1. Relevant RFC Link: https://www.rfc-editor.org/rfc/rfc2616.txt
    a. Under the specifications of the HTTP/1.1 RFC, in order for a connection to be closed, either the client or server must include within the connection header of a packet the ‘close’ token. This signals to the other side that this communication will be the last one of that connection and any subsequent messages will need to re-establish a connection. Either the client or server may close the connection in this way, by sending the ‘close’ token to the other.
    b. HTTP does not itself provide any native encryption. In section 15 of RFC 2616, they discuss security considerations, including encryption, by first stating that there are no definitive solutions presented. It leaves any form of encryption up to the end systems it is transporting data between.
2.
    a. Since the connections are non-persistent, for each object requested, the HTTP protocol will have to perform its handshaking functions. In a non parallel operation, each object will need to wait for the propagation delay, P, associated with its communications (4 of them) for every single object, totalling 40P. When this is serialized, they all take place virtually simultaneously, meaning that the overall wait time decreases to 4P. Splitting the bandwidth of the link in 10 means that the download speeds are the same for both parallel and serial downloads and the only difference is the wait time for propagation of communication, So parallel downloads would make sense in this application.
    b.

First, let us calculate the transmission time for a parallel, non-persistent request of 10 objects that are 120 000 bits long. 3 control packets will need to be sent for each object, (a SYN, a SYN-ACK, and an ACK), and each of these will take 240/(250000/10) second (0.0096). Then, the transmission of the 120 000 bit object will take 120000/(250000/10) seconds (4.8). Since these are conducted in parallel, the total time for downloading the objects would be 4.8096 + 4P where P is the propagation delay on our transmission media.

Next, let us calculate the transmission time for a persistent connection. 3 control packets will need to be sent total, taking 240/250000 seconds (0.00096). Then, all 10 downloads will be done in series, 10*(120000/250000) seconds (4.8 seconds). This totals to a download time of 4.80096 + 13P.

In order to determine if this represents a significant time difference between the two transmission methods, we must estimate the propagation delay for a 10 meter link cable. Without knowing the specifics of the transmission medium (i.e. if the cable is twisted pair, fiber optic, coaxial, etc.) it would be impossible to determine the precise propagation delay, but for the sake of estimation, let us assume it is close to the speed of light, or approximately 275,000,000 meters/second. This gives us a propagation delay P that can be approximated as 3.6*10^-8 seconds. There is a negligible difference between the two cases and we do not expect significant gains over the non-persistent, parallel case.