COMP347 Tutorial 12

In this course you have learned a lot about network protocols at various levels. In this tutorial, you will apply what you have learned to the design of an application level network protocol.

Consider the following application from a networking perspective. The company has developed a camera system that is capable of detecting traffic congestion on the roads. A network of these intelligent cameras has been installed in the Sydney CBD. Each camera monitors an intersection to detect traffic congestion. Human operators at various locations may wish to be notified when one of the cameras detects traffic congestion. The task is to design a networking protocol that can be used by the cameras to report incidents of traffic congestion to a central server.

The following high-level design decisions have already been made.

1. The algorithm to detect traffic congestion requires parameter data to be sent to each camera. The parameter data may change from time to time and may be different for different cameras. You are not given the specification of the contents of the parameter data. The number of bytes of parameter data is variable. Your protocol must provide a mechanism by which up to 2 kB of parameter data may be sent to the camera. Your protocol should treat the parameter data as opaque data to be transported within your protocol. Upon receiving the parameter data, the camera will send back a status response as defined below.

2. When the camera detects traffic congestion, it will send a message to the server notifying it of the traffic congestion. The message will contain the camera identification number (which identifies the intersection where the camera is located), the date and time of the congestion report, and the number of seconds that the intersection has been congested. The server will acknowledge the receipt of each congestion report by sending a status reply that contains the camera identification number and date and time of the report that is being acknowledged. An invalid congestion report (if that is possible) will not be acknowledged.

3. After sending a report, the camera will continue monitoring intersection. If the congestion is relieved, the camera will send a report indicating that the intersection is no longer congested. This report will contain the camera identification number, the date and time of the report, and the number of seconds since the intersection was last considered to be congested. The server will acknowledge the receipt of each such report by sending a status reply that contains the camera identification number and the date and time of the report that is being acknowledged.

4. If the intersection remains congested for a long period of time, the camera will send additional congestion reports confirming that the intersection remains congested. Each such report will contain a number of seconds that the intersection has been congested since the beginning of the congestion period. The maximum congestion period is expected to be 12 hours and any congestion period longer...
than this does not require the report to contain the exact duration of the congestion period. Each such report will be acknowledged by the server.

5. In order to prevent excessive load on the server, the server may vary the interval between additional congestion reports. The server may specify an interval in seconds between additional congestion reports. The minimum interval is one second and the maximum interval is two hours. The server varies the interval by sending a command to the camera. The camera will respond with a status message. If the interval specified by the server is invalid, and the camera will reject the setting.

Status messages
- Acceptance/OK
- Invalid report interval
- Invalid parameter data

Your task is to design a suitable protocol for the camera to communicate with the server.

1. List the messages that the server must be able to send to the camera and that the camera must be able to send to the server according to the above specification.
2. Specify a transport mechanism to be used for your protocol and justify your decision.
3. Specify the syntax of each of the messages in your protocol and explain your design decisions.
4. Specify the framing for messages in your protocol and justify your decision.
5. Show examples of interactions between the camera and the server in each of the five aspects described above. Consider both success and failure responses where appropriate.
6. Discuss how your protocol would handle transmission failures -- lost messages/packets.
7. Suppose the protocol was to be extended to allow the server to request an immediate status report from all the cameras in a particular street. Discuss how this may be achieved and consider the pros and cons.