International Conference on Artificial Reality and Telexistence
Eurographics Symposium on Virtual Environments (2017)
A. Dey (Editor)

An AR Network Cabling Tutoring System for Wiring a Rack

B. M. Herbert[1] and A. Weerasinghe[2] and B. Ens[1] and M. Billinghurst[1] and G. Wigley[1]

[1]University of South Australia, School of Information and Mathematical Sciences, Australia
[2]University of Adelaide, School of Computer Science, Australia

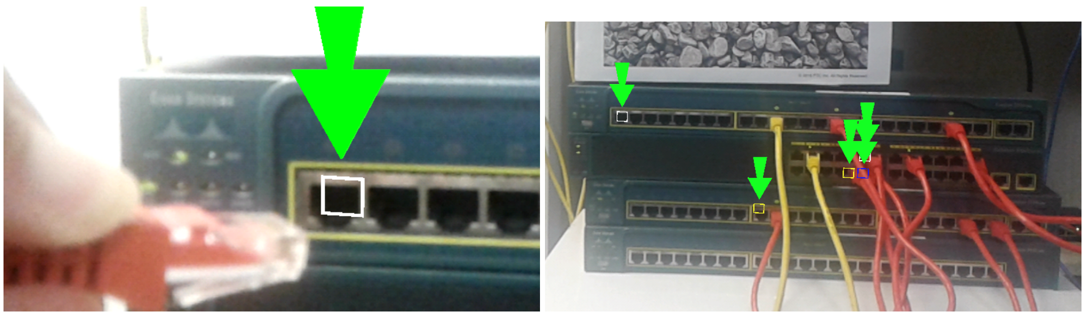

Figure 1: Screenshot of Working AR Interface

Abstract

We present a network cabling tutoring system that guides learners through cabling a network topology by overlaying virtual icons and arrows on the ports. The system determines the network state by parsing switch output and does not depend on network protocols being functional. A server provides a web-based user interface and communicates with an external intelligent tutoring system called The Generalized Intelligent Framework for Tutoring. Users use a tablet to view AR annotations, though support for HoloLens HMD will be added soon.

CCS Concepts

***•Human-centered computing** → **Mixed / augmented reality;** Command line interfaces; Web-based interaction; **•Applied computing** → **Computer-assisted instruction;** **•Networks** → Topology analysis and generation; Physical topologies;*

1. Introduction

We present a practical implementation of a Network Cabling Tutoring System (NCTS) that helps learners cable a data centre rack using Augmented Reality (AR) as shown in Fig. 1. Unlike a simple home network, cabling data centers is often complex due to many cables appearing identical. Using AR can help reduce errors in spatial mapping, reducing cabling mistakes.

Our system detects and updates the network devices states by parsing the console output of each device using the device's serial management port. It uses this information to build the physical topology. Our system does not rely on any functional network protocols and works even if the learner disables all networking on the switch, allowing learners to experiment freely.

Our prototype uses a client/server architecture. An AR client runs on a smart device and connects to a socket server developed in C#. There are three major components to the system: (1) the NCTS server that fetches and updates the network state; (2) the NCTS AR client application developed in Unity3D and runs on an android device and (3) a patched instance of a modular tutoring framework that provides learning support called Generalized Intelligent Framework for Tutoring (GIFT) [SH13]. The C# server fetches learner state information from the GIFT tutoring framework and presents this in JavaScript Object Notation (JSON) format to the AR client.

The server provides a web-based User Interface (UI) to users, which is used to configure switches using a Telnet-like interface, view a visual topology that shows the current state of the network and buttons for rebooting the switches as shown in Fig. 2. Although, our prototype is focused around network cabling tasks, it uses a modular architecture so that it can be easily extended to support other learning domains.

DOI: 10.2312/egve.20171381

2. User Experience

The system consists of three user interfaces: (1) An NCTS AR client, showing the learner where to plug in the cables (Fig. 1); (2) an alternative web-based UI for configuring switches using a Telnet-like console, for viewing a visualisation of the network topology (Fig. 2) and for performing administrative tasks. Lastly, a modified GIFT Authoring Tool (GAT) UI is used by instructors to author the training scenarios for learners.

2.1. NCTS AR Client

Users use an Android tablet, which connects to the network infrastructure using Wifi, to view visual annotations overlaid on the hardware. Feedback is displayed on the tablet screen and is dismissed by tapping on the screen. The android application is a client application developed using Unity3D and listens on TCP port 9001 for messages from the C# parser, such as feedback from GIFT. Tracking capabilities were implemented using Vuforia, a commercial AR tracking solution. When the user points the tablet at the Vuforia marker attached to the physical network rack, annotations appear on the switches, showing the learner which ports need connecting. Initially, Vuforia's object-based feature tracking was used, but was abandoned due to inconsistent tracking performance. The annotations provide learning support and as such, only appear if the learner experiences difficulty completing the activity. This interface is being extended to add support for the Microsoft HoloLens.

2.2. Web User Interface

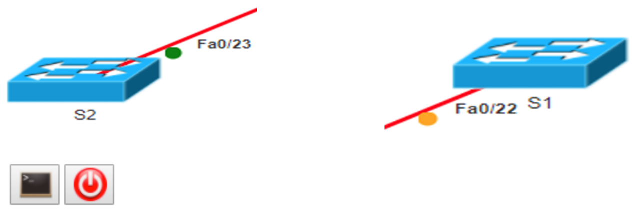

Figure 2: Screenshot of Working Desktop Interface

Users configure the network switches using a Telnet-like console implemented using jQuery Terminal plugin [jQu]. As shown in Fig. 2, users also see a network topology that shows them the current connection state between the nodes. Finally, users can choose to reboot the switch. Other administrative tasks such as showing more complex topology information are being added to the UI.

2.3. Modified GAT

The web-based UI bundled with GIFT was modified to provide form controls for specifying the layout of the topology, setting up the device host names, model names, device configuration and hardware parameters for each of the network switches, so that the tutor knows how to adapt the feedback. The configuration enables a switch to be setup programmatically before it is used for the learning activity.

3. Network Infrastructure

The system consists of a server that runs the C# parser, a Wifi access point that allows tablets and/or wireless displays to wirelessly connect to the infrastructure, an OpenGear console-to-Ethernet management appliance used to logging into the switches and finally, a switch rack that is manipulated by the learners in the learning activity. Standard Cat5 straight through Ethernet cables connects each switch's console port to the OpenGear appliance.

The NCTS server communicates with the GIFT gateway module using the XML-RPC protocol on TCP port 8080. We simplify interoperability by sending C# objects to GIFT as JSON formatted strings. Similarly, feedback from GIFT likewise uses JSON strings input manually into the authoring tool, making it interoperable with GIFT's current implementation.

3.1. Network State Detection

One of the major challenges with the implementation is correctly ascertaining the state of the network and sending this information to a back-end system for processing. Generally, this relies on a functional network to visualise the topology [Ahm09], but for learning, it is preferable to enable the learner to change the network settings. This means that protocols used for updating the topology like Cisco Discovery Protocol (CDP) may not be enabled to provide topology information. Our solution involves using a serial-to-Ethernet management appliance to connect to the switch's console port over Telnet port 600x for each device. This allows the NCTS to continuously read the console output of the switch. The logging messages are then used to process network changes on the device.

4. Conclusion

We developed an AR prototype that reads and interprets the console output of network switches and sends a state message formatted as JSON to GIFT to enable tutoring decisions. GIFT then provides feedback to learners using both the AR interface and the web-based interface. On the desktop, the topology is updated to illustrate network connections. On the tablet, learners who incorrectly connect a cable will see arrows and icons appear on the ports of the switch to provide visual guidance. In our ongoing work, this prototype will be used to evaluate multiple learning dimensions on desktop and AR interfaces [HEB17].

References

[Ahm09] Ahmat K.: Ethernet Topology Discovery: A Survey. *arXiv:0907.3095 [cs]* (July 2009). arXiv: 0907.3095. URL: `http://arxiv.org/abs/0907.3095`. 2

[HEB17] Herbert B., Ens B., Billinghurst M.: An Adaptive AR Tutor For Cabling a Network Topology. In *International Conference on Artificial Reality and Telexistence Eurographics Symposium on Virtual Environments (2017) - Submitted For Review* (Adelaide, Australia, Oct. 2017), The Eurographics Association. 2

[jQu] JQuery Terminal Emulator Plugin. URL: `http://terminal.jcubic.pl/`. 2

[SH13] Sottilare R. A., Holden H. K.: Motivations for a generalized intelligent framework for tutoring (GIFT) for authoring, instruction and analysis. In *AIED 2013 Workshops Proceedings* (2013), vol. 7, p. 1. 1