```
#include <iostream>
#include <vector>
#include <string>
#include <algorithm>

using namespace std;

// Structure to hold Bus information
struct Bus {
    int id;
    int capacity;
    string status; // "Available", "In Use", "Under Maintenance", etc.
};

// Structure to hold Route information
struct Route {
    int id;
    string start_location;
    string end_location;
    int distance; // in kilometers
    int demand;   // number of passengers
};

// Structure to hold Schedule information
struct Schedule {
    int bus_id;
    int route_id;
    string departure_time;
    string arrival_time;
};

// Function to calculate travel time based on distance
string calculateTravelTime(int distance) {
    int hours = distance / 50;  // Assume average speed of 50 km/h
    int minutes = (distance % 50) * 60 / 50;
    return to_string(hours) + "h " + to_string(minutes) + "m";
}

// Function to assign buses to routes
vector<Schedule> assignBusesToRoutes(vector<Bus>& buses, vector<Route>& routes) {
    vector<Schedule> schedules;
    for (auto& route : routes) {
        auto it = find_if(buses.begin(), buses.end(), [&route](Bus& bus) {
            return bus.capacity >= route.demand && bus.status == "Available";
```

```
        });

        if (it != buses.end()) {
            // Found a bus for the route
            Schedule schedule;
            schedule.bus_id = it->id;
            schedule.route_id = route.id;
            schedule.departure_time = "08:00 AM"; // Placeholder, logic needed for dynamic time
            schedule.arrival_time = calculateTravelTime(route.distance);

            schedules.push_back(schedule);

            // Mark bus as in use
            it->status = "In Use";
        } else {
            cout << "No available bus for route " << route.id << endl;
        }
    }
    return schedules;
}

// Function to print the schedule
void printSchedule(const vector<Schedule>& schedules) {
    cout << "Bus Schedule:\n";
    for (const auto& schedule : schedules) {
        cout << "Bus ID: " << schedule.bus_id
             << ", Route ID: " << schedule.route_id
             << ", Departure: " << schedule.departure_time
             << ", Arrival: " << schedule.arrival_time << endl;
    }
}

int main() {
    // Example buses
    vector<Bus> buses = {
        {1, 50, "Available"},
        {2, 40, "Available"},
        {3, 30, "Under Maintenance"}
    };

    // Example routes
    vector<Route> routes = {
        {1, "A", "B", 120, 45},
        {2, "B", "C", 80, 30},
```

```
        {3, "C", "D", 60, 25}
    };

    // Assign buses to routes and create schedules
    vector<Schedule> schedules = assignBusesToRoutes(buses, routes);

    // Print the schedules
    printSchedule(schedules);

    return 0;
}
```

Output

```
No available bus for route 3
Bus Schedule:
Bus ID: 1, Route ID: 1, Departure: 08:00 AM, Arrival: 2h 24m
Bus ID: 2, Route ID: 2, Departure: 08:00 AM, Arrival: 1h 36m
```