Virtual Robotics Challenge
Technical Guide

DRAFT Version 2 on November 21, 2012

DISTAR Case 20372

Table of Contents

1 Introduction

This document provides a technical guide to the Virtual Robotics Challenge (VRC). The scope of this document includes the computing environment, how runs will be conducted, and what parameters will be varied.

Related documents include the following:

- Virtual Robotics Challenge Rules – States the rules and the overall schedule leading to the VRC in June 2013
- Virtual Robotics Challenge Qualification – Defines procedures and schedule for qualifying for the VRC – Not yet available
- Virtual Robotics Challenge Practice – Defines procedures and schedule for conducting practice runs after qualification and before the VRC – Not yet available

2 Computing Environment

Figure 1 shows a block diagram of the computing environment for the Virtual Robotics Challenge.

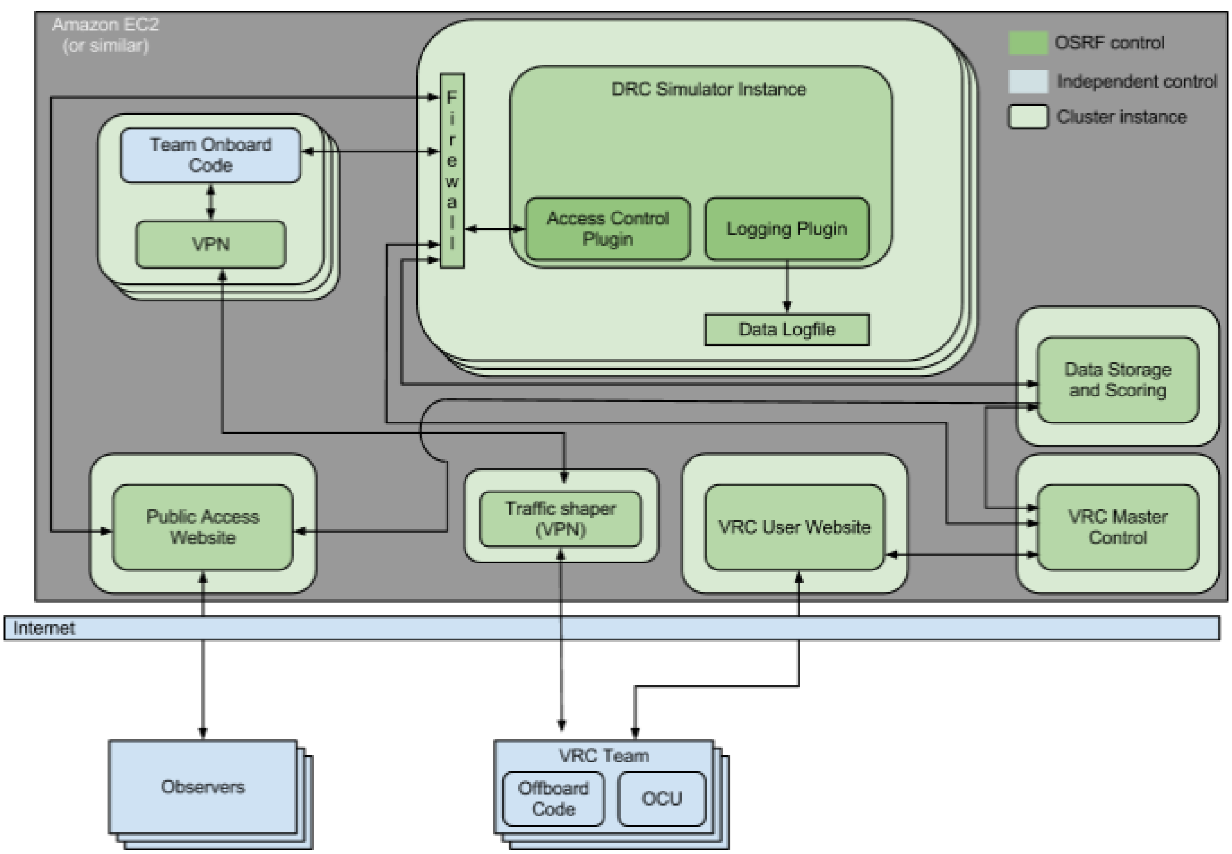

Figure 1. Block diagram of computing environment

The DRC Simulator runs on a computer in the cloud (see “DRC Simulator Instance” in the block diagram). This simulates the robot and the environment. During the VRC competition, a separate instance of the DRC Simulator will run on a separate, dedicated machine for each team.

Team Onboard Code runs on a computer in the cloud (see “Team Onboard Code” in the block diagram). This computer is a surrogate for the computer that will be on the utility vehicle (along with the power generation module) and tethered to the robot in the DRC-13 and DRC-14 events. One expects that the team will run perception, planning, and control code on this computer, that is, all the processes requiring high-bandwidth communications with the robot. During the VRC competition, a separate instance of the Team Onboard Code will run on a separate, dedicated machine for each team. That machine will run Linux, and teams will have VPN access, SSH access, and super-user permissions. Each team is responsible for deploying their code on the Team Onboard Code machine.

It follows that for *N* teams running at the same time, there will be *2N* cloud computers, *N* for running simulators and *N* for running Team Onboard Code.

Team Offboard Code runs on a system of one or more computers at the team facility (see “Offboard Code” within “VRC Team” in the block diagram). This computer system serves as the Operator Control Unit. One expects the operator controls (keyboard, joystick, gaming console, or equivalents) to run here (see “OCU” within “VRC Team” in the block diagram).

Between the Team Offboard Code and the Team Onboard Code instances, the OSRF-controlled VPN handles all data. This VPN will have the ability to throttle bandwidth, increase latency above the baseline value, and inject dropouts (as specified in Section 4.2).

Note that teams may have any number of machines at their end, including local or cloud-hosted machines. But all machines containing Team Offboard Code are on the other side of the data-metered link from the Team Onboard Code.

3 Conducting Runs

The control diagram in Figure 2 depicts the sequence of operations to conduct a run.

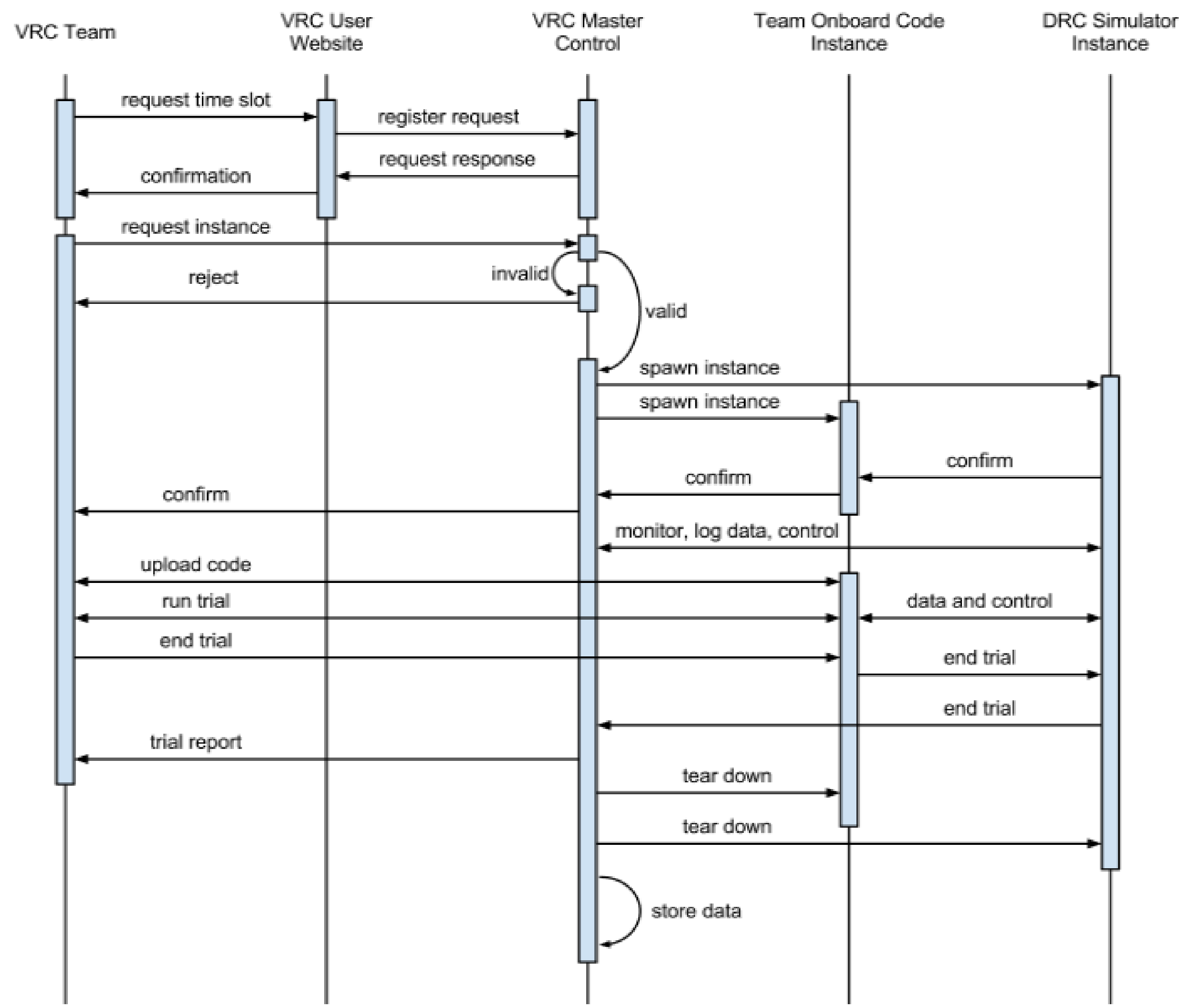

Figure 2. Sequence of operations for conducting a run

Each team requests a time slot for a trial using the VRC User website. This website checks and confirms with the VRC Master.

At the requested time, the team requests a simulation instance from the VRC Master. The master either accepts or rejects the request, depending on the authentication and validity of the request. If the request is valid, the VRC Master spawns an instance of the DRC Simulator and an instance for the Team Onboard Code. Relevant connection information is passed back to the team through the confirm message. The team is then able to start their trial, with the VRC Master monitoring and recording the trial.

At the end of the trial, the team indicates completion, and the VRC Master shuts down the simulation and team cluster instances. A report is generated for the team, and data post processing is conducted on the Data Storage and Scoring module.

4 Configurations

Tests of each task will have multiple configurations in different parameter spaces described below. Configurations used will be distinct from run to run, but these distinct configurations will be identical for each team.

4.1 Object Location and Orientation

Items will be moved within the virtual environment between runs to simulate uncertainty in positioning, and prevent training the controller to handle a known geometry across runs. This will include the placement of obstacles (for example, rocks or barriers), the placement of tools (for example, the utility vehicle), and the starting placement and starting orientation of the robot itself.

4.2 Communications Parameters

To simulate the effects of signal degradation induced by beyond line of sight (BLOS) conditions, RF interference, or other related circumstances, the communications parameters will be varied by the OSRF-controlled VPN (see the "Traffic Shaper" block in Figure 1). The parameters to be varied include the following:

1. Latency. Beyond the latency inherent in the system caused by connection to the cloud, additional latency will be introduced. The additional latency imposed is as follows:

   Lower Bound: 0 ms
   Upper Bound: To be determined. [We are in the process of formulating an approach that determines the amount of latency to be added as a function of connectivity, so that teams with excellent connectivity and teams with poor connectivity experience approximately the same range of latencies. Suggestions welcome.]

2. Bandwidth. To replicate realistic conditions for available bandwidth, and to avoid development of operator control stations relying on high-bandwidth data at all times from the robot, the rate of data flowing between the Team Onboard Code and the Team Offboard Code will be restricted.

   Lower Bound: 32 kbps (kilobits / second). Comparable to medium grade telephony
   Upper Bound: 1,000 kbps (kilobits / second). Comparable to VHS-quality video tele-conference

3. Dropouts. In addition to restricting or time shifting data, at times all bidirectional data will be lost completely for a set period of time.

   Lower Bound: 0 sec
   Upper Bound: 5 sec

4. Packet loss. At times all data may be dropped as in item #3, but other times individual packets of information will be lost. The packet loss will be designed to drop, rather than buffer packets.

Lower Bound: 0% packet loss
Upper Bound: 100% packet loss, dropped (unbuffered)

4.3 Contact Friction Properties

To simulate different surface conditions to include different road materials, liquid or loose solid spills, weather effects such as rain or ice coating roads / robot / surfaces, and different ground cover such as loose gravel, the friction models used will be varied. An explicit range cannot be given as the coefficient of friction will vary for every interface, but it can be assumed that the contact friction properties between the robot and the ground, the ground and environment objects, and the robot and environment objects may vary substantially.

4.4 Other Relevant Parameters

The event designers may include additional parameters to vary between runs, to include obstacle sizing, vision obscuration, simulated interference, or other deviations from the standard configuration. The nature of these parameters will be announced in advance of the VRC event, though the exact parameters may not be revealed prior to entry.

Appendix: Definitions

DARPA Robotics Challenge Website

Application forms and the most authoritative and up-to-date information about the DARPA Robotics Challenge program in general, and the VRC in particular, can be obtained at TheRoboticsChallenge.org.

OCU

Operator Control Unit. This refers to the control station for the robot at the performer site, and for the VRC is equivalent to the Team Offboard Code.

Run

A trial of a simulated task with a particular configuration (including but not limited to starting position, starting orientation, and communication parameters)

Team Onboard Code

Team Onboard Code is hosted on the cloud and represents code that would be loaded onto the robot onboard computer (or a computer tethered to the robot and located in the utility vehicle) in a physical scenario.

Team Offboard Code

Team Offboard Code is located outside of the OSRF-managed VPN controlling the VRC communications parameters, and reflects some combination of code hosted local to a performer site and on the cloud that will be used to operate the robot. It represents the software that would be located at the operator's terminal (as opposed to on-board the robot) in a physical scenario.

VRC

Virtual Robotics Challenge

VPN

Virtual Private Network