It is recommended that students work in pairs with a least one skilled programmer. Java and C would be preferable languages but anything else and any open source libraries can be used to implement the functionality below.

Spaceship Enterprise (S) is heading to a hyperspace wormhole (HWH). Due to an accidental heptronic resonance, the onboard Teonic clock lost synchronization at time x=0. S' clock needs to re-lock with the reference hyperspace data beacon (R) that regulates HWH traffic, or face disaster. The data beam is a sustained stream of data packets traveling between S and R in both directions and relayed through multiple forwarding nodes placed in space. Due to the stochastic nature of the cosmic time and the space anisotropic nature, the propagation delay is subject to variations and asymmetric delays between directions.
The attached .xlsx file contains the propagation delay (PDout and PDin) of individual data packets traveling between S and R. Assume that at moment 0 the $PD_{out} = PD_{in.}$
The training file was built with known w1 and w0. Also the w1 and w0 can be modified to generate new data and verify the program. In and out directions are relative to S.
PD can be calculated from real timestamps collected at both ends and sent to a location for computing according to the formulas:
PDin = Arrival time S - Departure time at R = y2-x1 ~= (w1in(k) -1)*x + w0in(k);
PDout = Arrival time R - Departure time at S = x4 – y3 ~=(1-w1out(k))*x - w0out(k);
(indexes 1,2,3,4 were added for clarity)

=========== / =============

1. Determine the relation between the R and S time assuming the relation is quasi linear for short time intervals (fractions of second) and can be expressed by the formula $PD_{out} = w1_{out}(k) * x + w0_{out}(k)$ and $PD_{in} = w1_{in}(k) * x + w0_{in}(k)$ where x is the S time domain. You must write the program that estimates in real time the parameters $w1_{out}(k)$, $w0_{out}(k)$ from consecutive sets of 1000 records (test set k). Similarly calculate $w1_{in}(k)$ and $w0_{in}(k)$ from sets of 1000 PD_{in} records (make sure that both are during the same x time interval) . After calculating the w1(k), w0(k) parameters from a test set "k" the program has to measure the actual error and Loss on the next 1000 PD pairs (set k+1). Repeat the process over the entire data set of the file. Draw the graphs for the error (for each PDout and PD in) and for measured Loss associated with each set k of 1000 PD pairs collected during a time interval (indexed by k) and provide the data in a .xlsx or .csv format.
Represent it in graphic form.
Calculate the correction of the S timing for an interval k with the formula $C(k)= (PD_{out} – PD_{in}) /2$.
Provide another no parametric method to predict (learn) the next 1000 propagation delays in both directions.

(Thursday Nov 12)

2.
Build a neural network with "N" C(k) inputs. The network can assume that the local clock cannot drift more than H during each k cycle. H is a constant and could be 50 ppB/S (or 0.5 during interval k).
Use neural networks / Bayesian networks to predict the next C(k)
Consider the online approach.
Saturday (Nov 14)
3.
Collect real data. Collect the PDin in and PDout with two computers at different locations in different networks (same room or different rooms) using wired and wireless communications. Send data to a common location (can be the S computer). (Details on sending will be discussed on Nov 12). Run the previous programs over the new sets of data.
Can use very smallest packets (64 bytes) and make sure there is enough bandwidth available for other wifi users at that time. Wired GE connections should not be a problem.

(Sat Nov 21)