All of the suggested course projects for CSE 758 revolve around Bitwiser Lab's flagship product; Netwiser. Netwiser is a unique platform for network software development. It includes an easy to use graphical user interface and a C++ framework for creating all types of network-intensive applications. Netwiser eases the difficulty of developing network applications and protocols by providing a higher-level of abstraction for creating network software. It provides you with a clear view of the network and a powerful set of tools to model, develop and deploy. See [http://www.bitwiserlabs.com/netwiser/](http://www.bitwiserlabs.com/netwiser/) for more details.

We have provided two suggested course projects. The project ultimately selected would have to be based on the experience and interest of the students.

1. One of Netwiser's primary use cases is the development of a networking protocol. Netwiser's unique C++ framework allows a networking protocol to be implemented as a reusable software component, called a module. The framework allows modules to be compiled to run on a variety of platforms such as Linux running on a desktop PC or embedded device, VxWorks on an embedded device, or even Netwiser's discrete event simulator. Netwiser currently has a library of reusable modules that implement various network protocols such as AODV, 802.11, IP, ARP, UDP and others.

   The course project would be to extend this library by implementing Dynamic Source Routing (DSR), an ad-hoc routing protocol. The specific details of DSR are defined in the IETF draft version 10 available at [http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt](http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt). The group would need experience with C++ and basic networking concepts or have a willingness to learn.

   (a) The group should implement the basic route discovery (section 3.1, 8.2) and basic route maintenance (section 3.2, 8.3) schemes of DSR.

   (b) A DSR implementation can do nothing alone. The group should build a working node from their DSR implementation and other software modules from the Netwiser library, such as 802.11, ARP, UDP, or IP.

   (c) The implementation should be tested and validated in simulation. The simulation should show DSR routing packets through a simple multi-hop network.

   (d) More complex simulations should be created that show how the DSR implementation reacts under varying network conditions, such as increased density or traffic.

2. The Ohio State University has a history of research in the field of sensor networking and sensor network applications. Netwiser is well suited for building these kinds of applications. The assignment is to use Netwiser to develop a target tracking service for sensor network applications. This service will be used by sensor network developers through the Netwiser module facility.

   The scenario is that sensor nodes are uniformly spread through a field. Each node has a sensor
and a wireless radio. The sensor can be any type as long as it is related to the distance from the object it is sensing. This could be a magnetic sensor, a vibration sensor, a heat sensor, or even a small radar. The only requirement is that a sample from the sensor returns a value that is related to the distance from the object being sensed. The sensor has low sensitivity and therefore can only sense objects that are relatively nearby.

The wireless radio is used to transmit data back to a base station that analyzes this data. Every node in the network individually collects data from its sensor and sends it to the base station via its wireless radio. The wireless radios are low power and therefore can only communicate with nodes that are relatively close. Nodes that are far away from the base station must forward their data through other sensor nodes to reach the base station. The nodes form an ad-hoc network because routes to the base station must be automatically discovered by the network.

The primary goal is for the base station to determine the location of any targets in the field and to track those targets as they move through the field. The location is calculated by processing the magnitude of the sensed values from the nodes near each target. Once a target is discovered, the base station assigns it a target ID and tracks it as it moves through the field. For existing targets (ones that have already been discovered and assigned an ID), the base station will make a simple statistical decision as to which target has which ID based on previous locations and velocities.

The software for this project will be built to run within the Netwiser simulator and will use existing modules available from the Netwiser library. AODV can be used as the ad-hoc routing protocol which is already available from the Netwiser library. The following three software components are deliverables:

(a) A target simulator and sensor module to simulate the tracked targets. The target simulator will simulate targets moving around an open field. A sensor module will return a value each time it is sampled that reveals the distance to simulated targets. The sensor module on each node works with the target simulator to decide what value to return for each sample.

(b) Each sensor node will have a data collection engine. This is a module that goes on each sensor node and decides when to sample the sensor and when to send data to the base station. It should make intelligent decisions as to what data is important to send, how often to send it and how to aggregate data collected on that node. It may also take advantage of the multi-hop routing and aggregate or even process and integrate data being forwarded from other nodes.

(c) The base station will have a data analysis engine. This is a module that goes on the base station and processes the data collected from the entire network. This module is responsible for discovering targets, assigning target IDs, and calculating the location of those targets. A target tracking interface shall be defined for this module to implement and therefore provide the target tracking service to other modules. This interface shall support querying which targets currently exist in the field and querying the location of a target given its ID.

To demonstrate the effectiveness of the target tracking service, a small application will run on the base station in a separate module that uses the target tracking service and periodically logs the time along with the target ID and location of each target. This log file will be compared with the actual target motion in the target simulator to validate the accuracy of the service.