or processing delay or transmission delay between all nodes, and 1RTT = 2*(1500,000m/2.5*10^8m/s) = 0.012s, then it takes:
0.012s for TCP + 0.006s for web page request + 0.05s for web page download + 0.006s for image1 request + 0.01s for image1 download + 0.006s for image2 request + 0.01s for image2 download ≈ 0.1s

Q6

Why pipelining is closed by default?