Read Section 1 and 2 of the *Modbus over Serial Line Specification and Implementation Guide*.

1. We can classify the network environment along two lines, propagation delay and load on the system. These can be viewed as a $2 \times 2$ matrix as below.

```
<table>
<thead>
<tr>
<th>short delay</th>
<th>long delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>low load</td>
<td></td>
</tr>
<tr>
<td>high load</td>
<td></td>
</tr>
</tbody>
</table>
```

Recall that “long delay” means the propagation delay is on the order of the insertion time or longer; and that “high load” means that the fraction of time spent sending data assuming perfect coordination approaches 100% of the time.

For the following five types of media access protocols, place them in the cells where it is reasonable to use them. Each one may appear in some, none, or all of the cells.

- Polling
- Token Passing
- Aloha
- Slotted Aloha
- Reservation

2. The Modbus protocol is typically implemented over a 19,200 baud (19200 bps) serial line with one *master* and up to 247 *slaves*. All communication is controlled by the master which initiates all communication in a simple polling scheme. The minimum message consists of a 4 bytes (address, command, and CRC). Each byte is sent as 11 bit characters with 8 bits of data and 3 bits for start, stop, and parity. Messages are delimited by a minimum 3.5 character gap between messages.

A control application polls each of the slaves one after the other in a round-robin fashion. Each poll consists of a request from the master and a response from the slave.

(a) Assume a minimum size message in each direction, no propagation delay, no gap between characters, and no processing delays; compute the maximum number of times a slave can be polled every second.

(b) Assume, instead, that the response requires 252 bytes (the maximum allowed by the Modbus protocol). Compute the maximum number of times a slave can be polled every second? How long would it take to poll 247 slaves?

(c) If a request or response is lost or garbled, then the master simply repeats the request after a timeout. The timeout is not specified, but it is recommended to be at least one second. If the timeout value is one second and a request or a response arrives correctly 99.5% of the time (i.e. the probability of both the request and the response of getting through is $(0.995)^2$), compute the expected number of slaves that can be polled every second with the assumptions in part (a). Do these errors significantly change the answer in (a)?