Aligning the SUNY Poly NCS Program with Nationally Recognized Accreditation

A Master’s Thesis

Presented to

Department of Telecommunications

In Partial Fulfillment

of the Requirements for the

Master of Science Degree

State University of New York

Polytechnic Institute

By

John Cook

January 2015
Aligning the SUNY Poly NCS Program with Nationally Recognized Accreditation

Declaration

I declare that this project is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

John Cook
1/17/2015
SUNY Polytechnic Institute

DEPARTMENT OF

COMPUTER AND INFORMATION SCIENCES

Approved and recommended for acceptance as a thesis in partial fulfillment of the requirements for the degree of Master of Science in Telecommunications

1/29/15
DATE

Dr. John Marsh
Thesis Advisor

Dr. Larry J. Hash

Mr. Ronny Bull
Aligning the SUNY Poly NCS Program with Nationally Recognized Accreditation

Declaration

I declare that this project is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

John Cook
1/17/2015
Abstract

This document is an exploration into what types of curriculum changes must be made to accommodate accreditation. In the review of program accrediting bodies, none is more authoritative or more appropriate than the Accreditation Board for Engineering and Technology (ABET). In ABET’s requirements for accreditation, computing related programs are defined and delineated. On further exploration, it can be seen that the Association for Computing Machinery (ACM) has driven the development of those definitions. The ACM further defines goals and objectives for these disciplines, as well as curriculum models. When reviewing other accreditations, not only are these ACM definitions recognized within those accreditations, goal and outcome alignment is also present. This ‘goal and outcome’ methodology is also present in institution level accreditations that SUNY Poly must comply with. After reviewing the ACM program definitions and comparing them to the NCS program, it is concluded that NCS most closely resembles an ACM IT defined program. This leads to the recommendation of adopting and aligning with ACM IT program guidelines, which provides solutions to multiple program and institution requirements as well as creating a solid pathway to accreditation.
Table of Figures

Figure 1: Table of ACM Recommended Courses and SUNY Poly Equivalents ................................. 28
Chapter One: Introduction

This paper will examine the feasibility and benefits of seeking and obtaining nationally recognized accreditation for the Network and Computer Security (NCS) Bachelor of Science Degree Program at SUNY Polytechnic Institute (SUNY Poly).

1.1 Intended Audience

The intended audience for this paper is SUNY Poly faculty, administrators, and students as well as faculty from other institutions seeking pathways to accreditation.

1.2 Role of Accreditation for Academic Programs

Accreditation adds value to a program and an institution, and, in most cases, some level of accreditation is required for an institution to exist. Having accreditation demonstrates that an outside body recognizes a program or institution as meeting specified criteria. Depending on the accrediting body, criteria may be extremely rigorous, and, in some cases, mandated by law.
The primary academic accrediting body that most public and private colleges adhere to in the mid-eastern part of the United States is the Mid-Atlantic Region Commission on Higher Education, also known as the Middle States Commission on Higher Education (MSCHE). Founded in 1919, this organization is recognized by the U.S. Secretary of Education [i] to conduct accreditation and pre-accreditation (Candidacy status) activities for institutions of higher education ... including distance education and correspondence education programs offered at those institutions. ... and to accredit degree-granting institutions which offer one or more post-secondary educational programs of at least one academic year in length. ... The Commission is a voluntary, non-governmental, membership association that defines, maintains, and promotes educational excellence across institutions with diverse missions, student populations, and resources. It examines each institution as a whole, rather than specific programs within institutions.

Meeting the standards of the MSCHE ensures that an institution is eligible for major sources of state and federal funding. MSCHE’s guiding principles set benchmarks to ensure that member institutions meet academic standards. [ii] Those benchmarks require institutions to

- create plans based on goals and objectives,
- directly and indirectly measure the effectiveness of those objectives,
- continuously strive to improve them.

It should be noted that it is essential that goals and objectives be appropriate, achievable, and measurable.

In addition to MSCHE accreditation, SUNY Poly, along with all other schools in the SUNY system, must meet other criteria. One such example of these is contained in Open SUNY. According to its webpage [iii]

Open SUNY is a system-wide effort designed to maximize online-enabled learning opportunities for all SUNY students current and future. Open SUNY, as a concept, seeks to support campuses and faculty in increasing access, completion, and success for their
students. Open SUNY draws on a rich history of innovative instruction within the SUNY system including the SUNY Learning Network (SLN), Empire State College Center for Distance Learning and much more.

With Open SUNY, the SUNY system as a whole has added requirements for individual institutions to be more consistent within the SUNY system in order to provide a consistent student experience for general education classes. These new requirements strongly encourage institutions to collaborate with other institutions to meet these requirements. For example, it is recommended within the entire SUNY system that courses with the same goals have the same title and course objectives.

In recent years, all levels of educational institutions have undergone increased scrutiny; as a result, more stringent criteria have been mandated. A consistent and widely used method for implementing many of these changes is to implement goals and objectives at course, program, and institutional levels. These goals and objectives are then to be documented, proven, and improved through a cycle of assessment and evaluation. These goals and objectives are also expected to be linked to virtually all decisions within the institution including, but not limited to, justification for budgeted items. Goals and objectives set at all levels are often used to verify that an institution is acting in a manner consistent with its own stated mission and with other institutions.

Using nationally and/or internationally recognized accreditation standards helps programs and institution adopt appropriate goals and objectives as well as prove their validity. Appropriately and consistently used, accreditation guidelines assist the programs and institution to meet the requirements of both the SUNY system and MSCHE.

Most accrediting organizations have members drawn from academia, business, and government. This diversity of viewpoints provides goals and objectives that have been developed from a cross-section of stakeholders and provides the most appropriate and comprehensive criteria. Within an academic program,
this increases the likelihood that successful graduates of these programs will have the appropriate skills to be successful in their fields.

1.3 What is the NCS program?

The NCS program emerged from the Bachelor of Science in Telecommunications undergraduate program in 2010 and is separate and distinct from SUNY Poly’s undergraduate programs in Computer Engineering Technology (BS/CET), Computer and Information Science (BS/CSC), and Computer Information Systems (BS/IS). The Telecommunications program originally focused on primarily analog and digital telephony systems, data networks and regulatory issues. As technology progressed and converged over time, telecommunications, and the program, evolved to more broadly include network and computer communications topics. The current program description, from SUNY Poly’s website is described thus [iv]:

The Network and Computer Security program provides graduates with a thorough understanding of information security principles, data communications networks, and computing systems. The core courses provide a firm background in information security, data communications networks, and computer system security, while the wide range of electives enables a customized program to be developed.

1.4 A move toward goal-and-objective based programs

Based on the review of the information from SUNY and MSCHE, it is evident that objective-based methodologies are becoming more frequently mandated and utilized. In order to meet ever-growing concerns that academic programs are appropriate and rigorous, it would be prudent to consider applying this method to the NCS program by seeking a program accreditation that has established and recognized standards and recommendations. This would enable the program to be both more consistent with programs at other educational institutions and to reflect industry needs from large companies where potential graduates may seek employment. Previous B.S. Telecommunication program graduates found employment
with Cisco, IBM, Compaq Computers as well as many others [v]. A valid approach would be to look at accreditation programs that represent and respond to the interests of such companies.

The result of implementing nationally recognized accreditation for the NCS program would provide verification of goals and objectives, providing clear direction

- at the course level for content and resource needs;
- at the department level for staffing and facilities needs;
- at the institution level for alignment of programs with the overall mission of the college.

This will also benefit all aspects of the program by providing opportunity and impetus for continuous improvement through the utilization of direct and indirect measures from external entities, including the perspective accrediting body, to ensure goals are being met.

The SUNY Poly NCS program should align with a nationally recognized curriculum that contains elements consistent with the current NCS program which will enable it to meet accreditation requirements of the institution and seek program accreditation.
Chapter 2: Need for NCS program accreditation

As mentioned in the previous section, accreditation adds value at course, program and institutional levels. The scope of this document is to look at what can specifically be done in regard to the NCS program, therefore institution-wide accreditation will not be considered. Another facet that needs to be considered is that some accreditations will look at a subset of a program, ensuring that classes meet specified knowledge areas. This may not be a preferred method as the NCS program should be as cohesive as possible. It may be more appropriate to consider accreditation that is comprehensive for a complete and consistent program experience. This does not exclude the possibility of seeking more than one accreditation.

2.1 Curriculum as a pathway to accreditation

One of the primary requirements to accreditation is to have appropriate curriculum. Likely, any accreditation would require some changes to the NCS program. Taking into consideration that goals, objectives, outcomes, and a continuous improvement system are a requirement to any curriculum, any possible modifications to meet these requirements must be taken into account. A logical step is to review the curricula to determine how it aligns with an authority in the area of NCS. One organization that appears
to have the most appropriate information when seeking computing related curricula information is the Association of Computing Machinery (ACM).

The ACM, in conjunction with the IEEE and other member societies, publishes countless other scholarly publications, papers and articles, represents authoritative and peer reviewed computing machine subject matter from public and private sector organizations. The collective expertise of organizations that contribute to these works, in turn, contribute to and are also derived from these documents.

When reviewing ACM documents related to computer curricula, none is more comprehensive than the *Computing Curricula Series* [vi]. The series is a collection of documents related to methodology, specific knowledge areas and general information as to computing curriculum. As far back as the late 1980s, the ACM and IEEE-CS have jointly developed curriculum recommendations. Since 2001, the Computing Curricula Series has been a continuation of those recommendations. It is a model for computing curricula in which five distinct computing disciplines have been developed, studied and reviewed by the ACM, IEEE and other experts in the field of computing technology. This collection of reports and documentation is the standard by which guidelines are measured.

### 2.1.2 ACM Overview on Computing Programs in the Computing Curricula Series

According to the Association of Computing Machinery (ACM), there are five major disciplines within the field computing machinery. They are:

- Computer Engineering (CE)
- Computer Science (CS)
- Information Systems (IS)
- Information Technology (IT)
- Software Engineering (SE)

While these are not necessarily going to cover all possible majors at all educational institutions, it covers the vast majority of computing curricula recognized today. These disciplinary delineations are the
result of efforts to reliably classify the essence of what is embodied within computing disciplines and provide appropriate curricula guidelines. It is expected, regardless of the title of a degree program that a majority of available computing-related undergraduate degrees in North America will fall within one of these five disciplines. The ACM also recognizes that this is a changing field and future disciplines may emerge. For the purposes of aligning to the NCS curriculum, this document will use the ACM’s model that has been developed over a period of greater than fifteen years and has contributions from leading computer societies and higher educational institutions. As a result, these guidelines have become the cornerstone for reliable information as to not only curricula content but also a guide to assist in obtaining accreditation.

Based on ACM documentation, it would be appropriate to examine the programs at SUNY Poly to determine how the four-year computing programs that are offered would fit into the ACM’s model. Additionally, observations can be made as to how programs fall into which classifications in an attempt to learn the relationships between programs. Based on that observation, this paper will attempt to verify and make recommendations for adjustments to align the NCS program in an effort to maintain curricula falling within the recognized standards for the purposes of accreditation. In this document, all curricula will be loosely classified however, recommendations will only be made for the NCS program.

In order to make classifications, a brief review of program titles’, descriptions and core classes from within the programs listed under Computer and Information Sciences will be compared to the ACM guidelines. In addition, any recent efforts to align curricula with external guidelines or accreditation will also be recognized.

2.1.2 Overview of ACM recognized programs

The five disciplines previously mentioned have come to be known by the acronyms in the following manner:

- Computer Engineering (CE)
- Computer Science (CS)
These acronym designations can be seen in various documents that describe curricula and other scholarly publications. These are often used to describe the disciplines and are assumed to be understood by the reader. To avoid any confusion, they will be explicitly defined here. One concise definition summary can be found in ACM’s *Computing Curricula 2005: The overview report*. It is a summary describing in a very brief manner, what is expected as outputs of programs defined under ACM guidelines [vii].

- **Computer engineers** should be able to design and implement systems that involve the integration of software and hardware devices;
- **Computer scientists** should be prepared to work in a broad range of positions involving tasks from theoretical work to software development;
- **Information systems** specialists should be able to analyze information requirements and business processes and be able specify and design systems that are aligned with organizational goals
- **Information technology** professionals should be able to work effectively at planning, implementation, configuration, and maintenance of an organization’s computing infrastructure; and
- **Software engineers** should be able to properly perform and manage activities at every stage of the life cycle of large-scale software systems.

In order to classify programs from the SUNY Poly programs listed under Computer and Information Sciences, certain keywords, class titles and core classes will be compared to the ACM definitions. Classification will also utilize declarations within the program descriptions.
2.2 Programs at SUNY Poly

According to the 2014-2015 SUNY Poly Undergraduate Catalog, the programs listed under Computer and Information Sciences include:

- Accelerated B.S./M.S. in Computer and Information Science
- B.S. in Applied Computing
- B.S. in Computer and Information Science
- B.S. in Computer Information Systems
- B.S. in Network and Computer Security

For the purposes of this paper, the Accelerated B.S./M.S. in Computer and Information Science program is a program that includes M.S. content. For this reason, this program will not be classified.

2.2.1 B.S. in Applied Computing

According to the catalog [viii):

The Bachelor of Science program in applied computing prepares the graduate to apply the analytic and programming skills of the science of computing to a cognate intellectual domain. The degree combines the core of the baccalaureate program in computer information systems or computer and information science with strong academic preparation in another area of study. The capstone project requires the student to apply the tools and techniques of science of computing to the cognate area through the design and implementation of a project. The cognate area requirement may be fulfilled by an associate degree in the cognate area, completion of an approved SUNYIT minor, or courses in another area approved by an advisor. With appropriate course selection, the applied computing graduate may continue into the M.S. program in computer and information science.
It also includes, in section III, a two-track course in either CS or IS. Both include CS 240 – Data Structures & Algorithms which would imply that the CS-Track is an ACM CS-related program and the IS-Track an ACM IS-related program.

2.2.2 B.S. in Computer and Information Science

According to the catalog [ix]:

The Bachelor of Science program in computer and information science provides a broad education in major areas of the field. The program, which closely follows the Association of Computing Machinery (ACM) recommendations, gives students the flexibility to concentrate studies according to their interests.

It also includes a list of possible areas of concentration including:

- Information Assurance/Data Security
- Entertainment Computing (including game design and game programming)
- System Administration
- Scientific and Engineering Computing
- Network and Grid Programming
- System Modeling and Simulation
- Information Technology

CS Program Requirements and CS Core and Intermediate Coursework descriptions include required core courses titled CS 240 – Data Structures & Algorithms and CS 249 – Object-Oriented Programming. According to this description, the phrase “closely follows the ACM recommendations” is confusing in that by including the courses above, it appears to clearly fall under the definition of Computer Science, however, under concentrations, Information Technology and System Administration are listed. Information
Technology is, according to the ACM, a related but clearly different discipline. System Administration also falls into a practitioner’s discipline.

It appears from the core requirement of courses and the “CS” in the CS Program Requirements, this program falls within the ACM’s definition of a CS program.

2.2.3 **B.S. in Computer Information Systems**

According to the catalog [x]:

The Bachelor of Science program in computer information systems places an emphasis on business applications of computing. Students acquire basic skills in computer systems areas, including programming, database management, and other business-oriented areas. The program is designed to follow the curricular guidelines of the ACM, which are endorsed by the Association for Information Technology Professionals (AITP). Many graduates who pursue advanced study enter graduate programs in management or business administration. Also, with appropriate course selection, a student in computer information systems may be prepared to continue on into the M.S. program in computer and information science.

It also continues to include, in section III of the program description, to list IS Program Requirements. This and courses in Business and Management appear to have this generally fall under the ACM’s definition of an IS program.

2.2.4 **B.S. in Network and Computer Security**

According to the catalog [xi]:

Cyber Security professionals are in high demand throughout industry and government sectors. SUNYIT’s Network and Computer Security (NCS) program provides graduates with a thorough understanding of the technologies used to provide and secure modern network and computing infrastructures. Core courses provide a firm background in
information security principles, data communications networks, and computer system security, while the wide range of electives enables a customized program to be developed.

In section III, the program description does not include any advanced programming, data structures or algorithms. It also continues on to include intermediate and advanced coursework in Networking Infrastructure, Internetworking and Data Network Design. These courses indicate that the level of depth in networking is sufficient to integrate this technology into a solution. It also is stated that CS courses (under advanced) that could be taken have “prerequisites beyond Cybersecurity degree requirements”

Although the confusion of name of the program has been interchanged throughout synonymously between “Cybersecurity” and “Network and Computer Security”, this program has elements of the in-depth knowledge necessary to design and implement information technology. This coupled with the fact that many core classes contain a practical component, indicates that students are expected to use and apply technical concepts and practices. The only ACM curriculum that has objectives that describe “use, implement, design, apply and practice” is IT. This is an indication that this is a program has elements that fall within the definitions of an ACM IT program.

### 2.3 NCS as an IT program

Based on the review of the catalog description and given the history of the telecommunications program, it is reasonable to conclude that elements of the NCS program fall within the ACM definition of an IT program. Looking at the NCS program from this perspective provides an opportunity to determine what, if any, changes would have to be made to align the goals and knowledge areas to an ACM-defined IT program. It is recognized that an IT designation can present challenges to department values and staffing needs.
2.3.1 Challenges to an IT program

According to the ACM, IT is a relatively new discipline that is often not understood by academia. Despite this fact, it has, over the last ten years, become more prevalent at academic institutions around the world. The definition of IT, however, has not consistently been adopted and, in some cases, IT has not been accepted as a discipline. This and other challenges to IT have been specifically addressed in the Computing Curricula Series, particularly in the Computing Curricula 2005: The Overview Report [xii] and the Information Technology 2008: Curriculum Guidelines for Undergraduate Degree Programs in Information Technology [xiii].

The ACM made many clear declarations about IT, as some of the challenges that IT as a discipline have faced were recognized by the committees of these reports. One of the first topics that is covered under the description of IT is “The Pace of Change in Academia: Disciplines and Available Degrees” which addresses the question of IT as an academic discipline [xiv]. The reports addressed this question in the CC2005 and reiterated it in following reports. An excerpt of the answer to this question was as follows:

Rigor: Planning and managing an organization’s IT infrastructure is a difficult and complex job that requires a solid foundation in applied computing as well as management and people skills. Those in the IT discipline require special skills – in understanding, for example, how networked systems are composed and structured, and what their strengths and weaknesses are. There are important software systems concerns such as reliability, security, usability, and effectiveness and efficiency for their intended purpose; all of these concerns are vital. These topics are difficult and intellectually demanding.

One of the reasons cited for this response was that some in academia question if IT is a subset of CS. This and a host of other issues were topics that were addressed, among which some harsh observations were made. In an attempt to ensure that this document does not miscommunicate the gravity of the language
used in these reports, the following extended quote is necessary to aid readers of this document of the challenges recognized by the committee in the CC2005 report [xv]:

4.3. Institutional Challenges to Diversity

For any college or university trying to come to terms with the new diversity of computing degree programs, there are at least three areas in which effective leadership and a willingness to change are necessary. These areas are faculty development and adaptation, organizational structure, and curricular structure. Each of these areas involves issues that, by their nature, invite polarities of opinion among faculty. As is often the case when issues elicit strong differences of opinion, there are implicit value choices that underlie the explicit issues. Any honest and thorough planning effort concerning an institution’s computing degree programs should not only face these issues themselves but should also examine the fundamental value choices that underlie differences of opinion in each of these three areas.

4.3.1. Faculty Development and Adaptation

When an institution that currently offers a CS degree decides to diversify by expanding its mission to include SE and/or IT programs, it is likely that some difficulty will arise with respect to finding appropriate faculty. After all, for most institutions, CS faculty is the only available computing faculty. Most of the faculty members will naturally be oriented to the CS mission that shaped their own professional growth and development. Suggesting that they broaden their mission to fully encompass the SE and/or IT agenda is likely to produce a mixed reaction along each of the following lines.

• Legitimacy. Some CS faculty will have the view that SE and/or IT have not yet developed to the stage where they can be considered as academic disciplines. Some will argue that SE deserves a course or two within the context of a CS major but does not warrant or require a distinct programmatic focus. Some will argue that IT offers an agenda that is too vocational. Other CS faculty will be more aware of the important social functions provided by each of the scientific (CS), engineering (SE), and practitioner (IT) professions, and therefore will be more inclined to see a need to broaden the computing education agenda. The latter group can play a valuable role in persuading a skeptical faculty of the validity and importance of the engineering perspective that is central to SE and the practitioner perspective that is central to IT.
Preparedness. Most CS faculty will not have the necessary background to immediately teach courses that substantially differentiate SE and IT from CS. This is a natural consequence of the fact that their background is in CS rather than SE or IT. It is important that these issues are not swept under the carpet as faculty shortcomings can be disguised to the detriment of all. This risk is especially present because CS faculty will be knowledgeable about most computing topics but with a CS orientation that falls short of the needs of SE and/or IT students. For example, both CS and IT students need to know about computer networks. Much of the material is equally important to both CS and IT but, after a point, the agenda splits: CS will emphasize underlying models and principles, while IT will emphasize practical application skills related to network management and security. In practice, most CS faculty have never managed a network or been responsible for maintaining network security. Similarly, most CS faculty will be able to teach a CS course about SE but have not had occasion to obtain the knowledge and experience to teach a full-fledged SE agenda that helps students become software engineers. Care must be taken in at least two dimensions.

1. A CS treatment of shared topics masquerading as SE or IT coursework is quite inappropriate;

2. If CS faculty members are to develop the ability to teach SE and IT courses, it is necessary that they be provided substantial resources to help them prepare and adapt to the very different agendas implied by the engineering and practitioner perspectives.

With respect to both legitimacy and preparedness, the core challenge is to enlist faculty support for new programs and to respond by providing faculty with the support they will require. A high-level decision for a broader computing agenda may well be necessary, but it is unlikely to be sufficient. Some faculty members will be more adaptable and more willing to embrace the challenges presented. Successful efforts will identify faculty who want to develop new capabilities and then provide support for them in order to undertake the kind of self-education and preparation that is needed for them to succeed.

It is quite clear from the previous passage that the ACM has defined IT as a practitioner’s discipline and that the clear distinction of that can be met with resistance and misunderstanding from traditional CS faculty. Regardless of the status at SUNY Poly, the need for some of the practical elements of IT is
recognized in the NCS program. This is clearly a step that acknowledges the need for a practical, hands-on approach in the curricula.

2.3.2 Application of ACM recommendations to the NCS program

In the previous section, it is indicated that NCS does, in fact, contain elements of an ACM-defined IT program. It would be appropriate to do a more in-depth comparison of ACM’s IT Body of Knowledge [xvi] and the required NCS courses, noting any deficiencies that may exist. The ACM states that Information Technology is characterized by a combination of theory, practice, knowledge, and skills. It will not necessarily be captured in this comparison; however, it should be an overall theme when developing goals of the NCS program if it is to fall into the ACM-IT definition.

For comparison purposes, it is important to note that the ACM’s core recommendations for IT is a framework that is based on hourly classroom time coverage of topics. It may be difficult, for some courses, to determine amount of time and focus for a particular topic. For the purposes of this comparison, it will be assumed that at least one hour will be used to cover core topics mentioned in the course descriptions or objectives. In some cases, assumptions will be made as to where they likely exist. Some variation is expected due to expertise of individual course instructors and lack of extensive course descriptions and outcomes for some courses.

The ACM IT Body of Knowledge includes core and advanced outcomes. It is expected that advanced outcomes will be fulfilled in some core courses but more so in advanced courses. As advanced electives are not mandatory, and only a selection of them are required, they will not be considered. The following are being considered as core NCS Courses and will not be compared to advanced core outcomes:

- NCS 181 – Introduction to Cybersecurity
- NCS 210 – Network Transmission Technology
- CS 108 – Computing Fundamentals
- CS 307 – UNIX Programming Environment
- IS 310 – Hardware and Network Infrastructure
- IS 315 – Networking and Information Systems
- NCS 320 – Information Assurance Fundamentals
- NCS 330 – Info. Assurance Ethics, Policies & Disaster Recovery
- NCS 350 – Wireless Systems and Security
- MAT 112/15 – Calculus
- STA 100/225 – Statistics
- MAT 115/413 – Finite/Discrete Math
- PHY 101/201 – Physics I
- PHY 201/202 – Physics II

The following table lists the core ACM IT requirements [xvii] and the classes that meet the objectives:

<table>
<thead>
<tr>
<th>ACM RECOMMENDATIONS</th>
<th>SUNY Poly EQUIVALENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITF. Information Technology Fundamentals (25 core hours)</td>
<td></td>
</tr>
<tr>
<td>ITF. Pervasive Themes in IT (17)</td>
<td>NCS 181</td>
</tr>
<tr>
<td>ITF. History of Information Technology (3)</td>
<td></td>
</tr>
<tr>
<td>ITF. IT and Its Related and Informing Disciplines (3)</td>
<td></td>
</tr>
<tr>
<td>ITF. Application Domains (2)</td>
<td></td>
</tr>
<tr>
<td>HCI. Human Computer Interaction (20 core hours)</td>
<td></td>
</tr>
<tr>
<td>HCI. Human Factors (6)</td>
<td>NCS 181</td>
</tr>
<tr>
<td>HCI. HCI Aspects of Application Domains (3)</td>
<td>NCS330</td>
</tr>
<tr>
<td>HCI. Human-Centered Evaluation (3)</td>
<td></td>
</tr>
<tr>
<td>HCI. Developing Effective Interfaces (3)</td>
<td></td>
</tr>
<tr>
<td>HCI. Accessibility (2)</td>
<td></td>
</tr>
<tr>
<td>HCI. Emerging Technologies (2)</td>
<td></td>
</tr>
<tr>
<td>HCI. Human-Centered Computing (1)</td>
<td>NCS330</td>
</tr>
<tr>
<td>IAS. Information Assurance and Security (23 core hours)</td>
<td></td>
</tr>
<tr>
<td>IAS. Fundamental Aspects (3)</td>
<td>NCS 181,NCS320</td>
</tr>
<tr>
<td>IAS. Security Mechanisms (Countermeasures) (5)</td>
<td>NCS 181,NCS320</td>
</tr>
<tr>
<td>IAS. Operational Issues (3)</td>
<td>NCS 181</td>
</tr>
<tr>
<td>IAS. Policy (3)</td>
<td>NCS 181</td>
</tr>
<tr>
<td>Course Title</td>
<td>Credits</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>IAS. Attacks (2)</td>
<td>NCS 181</td>
</tr>
<tr>
<td>IAS. Security Domains (2)</td>
<td>NCS 181</td>
</tr>
<tr>
<td>IAS. Forensics (1)</td>
<td></td>
</tr>
<tr>
<td>IAS. Information States (1)</td>
<td>NCS 181</td>
</tr>
<tr>
<td>IAS. Security Services (1)</td>
<td>NCS 181</td>
</tr>
<tr>
<td>IAS. Threat Analysis Model (1)</td>
<td>NCS 181</td>
</tr>
<tr>
<td>IAS. Vulnerabilities (1)</td>
<td>NCS 181</td>
</tr>
<tr>
<td>IM. Information Management (34 core hours)</td>
<td></td>
</tr>
<tr>
<td>IM. IM Concepts and Fundamentals (8)</td>
<td></td>
</tr>
<tr>
<td>IM. Database Query Languages (9)</td>
<td></td>
</tr>
<tr>
<td>IM. Data Organization Architecture (7)</td>
<td></td>
</tr>
<tr>
<td>IM. Data Modeling (6)</td>
<td></td>
</tr>
<tr>
<td>IM. Managing the Database Environment (3)</td>
<td></td>
</tr>
<tr>
<td>IM. Special-Purpose Databases (1)</td>
<td></td>
</tr>
<tr>
<td>IPT. Integrative Programming &amp; Technologies (23 core hours)</td>
<td></td>
</tr>
<tr>
<td>IPT. Intersystems Communications (5)</td>
<td>IS310</td>
</tr>
<tr>
<td>IPT. Data Mapping and Exchange (4)</td>
<td></td>
</tr>
<tr>
<td>IPT. Integrative Coding (4)</td>
<td></td>
</tr>
<tr>
<td>IPT. Scripting Techniques (4)</td>
<td>CS307</td>
</tr>
<tr>
<td>IPT. Software Security Practices (4)</td>
<td></td>
</tr>
<tr>
<td>IPT. Miscellaneous Issues (1)</td>
<td></td>
</tr>
<tr>
<td>IPT. Overview of Programming Languages (1)</td>
<td>CS108</td>
</tr>
<tr>
<td>MS. Math and Statistics for IT (38 core hours)</td>
<td></td>
</tr>
<tr>
<td>MS. Basic Logic (10)</td>
<td>MAT115</td>
</tr>
<tr>
<td>MS. Discrete Probability (6)</td>
<td>STA100,MAT115</td>
</tr>
<tr>
<td>MS. Functions, Relations and Sets (6)</td>
<td>MAT112,MAT115</td>
</tr>
<tr>
<td>MS. Hypothesis Testing (5)</td>
<td></td>
</tr>
<tr>
<td>MS. Sampling and Descriptive Statistics (5)</td>
<td>STA100</td>
</tr>
<tr>
<td>MS. Graphs and Trees (4)</td>
<td>MAT112</td>
</tr>
<tr>
<td>MS. Application of Math &amp; Statistics to IT (2)</td>
<td>Various</td>
</tr>
<tr>
<td>NET. Networking (22 core hours)</td>
<td></td>
</tr>
<tr>
<td>NET. Foundations of Networking (3)</td>
<td>NCS320,NCS350</td>
</tr>
<tr>
<td>NET. Routing and Switching (8)</td>
<td></td>
</tr>
<tr>
<td>NET. Physical Layer (6)</td>
<td>NCS210</td>
</tr>
<tr>
<td>NET. Security (2)</td>
<td>NCS 181</td>
</tr>
<tr>
<td>NET. Network Management (2)</td>
<td>NCS315</td>
</tr>
<tr>
<td>NET. Application Areas (1)</td>
<td>NCS350</td>
</tr>
<tr>
<td>PF. Programming Fundamentals (38 core hours)</td>
<td></td>
</tr>
<tr>
<td>Course Category</td>
<td>Course Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-----------------------------------------</td>
</tr>
<tr>
<td>PF.</td>
<td>Fundamental Data Structures (10)</td>
</tr>
<tr>
<td>PF.</td>
<td>Fundamental Programming Constructs (10)</td>
</tr>
<tr>
<td>PF.</td>
<td>Object-Oriented Programming (9)</td>
</tr>
<tr>
<td>PF.</td>
<td>Algorithms and Problem-Solving (6)</td>
</tr>
<tr>
<td>PF.</td>
<td>Event-Driven Programming (3)</td>
</tr>
<tr>
<td>PT.</td>
<td>Platform Technologies (14 core hours)</td>
</tr>
<tr>
<td>PT.</td>
<td>Operating Systems (10)</td>
</tr>
<tr>
<td>PT.</td>
<td>Architecture and Organization (3)</td>
</tr>
<tr>
<td>PT.</td>
<td>Computing Infrastructures (1)</td>
</tr>
<tr>
<td>PT.</td>
<td>Enterprise Deployment Software</td>
</tr>
<tr>
<td>PT.</td>
<td>Firmware</td>
</tr>
<tr>
<td>PT.</td>
<td>Hardware</td>
</tr>
<tr>
<td>SA.</td>
<td>System Administration and Maintenance (11 core hours)</td>
</tr>
<tr>
<td>SA.</td>
<td>Operating Systems (4)</td>
</tr>
<tr>
<td>SA.</td>
<td>Applications (3)</td>
</tr>
<tr>
<td>SA.</td>
<td>Administrative Activities (2)</td>
</tr>
<tr>
<td>SA.</td>
<td>Administrative Domains (2)</td>
</tr>
<tr>
<td>SIA.</td>
<td>System Integration and Architecture (21 core hours)</td>
</tr>
<tr>
<td>SIA.</td>
<td>Requirements (6)</td>
</tr>
<tr>
<td>SIA.</td>
<td>Acquisition and Sourcing (4)</td>
</tr>
<tr>
<td>SIA.</td>
<td>Integration and Deployment (3)</td>
</tr>
<tr>
<td>SIA.</td>
<td>Project Management (3)</td>
</tr>
<tr>
<td>SIA.</td>
<td>Testing and Quality Assurance (3)</td>
</tr>
<tr>
<td>SIA.</td>
<td>Organizational Context (1)</td>
</tr>
<tr>
<td>SIA.</td>
<td>Architecture (1)</td>
</tr>
<tr>
<td>SP.</td>
<td>Social and Professional Issues (23 core hours)</td>
</tr>
<tr>
<td>SP.</td>
<td>Professional Communications (5)</td>
</tr>
<tr>
<td>SP.</td>
<td>Teamwork Concepts and Issues (5)</td>
</tr>
<tr>
<td>SP.</td>
<td>Social Context of Computing (3)</td>
</tr>
<tr>
<td>SP.</td>
<td>Intellectual Property (2)</td>
</tr>
<tr>
<td>SP.</td>
<td>Legal Issues in Computing (2)</td>
</tr>
<tr>
<td>SP.</td>
<td>Organizational Context (2)</td>
</tr>
<tr>
<td>SP.</td>
<td>Professional and Ethical Issues and Responsibilities (2)</td>
</tr>
<tr>
<td>SP.</td>
<td>History of Computing (1)</td>
</tr>
<tr>
<td>SP.</td>
<td>Privacy and Civil Liberties (1)</td>
</tr>
<tr>
<td>WS.</td>
<td>Web Systems and Technologies (22 core hours)</td>
</tr>
<tr>
<td>WS.</td>
<td>Web Technologies (10)</td>
</tr>
</tbody>
</table>
The table shows that many objectives are met at the theory and knowledge level. This does not necessarily cover the skills or the practice in these areas. It is essential to also include labs and projects to meet core requirements. A large number of core requirements include the words install, configure, perform, demonstrate, deploy and implement. This prescribes the need for learners to perform hands-on activities to gain proficiency. This is clearly not meant as just a simple prescribed procedural exercise. This is done from a conceptual standpoint, utilizing theory and methodology, to implement solutions.

2.3.3 Deficiencies of the NCS program as an ACM IT program

The previously mentioned hands-on terms often refer to interaction with modern operating systems, networking hardware, software, and applications. In reference to modern operating systems, while UNIX is specifically mentioned in a course title, no mention of Microsoft® operating systems exists. A few specific course objectives mention Windows user accounts and “hacking” Windows, yet no class specifically mentions concepts related to Active Directory, the model which closely follows the x.500 directory services standard and is used as the basis for Microsoft’s Server operating systems network object security.

According to Microsoft [xviii], “Windows Server now holds about 75 percent of the market share for x86 server operating systems.” Any number close to this percentage should be an indication that Microsoft operating systems should be included when meeting many of the objectives prescribed by the ACM’s server administration recommendations, as well as others.

A large number of references to SQL can also be found in the ACM recommendations. This includes references to database theory as well as security and deployment. Neither SQL nor database concepts are
mentioned in any of the NCS classes or curriculum. A Database Programming class is listed as an advanced
elective, however it is also noted that it may have prerequisites, which also implies that no introductory
database concepts are covered.

Web servers, Web driven database concepts, and Web security are also prevalent in the ACM
recommendations. Developing basic web pages, including ones that access databases is mentioned. It
appears that little, if any, of the NCS classes include any Web server, Web service components, or Web
programming.

The ACM also recommends that, late in the curriculum, a major project be completed. The rationale is
that by the very definition of IT is that IT practitioners apply theory, knowledge, skill, and practice to
implement solutions. In order to experience the entire process, it is necessary to complete the process to
better understand all of the aspects of such an undertaking.

2.4 NCS 450 - Network Security - Case Study

One example of the structure of a practical lab environment that provides a good example of this is the
NCS – 450 Network Security class. This is an elective class for the NCS program. The objectives for this
class are [xix]:

- Describe in detail the fundamental concepts and principles of computer and
  network security
- Identify and assess different types of security threats, spyware, viruses and
  vulnerabilities in computer networks
- Explain the concept of web security, several security implementations such as
  SSL/TLS, HTTPS, and web security vulnerabilities
- Develop an understanding of a variety of cryptographic algorithms and
  protocols underlying network security applications
This is listed as a lecture class however it has a lab component. I had an opportunity to work with Dr. John Marsh in the Spring 2014 semester class. My main responsibilities were to assist students during labs, develop some labs or modify portions of labs. The labs from the previous semester were available and were used as a general topical outline for labs throughout the semester. Dr. Marsh improved many of the labs, as this was the second time that he was teaching this class.

One of the things that came from our initial meeting about the labs is that they were very much procedural and scripted. This was something that we both agreed needed to be modified. One of the big concerns was that the labs being scripted provided “do-able” labs, however they geared as exercises in following a guide and not necessarily learning or applying theory learned in class. It was understood that one of the reasons that the scripting was done to provide the list of appropriate commands to provide desired results. Unfortunately, the large number of possible commands in Cisco routing devices is in the hundreds. This provided a learning challenge in itself. One way of mitigating this issue was that in the initial lab a brief command reference was provided. The rationale was that through clear objectives in the labs, students would look at the overall objectives presented and then make a decision as to how to provide a solution. This made the commands more incidental and emphasized more of the theory and concepts to provide a solution. For this reason, the first two labs were very similar to the labs from the previous semester as part of the goal is for students to understand how to use the configuration interface, commands, and how equipment was connected.

After the first two labs, some of the labs were introduced with a diagram of the overall procedure and a brief description of the requirements. These encouraged students make decisions about many of the steps before any configuration took place. This also placed more emphasis on the need to understand the theory presented in the lecture that was associated with the lab, enabling students to determine the appropriate action to take and to reference any syntax needed to meet that action. Examples of the previous and updated labs can be seen in Appendix A.
As the class progressed students, through understanding of need rather than script, knew the syntax and concepts to perform regular configuration of operations like assigning IP addresses and interacting with the TFTP server. The later was a problem noted in the previous semester as students had saved and retrieved configurations in such a scripted manner that, late in the previous semester, they were still heavily dependent on the documentation rather than understanding what was happening and how it was done.

The more conceptual approach also exposed some weaknesses in students’ knowledge of fundamental networking, particularly in graduate students. There was some discussion about the need for a pre-requisite class or an entrance exam that ensured students enrolling in the class had the required fundamental knowledge. Most students worked outside of class to review many of the fundamental networking concepts to ensure success. A number of students commented on multiple occasions that they were not prepared for some of the NCS classes they were taking. Most students identified network services like DHCP, DNS, and ARP to be areas in which they had insufficient background knowledge. Other common deficient knowledge areas included Linux commands and advanced switching concepts.

Another interesting development as the semester progressed was that students were reading the objectives and drawing a diagram of the solution. This was something that was encouraged early on. The rationale was to encourage them to use a diagram like an algorithm that included the elements needed to provide a conceptual solution that they could then implement in the router. This certainly was more advantageous as it was easy to then communicate with the students as to why a particular configuration may not work as expected due to a conceptual misunderstanding. This also aided students in releasing them from the bounds of the syntax of this particular brand and version of the hardware used and allowed them to focus on the concepts of what needed to be completed. This provided students with the satisfaction of a problem-solving approach based in theory and knowledge. This then led to them to be able to implement the solution and determine if they were successful. The self-made diagrams also aided in troubleshooting and in
helping them to determine where the problem existed, rather than to try to determine what line of a scripted lab they had mistyped.

One good example of this was in regard to the difference between application of Access Control Lists (ACLs) in a procedural vs. conceptual manner in Lab 5. In previous semesters, students were provided a stepwise procedure to enter ACLs on interfaces and were prescribed not only the scope of the ACL, but also the interface and direction at which they were to be applied. In the updated, more conceptual lab, students started to ask questions that would not have occurred to have been asked, such as: Can the ACL be applied to the physical, VLAN or sub-interfaces? In the previous semester lab, it was just a “given” as to where the ACL was to be applied. With the more open ended “Our goal is to allow your pod PC to access your router via SSH remote login, but to prevent any other PC from doing the same” students started to have questions that had previously not been entertained. Employing deductive reasoning and previous lab knowledge, they were able to make decisions based on fundamental understanding rather than following stepwise procedures. This resulted in students entertaining many possible solutions which would not have been the result with a scripted lab.

Students often expressed that they had determined alternate ways in which they may be able to solve a problem based on theory they had learned in lecture. It was also not uncommon for students to say that some concepts that they had not understood in lectures in this, and other classes, now made sense. It also gave them direction as to what other areas they may need to explore to become successful. For example, a few of the students were able to complete the Cisco CCNA exam outside of class. This was something that they had decided to do independently. When considering the objectives of the class, it was reasonable to conclude that students were able to apply and practice the theory of the lectures, therefore re-enforcing and enhancing their learning experience.

Based on the previous information, many of the core topics that currently exist in the NCS program indicate that students would benefit from a practice-based curriculum.
Chapter 3: What accreditation should be pursued?

Although the ACMs Computing Curricula are built on a rigorous model from which these curriculum recommendations are created, the ACM has not been nor is an organization that provides accreditation. This is somewhat misleading as the ACM is, at many levels, involved as a primary source for other accrediting organizations; a prime example of this is the Accreditation Board for Engineering and Technology (ABET). ABET can and does accredit higher education programs. This leads to exploring what potential ABET accreditation options are possible and the relationship between the ACM and ABET.

According to ABETs website, ABETs mission is stated as [xx]:

ABET serves the public globally through the promotion and advancement of education in applied science, computing, engineering, and engineering technology. ABET:

- Accredits educational programs.
- Promotes quality and innovation in education.
- Consults and assists in the development and advancement of education worldwide.
- Communicates and collaborates with its constituents and the public.
- Anticipates and prepares for the changing educational environment and the future needs of its constituents.
ABET accomplishes its mission through a model where member societies provide policy and strategy, as well as accreditation activities. The member society that is the “Lead Society for Computer Science, Information Systems, Information Technology, Software Engineering” is Computing Sciences Accreditation Board (CSAB).

According to CSAB [xxi]:

CSAB is governed by its Board of Directors whose members are appointed by the member societies, the two largest technical, educational, and scientific societies in computer and computer-related fields: The Association for Computing Machinery, Inc. (ACM) and the IEEE-Computer Society (IEEE-CS). These organizations jointly established CSAB in 1985.

In summary ABET awards accreditation, utilizing CSAB to verify that an institution meets criteria for computer science, information systems, and information technology programs sourced from the ACM curriculum guidelines.

As illustrated above, ABET should be considered as a possible accreditation pathway. In addition to this, another program accreditation body exists in the computer security realm: The National Security Agencies (NSA) Certified Academic Excellence pathway.

3.1 ABET Accreditation

ABET has two main requirements for any program to be accredited. They are the general criteria, which all programs must meet, and a more specific program criteria. The program criteria is applicable if a program is believed to fall under the definition of an ABET recognized program.
3.1.1 ABET General Computing Criteria

ABET’s General Computing Criteria for 2014-15 is as follows [xxii]:

Criterion 1. Students

Student performance must be evaluated. Student progress must be monitored to foster success in attaining student outcomes, thereby enabling graduates to attain program educational objectives. Students must be advised regarding curriculum and career matters.

The program must have and enforce policies for accepting both new and transfer students, awarding appropriate academic credit for courses taken at other institutions, and awarding appropriate academic credit for work in lieu of courses taken at the institution. The program must have and enforce procedures to ensure and document that students who graduate meet all graduation requirements.

Criterion 2. Program Educational Objectives

The program must have published program educational objectives that are consistent with the mission of the institution, the needs of the program’s various constituencies, and these criteria. There must be a documented, systematically utilized, and effective process, involving program constituencies, for the periodic review of these program educational objectives that ensures they remain consistent with the institutional mission, the program’s constituents’ needs, and these criteria.

Criterion 3. Student Outcomes

The program must have documented student outcomes that prepare graduates to attain the program educational objectives. There must be a documented and effective process for the periodic review and revision of these student outcomes.

The program must enable students to attain, by the time of graduation:

(a) An ability to apply knowledge of computing and mathematics appropriate to the discipline

(b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
(c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

(d) An ability to function effectively on teams to accomplish a common goal

(e) An understanding of professional, ethical, legal, security and social issues and responsibilities

(f) An ability to communicate effectively with a range of audiences

(g) An ability to analyze the local and global impact of computing on individuals, organizations, and society

(h) Recognition of the need for and an ability to engage in continuing professional development

(i) An ability to use current techniques, skills, and tools necessary for computing practice.

**Criterion 4. Continuous Improvement**

The program must regularly use appropriate, documented processes for assessing and evaluating the extent to which the student outcomes are being attained. The results of these evaluations must be systematically utilized as input for the continuous improvement of the program. Other available information may also be used to assist in the continuous improvement of the program.

**Criterion 5. Curriculum**

The program’s requirements must be consistent with its program educational objectives and designed in such a way that each of the student outcomes can be attained. The curriculum must combine technical and professional requirements with general education requirements and electives to prepare students for a professional career and further study in the computing discipline associated with the program, and for functioning in modern society. The technical and professional requirements must include at least one year of up-to-date coverage of fundamental and advanced topics in the computing discipline associated with the program. In addition, the program must include mathematics appropriate to the discipline beyond the pre-calculus level. For each course in the major required of all students, its content, expected performance criteria, and place in the overall program of study must be published.
Criterion 6. Faculty

Each faculty member teaching in the program must have expertise and educational background consistent with the contributions to the program expected from the faculty member. The competence of faculty members must be demonstrated by such factors as education, professional credentials and certifications, professional experience, ongoing professional development, contributions to the discipline, teaching effectiveness, and communication skills. Collectively, the faculty must have the breadth and depth to cover all curricular areas of the program.

The faculty serving in the program must be of sufficient number to maintain continuity, stability, oversight, student interaction, and advising. The faculty must have sufficient responsibility and authority to improve the program through definition and revision of program educational objectives and student outcomes as well as through the implementation of a program of study that fosters the attainment of student outcomes.

Criterion 7. Facilities

Classrooms, offices, laboratories, and associated equipment must be adequate to support attainment of the student outcomes and to provide an atmosphere conducive to learning. Modern tools, equipment, computing resources, and laboratories appropriate to the program must be available, accessible, and systematically maintained and upgraded to enable students to attain the student outcomes and to support program needs. Students must be provided appropriate guidance regarding the use of the tools, equipment, computing resources, and laboratories available to the program.

The library services and the computing and information infrastructure must be adequate to support the scholarly and professional activities of the students and faculty.

Criterion 8. Institutional Support

Institutional support and leadership must be adequate to ensure the quality and continuity of the program.
Resources including institutional services, financial support, and staff (both administrative and technical) provided to the program must be adequate to meet program needs. The resources available to the program must be sufficient to attract, retain, and provide for the continued professional development of a qualified faculty. The resources available to the program must be sufficient to acquire, maintain, and operate infrastructures, facilities and equipment appropriate for the program, and to provide an environment in which student outcomes can be attained.

As seen above, the general criteria include requirements for goals and objectives. It also mentions program and course learning outcomes. These are mandatory if seeking any accreditation that ABET offers. This is consistent with practices from SUNY and MSCHE. It is possible for a program to seek accreditation in the general criteria only, if it can demonstrate that it is different from the ABET recognized programs. This is done to allow for emerging disciplines.

3.1.2 ABET Information Technology Criteria

As mentioned in the previous section, the NCS program contained elements of an ACM IT program. This was mainly indicated by its practical component. For comparison purposes, it would be appropriate to provide this criteria as ABET accredits IT based programs.

ABET’s criteria for Information Technology programs are as follows [xxiii]:

Student Outcomes
The program must enable students to attain, by the time of graduation:

(j) An ability to use and apply current technical concepts and practices in the core information technologies. [IT]

(k) An ability to identify and analyze user needs and take them into account in the selection, creation, evaluation and administration of computer-based systems. [IT]

(l) An ability to effectively integrate IT-based solutions into the user environment. [IT]

(m) An understanding of best practices and standards and their application. [IT]
(n) An ability to assist in the creation of an effective project plan. [IT]

Curriculum
Students must have course work or an equivalent educational experience that includes:
   a. Coverage of the fundamentals of
      1. the core information technologies of human computer interaction, information
         management, programming, networking, web systems and technologies. [IT]
      2. information assurance and security. [IT]
      3. system administration and maintenance. [IT]
      4. system integration and architecture. [IT]
   b. Advanced course work that builds on the fundamental course work to provide depth. [IT]

3.2 NSA Accreditation

Another possible accreditation pathway is the National Security Agency (NSA) Centers of Academic Excellence (CAE) in Information Assurance (IA)/Cyber Defense (CD) [xxiv].

The purpose of the National CAE designation program is to promote higher education in IA and CD and prepare a growing number of IA/CD professionals to meet the need to reduce vulnerabilities in the Nation’s networks. The initial National CAE in IA Education (CAE/IAE) program was started by NSA in 1998, with DHS joining as a partner in 2004 in response to the President’s National Strategy to Secure Cyberspace.

This accreditation program also prescribes knowledge areas that have a practical component and content similar to NCS. Unfortunately, this program has requirements at the institutional level that are beyond the scope of this document.
Chapter 4: Recommendations

Prior to any accreditation, an established curriculum must be in place that contains goals, objectives, outcomes, assessments and continuous improvement methods. These can only be established by modifying the curriculum to include these elements. As a result of my research, it is my conclusion that an ABET program specific accreditation is the most desirable to pursue. The most appropriate ABET accreditation to pursue is the IT accreditation, as it has the closest alignment with the current NCS program. To complete this process, all stakeholders must contribute to implement the following changes to the program:

- Adopt ABET IT program criteria for program level student outcomes[xxiii]:
  - An ability to use and apply current technical concepts and practices in the core information technologies.
  - An ability to identify and analyze user needs and take them into account in the selection, creation, evaluation and administration of computer-based systems.
  - An ability to effectively integrate IT-based solutions into the user environment.
  - An understanding of best practices and standards and their application.
  - An ability to assist in the creation of an effective project plan.

- Create mandatory courses must be added based the following topics:
Update all existing classes with goals and outcomes utilizing recommendations from the ACM IT 2008 curriculum guidelines.

Develop instruments for assessment of courses that use methods including:

- Written student evaluations
- In-class observations
- Interviews with students and faculty

Develop a system that uses the assessment data to make course and program improvements.

After the curriculum changes are adopted, assessment must occur before accreditation can be sought out. This will require at least one year of assessments to verify that learning outcomes are being met in all classes.
References


vii Ibid., p 27

http://sunyit.edu/undergraduate/applied_computing

http://sunyit.edu/undergraduate/computer-and-information-science/

x Computer Information Systems Program Description (2015), Retrieved January 4, 2015 from
http://sunyit.edu/undergraduate/cis


xiii Association for Computing Machinery (ACM), IEEE Computer Society. Information Technology 2008: Curriculum Guidelines for Undergraduate Degree Programs in Information Technology.


xvi Op cit., Association for Computing Machinery (ACM), pp. 23-27.

xvii Ibid, p 27.

xviii Microsoft by the Numbers: The Enterprise Cloud (2015), Retrieved January 4, 2015 from

xix 450 Course Syllabus. Personal email from John Marsh, January 2015


Appendix
Lab 2: Campus LAN Configuration

Objective

This lab uses the POD as a portion of a campus network. In this lab you will:

- Review “Basic Access” connectivity for the POD’s router and switch to the TFTP server
- Set the hostname on both the router and switch
- Set up DNS services on both the router and switch
- Set up secure access to the EXEC mode used to configure the router and switch
- Reconfigure the POD PC’s networking parameters, including IP address, subnet mask, default gateway, and DNS server

Introduction

This lab will give you a chance to set up your POD as if it were part of a campus network, where the central campus router is our connection to the rest of the world. In fact this is the situation we are in, except that the “campus network” we will connect to for Internet access is really the CS department’s 10.103.0.0/16 subnet. Our part of the campus network will use the POD’s Class-B subnet with address space 172.16.x.0/24, where x is the pod number as described in the CyberLab Guide.

Here are some new procedures that will be needed in today’s lab:

1) Troubleshooting the Basic Access Setup
   a) The Basic Access setup includes the router and switch. This setup should give you access to the Internet, although SUNYIT ITS (IT Services) blocks ICMP packets at our gateway router, so you cannot ping machines outside of the SUNYIT campus network.
   b) Troubleshooting connectivity involves several steps in general:
      i) First, simply try out the connectivity you expect to work. For example, pinging google.com is not expected to work (see above), however you should be able to ping various CS servers like fang and atlantis. You should also be able to do DNS queries (using nslookup on a Linux PC) for any machine hostname on the Internet, for example to resolve google.com to an IP address. If the connectivity you expect to be working is no working, then it is time to troubleshoot.
      ii) First check that all layer 1 cabling is set up as it should be.
         → Sometimes you will find the problem just by looking at the cabling!
iii) Next, check the configurations on the router and switch. See the list of commands commonly used to show router and switch configurations in the CyberLab Guide. Look at IP addresses, subnet masks, and interface status. 

→ Sometimes you will find the problem simply by looking at the configurations.

iv) The basic idea in troubleshooting your network is to work your way outward, and test connectivity at each step, finally identifying at which step you have a problem.

(1) Router connectivity testing:
   (a) First, ping its WAN-facing interface using that interface’s IP address
   (b) Next, ping your default gateway.

(2) Switch connectivity testing:
   (a) First, try to ping the switch’s own VLAN 1 “virtual” interface
   (b) Next, try to ping the LAN-side interface of the router
   (c) Next, try to ping WAN-side interface of the router
   (d) Next, try to ping the router’s default gateway

(3) POD PC connectivity testing:
   (a) Try to ping the POD PC’s own loopback address first: 127.0.0.1
   (b) Next, try to ping, in order, the same IP addresses used to test switch connectivity.

2) Setting Networking Device Host Names
   a) Setting the host name for your networking device is simple using the hostname command. For example, if you want to set your host name to pod_1_router, you would enter the command

```
hostname router_pod_1
```

3) Setting Up Domain Name Services (DNS)
   a) Setting up DNS on a networking device will allow you to contact various other machines by name rather than by IP address. For example, to contact the server google.com, without DNS you have to know the IP address of a Google server. With DNS properly configured, you could simply refer to google.com, and DNS would resolve the domain name to an IP address for you. This will even work for hosts connected to our LAN. For example, you could refer to a CS server by name rather than knowing its IP address.

   b) Because DNS is a network-layer protocol, the query and response messages are IP packets and are routable. So it is possible to have a DNS server that is not on the local LAN, as long as it is reachable. For example, Google makes two DNS servers publically accessible at IP addresses 8.8.8.8 and 8.8.4.4.

   c) Essentially the same commands can be used to set up DNS on both a switch and router.

   d) The first thing you need is to enable domain lookup for the device. Do this using the following command:

```
ip domain-lookup
```

   e) Next, specify a domain name server. You will need the IP address of the server. Enter the following command to specify your DNS name server:
ip name-server w.x.y.z

where w.x.y.z is the IP address of the DNS server. Before you enter this command, it is a good idea to ping the DNS server to make sure it is available.

f) To make it easier to specify devices by hostname rather than a fully formed domain name, you can specify the domain name for your LAN. For example, instead of specifying the full domain name fang.cs.sunyit.edu, we can simply specify fang, as long as we’ve set cs.sunyit.edu as our default domain. To set the default domain to cs.sunyit.edu, use the following command:

```
ip domain name cs.sunyit.edu
```

With this configured, DNS will append the domain name to your query before requesting name resolution. So if you simply specify “fang”, the DNS query will be run on the domain name fang.cs.sunyit.edu.

g) Instead of specifying a single domain name to append to host names, you can specify a list of domain names. DNS will try these in succession until it finds the correct one. If you specify a domain list, it will override a single domain name set with the IP domain name command shown above. You add names to the list one at a time using the ip domain list command. For example, for our CS network you might want to set up the domain list by entering the commands:

```
ip domain list cs.sunyit.edu
nip domain list suny.edu
```

h) To test whether DNS is running, try pinging a local host by just using the host name. For example:

```
ping fang
```
ii) To see the default gateway, type the following command:

```bash
netstat -nr
```

where the `netstat` command is a utility to view a variety of networking data structures, the `-r` flag will tell it to show us the routing tables, and the `-n` flag tells it to give IP addresses (i.e., “numerical” output) rather than hostnames when specifying routers (including the default gateway). The default gateway IP address will be in the Gateway Column of the output, in the row with Destination that is either “default” or “0.0.0.0”. For example, here a portion of the output of the command executed on fang, showing that the default gateway is at IP address 150.156.192.1:

```
fjam@fang:~>netstat -nr
Routing tables
Internet:
 Destination    Gateway     Flags Refs Use Netif Expire
 default        150.156.192.1 UGS 0 14280332  xn0
127.0.0.1       link#2      UH  0  116809  lo0
<etc.>
```

iii) To see the DNS server’s IP address, look at the contents of the file `resolv.conf` located in the `/etc/` directory. This is one of many configuration files (all with file extension `.conf`) that reside in the `/etc/` directory. To do this enter the following command:

```bash
more /etc/resolv.conf
```

For example, here is what the `resolv.conf` file looks like on `fang`:

```
fjam@fang:~>more /etc/resolv.conf
search cs.sunyit.edu
nameserver 150.156.192.76
nameserver 150.156.192.21
```

iv) Note that more than one name server may be specified in this file. This file also usually contains one or more domain names used to complete the domain name when only a local host name is given. For example, in the above example, `cs.sunyit.edu` is used to help resolve a DNS query on “fang” by first appending the domain name, forming the full domain name `fang.cs.sunyit.edu`.

d) For several of our labs, we will want to move the POD PC onto the local POD subnet. At a minimum, we will need to change the IP address, subnet mask, and default gateway to make this happen. Here are the commands you need to know to change the networking parameters on a Linux PC:
i) To change the IP address and subnet mask of the machine to \texttt{w.x.y.z} and \texttt{p.q.r.s}, respectively, enter the command
\begin{verbatim}
ifconfig eth0 inet w.x.y.z netmask p.q.r.s
\end{verbatim}
ii) The set the IP address of the default gateway to \texttt{w.x.y.z}, enter the command
\begin{verbatim}
rout
ate add default gw w.x.y.z
\end{verbatim}
iii) \textbf{Note:} the default gateway for a PC on our POD LAN will be the LAN-side interface of our router. So we will enter the IP address of \texttt{gi0/1} on the POD router.
iv) \textbf{Note:} You may need to precede these commands with the \texttt{sudo} command in order to make the work with correct privileges.

\section*{Procedures}

\subsection*{Part 1: Basic Access to TFTP Server}

\textbf{1) Establish Console Access to the Switch and Router}
\begin{enumerate}
\item Login to the POD PC with username \texttt{dh1240} and password \texttt{dh1240}.
\item Get \texttt{picocom} working for console access
\end{enumerate}

\textbf{2) Establish Console Access to the Switch and Router}
\begin{enumerate}
\item Login to the POD PC with username \texttt{dh1240} and password \texttt{dh1240}.
\end{enumerate}

\textbf{3) Reestablish Basic Access}
\begin{enumerate}
\item Follow the steps used last week to configure the router and switch for basic access.
\item Router
\begin{enumerate}
\item Reset the router to factory defaults
\item Configure the router interface \texttt{gi0/0} on the CS departments /16 private network
\item Configure the router interface \texttt{gi0/1} on the POD subnet
\item Start routing on the router
\item Add a default path on the router pointing to the gateway
\item Connect the router \texttt{gi0/0} interface to the uplink
\end{enumerate}
\item Switch
\begin{enumerate}
\item Reset the switch to factory defaults
\item Configure the switch \texttt{VLAN 1} interface on the POD subnet
\item Turn on switch port \texttt{gi0/1}
\item Connect the switch uplink port to the router downlink port
\end{enumerate}
\item Test for connectivity
\begin{enumerate}
\item Ping the gateway from both the router
\item Ping the gateway from the switch
\item Troubleshoot the basic access configuration as necessary
\end{enumerate}
Part 2: Set your router and switch hostnames

The lab introduction has instructions on setting the hostname for a router or switch.

1) Set your router's hostname
   a) Set your router's hostname to include your pod number, using the following format, illustrated for POD 8: router_pod_8

2) Set your switch's hostname
   a) Set your switch's hostname to include your pod number, using the following format, illustrated for POD 8: switch_pod_8

Part 3: Password Protect EXEC Configuration Mode

Perform the following steps 1 – 3 on the router.

1) Set the Enable password using the most insecure method
   a) Use the enable password command to set the Enable password to dh1240. See the CyberLab Guide for usage of this command.
   b) Observe that the password is shown in plaintext in the running configuration
   c) Copy down the line of your config file that shows the plaintext password:

2) Set the Enable password using another insecure method
   a) First remove the enable password set in the previous step using the no enable password command. See the CyberLab Guide for usage of this command.
   b) Now turn on password encryption using the service password-encryption command.
   c) Use the enable password command to set the Enable password to dh1240.
   d) Now observe that the password is now shown encrypted in the running configuration. Unfortunately, the encryption algorithm Cisco uses for service password-encryption command has been broken.
   e) Copy down the line of your config file that shows the encrypted password:

3) Set the Enable password using the secure method
   a) First remove the enable password set in the previous step using the no enable password command. See the CyberLab Guide for usage of this command.
   b) Next, disable service password-encryption service using the no form of the command.
   c) Use the enable secret command to set the enable password to dh1240.
   d) Now observe that the password is now shown in the running configuration with a different encryption method. This is finally a secure encryption algorithm from Cisco.
   e) Save your running configuration to the startup configuration.
   f) Save your startup configuration to the TFTP server.
g) Copy down the line of your config file that shows the encrypted password:

4) **Repeat the above steps 1 – 3 on the switch**
   a) Copy down the line of your configuration file that shows the plaintext password:

   b) Copy down the line of your configuration file that shows the insecure encrypted password:

   c) Copy down the line of your configuration file that shows the secure encrypted password:

5) **Save router and switch configuration files to the TFTP server**
   a) Save your router’s running configuration to the startup configuration.
   b) Save your router’s startup configuration to the TFTP server as "lab2". Be sure you save to the correct directory.
   c) Repeat steps 1 and 2 above for the switch.

**Part 4: Enable DNS**

We will enable DNS on both the switch and the router.

1) **See what your POD PC is currently using for a DNS name server**
   a) Use the information in the lab introduction above to find out the IP address for your POD PC’s DNS server. Record the DNS server’s IP address here:

2) **Enable DNS and set up DNS services on your router**
   a) Use the information in the lab introduction to perform the following tasks on the router:
      i) Turn on domain lookup. Record the command you used here:

      ii) Set the DNS server to the same IP address that your POD PC uses. Record the command you used here:

      iii) Set the IP domain name to cs.sunyit.edu. Record the command you used here:

      iv) Verify that DNS is working by pinging atlantis using its hostname:
          *ping atlantis*
          If this ping command does not work, then troubleshoot your configuration.
3) **Enable DNS and set up DNS services on your switch**
   a) Use the information in the lab introduction to perform the following tasks on the **switch**:
      i) Turn on domain lookup. Record the command you used here:

      ________________________________

      ii) Set the DNS server to the same IP address that your POD PC uses. Record the command you used here:

      ________________________________

      iii) Set the IP domain name to `cs.sunyit.edu`. Record the command you used here:

      ________________________________

      iv) Verify that DNS is working by pinging `atlantis` using its hostname:
          ```
          ping atlantis
          ```
          If this `ping` command does not work, then troubleshoot your configuration.

**Part 5: Move the POD PC to the POD switch**

1) **Gain Console Access to the Switch and Router**
   a) First, check that you have not used your CS department login to the POD PC. Instead login with username `dh1240` and password `dh1240`.

   **Note:** If you are logged in with your CS account you may lose the ability to re-authenticate to the machine (e.g., to reclaim your session when screen saver came on, or to invoke the `sudo` command). This is because as we change the way the POD PC is connected to the network, it may be without a connection for sometime while you troubleshoot the connection. During this time, the authentication database for CS accounts may be unavailable. But the `dh1240` account is configured locally on each machine, which means that you will always be able to re-authenticate as needed.

2) **Review your POD’s subnet information**
   a) Information about your POD’s 172.16.x.0/24 subnet is in the CyberLab Guide. Note it is up to you how to allocate the IP addresses in your POD’s /24 network.

   b) What is your POD’s subnet ID?

   ________________________________

   c) What is your the first host address in your POD’s subnet?

   ________________________________

   d) What is your the last host address in your POD’s subnet?

   ________________________________

   e) What is your POD subnet’s broadcast address?

   ________________________________

   f) What IP addresses in your POD’s subnet are already being used? For what?

   ________________________________
Spring 2014 - NCS 450/541

Name ____________________________

3) **Reconfigure POD PC layer 1 connections**
   a) Remove your POD PC's Ethernet connection from the CyberLab uplink port on the patch panel.
   b) Connect your POD PC's Ethernet to one the ports on the switch.

4) **Configure the POD PC with the new IP address and netmask**
   a) Write down the IP address and netmask you will use for your POD PC.

   b) Use the instructions in the lab introduction to change your POD PC's IP address and netmask. Write down the command you used to change the settings.

5) **Configure the POD PC with the new default gateway**
   a) Write down the correct IP address for the new default gateway. See the lab introduction above for details.

   b) Use the instructions in the lab introduction to change your POD PC's default gateway. Write down the command you used to change the setting.

6) **Test POD PC connectivity via the POD’s subnet**
   a) The basic idea to test connectivity is to ping various interfaces along the way to the outside network. Record the various commands you use here, along with your notes:

   __________________________________________
   __________________________________________
   __________________________________________
   __________________________________________
Part 6: Return the POD to its default configuration

We will enable DNS on both the switch and the router.

1) Move the POD PC back to the CS department’s subnet
   a) Return the POD PC’s layer 1 connection to the CS department’s subnet.
   b) Reset the POD PC’s networking configuration back to its default. This is accomplished using the following command:

```
sudo /etc/init.d/net.eth0 restart
```
   c) Verify that your POD PC is now back to its original configuration and has connectivity to the Internet. Record the commands you use to make this verification:

2) Return the switch and router to their factory defaults
   a) Return the switch to factory defaults.
   b) Return the router to factory defaults.

3) Remove layer 1 cabling on the POD
   a) Remove all cables except the POD PC connected using the patch from port 1 to port 5.
This lab will introduce commands to configure the 1921 router to provide NAT functionality. The pod switches will be configured to provide Internet access to attached devices. This will provide Internet access to the pod PC. At the completion of this lab, the pod PC should be able to access the Internet from the internal LAN that will be created for each pod.

The network diagram above depicts a typical NAT (Network Address Translation) configuration. Often, the Internet (WAN) side IP is a public IP address. This is not always the case or even necessary. As seen above, the outside (WAN) side IP is a private IP. Individual pod IPs have been provided in the CyberGuide. You should use this for the outside interface of your router. Those IP addresses are configured and recognized by other routers on the college network. In the last lab, pod PC’s could not reach the Internet because the pod LAN was unable to receive traffic back from the Internet. By utilizing NAT, the pod PC will be able to access the Internet because all Internal (LAN) side IPs will be translated to the outside IP before being forwarded to the next router. Before you begin, you should read and understand the objectives of this lab and fill in the missing IP addresses on YOUR pod diagram.

1. Consult the IP addressing chart from the CyberGuide for the router interface. Place this number in your pod network diagram.
2. Determine your inside IP scheme. As in previous labs, you should use:

   **172.16.P.x**, where “P” is your pod number

   Based on this, you should choose appropriate addresses and add them to your diagram for g0/1, the switch (VLAN1) and the pod PC interfaces.

I. Patch in the network
Obtain the following cables from the front of the room.
1. Qty 1 - 1 foot green patch cable
2. Qty 1 - 2 foot green patch cable
3. Qty 1 - 2 foot yellow patch cable

Cable the pod Cisco devices to create the layout in the lab diagram you created. Only alter cabling where necessary for devices within your pod.

1. Ensure the Pod PC is logged in **BEFORE** it is connected to a switch port
2. Attach the Router external interface (interface g0/0) to a patch panel uplink port
3. Patch the Switch uplink port into the router LAN interface (interface 1). The switch uplink port is on the far right of the device by itself.

### II. Configure the Cisco 1921 router and switch

Upon successful completion of this section, your router will be communicating on the classroom network with an internal LAN created for your pod.

**Procedure Overview:**
1. Connect to the router via console
2. Set your router and switch interfaces with the IPs from the diagram you have completed
3. Verify connectivity
4. Declare inside and outside NAT interfaces on your router
5. Define what IP addresses will be NAT’d
6. Enable NAT
7. Test to see if NAT is functioning correctly on your router
8. Save your final configurations to the TFTP server

**1. Connect to your router via serial console.**
   a. Connect the blue Cisco rollover console cable in to the appropriate patch panel port
   b. Access the router via serial console with the `picocom` utility from your pod PC.
   c. Access the privileged EXEC mode of the device with the `enable` command.

**2. Set your router and switch interfaces with the IPs from the diagram you have completed.**
   a. As in previous labs, configure g0/0 and g0/1. You should use the IPs from the diagram you created at the beginning of this lab. You should also configure DNS resolution on the router using 150.156.192.76 and ensure that the router has a default gateway of 10.103.0.1
   b. Connect to the console port of the switch and set the IP of VLAN 1 to the IP from the diagram you created at the beginning of this lab. You should also configure DNS resolution and ensure that the switch has a default gateway configured correctly.

**HINT:** The default gateway of the switch is the LAN interface of your router.

**3. Verify connectivity.**
   a. Reconnect your console cable to the router.
   b. At this point, you should attempt to ping the default gateway AND your switch. If you can’t, verify your cabling
and router configuration. **Do not continue until this is working correctly.**

c. Connect your console cable to the switch and attempt to ping fang.cs.sunyit.edu. 
   
   Can you ping fang.cs.sunyit.edu? ____________________

4. **Declare inside and outside NAT interfaces on your router.**

   a. Using the commands below, assign your network interfaces as “inside” or “outside” relative to NAT. You should be able to determine this by using your network diagram. You should be in the appropriate interface config to assign these commands.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ip nat outside</td>
<td>Defines the outside interface that NAT should use.</td>
</tr>
<tr>
<td>2</td>
<td>ip nat inside</td>
<td>Defines the inside interface that NAT should use.</td>
</tr>
</tbody>
</table>

5. **Define what IP addresses will be NAT’d.**

   a. The router needs to know what IP addresses should be NAT’d when they pass to the WAN. This is determined by an access list that has to be created. Access lists will be covered in greater detail in later labs. For this lab, follow the commands below to create the access list. Note: For the “P”, you should substitute the appropriate pod number.

   b. **Remain in global configuration mode for step 6.**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>conf t</td>
<td>Enter global configuration mode</td>
</tr>
<tr>
<td>2</td>
<td>access-list 1 permit 172.16.P.0 0.0.0.255</td>
<td>Creates an access list, numbered “1” and sets the list to include all hosts on the 172.16.P.0/24 network to be permitted.</td>
</tr>
</tbody>
</table>

6. **Enable NAT.**

   a. In this next step you will enable NAT. You should still be global configuration mode from the previous step. A few variations of NAT exist, but for the purposes of this lab, we are using the PAT (Port Address Translation) type of NAT. This is also known as “overload”.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ip nat inside source list 1 interface g0/0 overload</td>
<td>Enables NAT with the IP’s defined in access list 1 as the inside addresses to be NAT’d and defines that interface g0/0 should be used as the interface to NAT to. Overload indicates that many IPs will use the IP address(es) on the g0/0 interface.</td>
</tr>
</tbody>
</table>

7. **Test to see if NAT is functioning.**

   a. Now that our router is completely configured, we need to determine if NAT is functioning correctly. When NAT is active, we can view the active translations using the “show” command at the enable prompt. The following should be complete from the router:
8. Save your final configurations to the TFTP server.

   a. Save your router and switch configuration to the TFTP server.

III. Setup and test PC connectivity from your new network

Objectives:
1. Configure IP on your pod PC*
2. Testing your new LAN

1. Configure IP on your pod PC.

   Your pod PC has just moved to a different network with a different IP address scheme. IP addressing for this 
host must now be changed. It will not be able to communicate with the Internet until it is given a proper IP address. 
Before you execute the following commands on your Linux LAN PC, fill in the blanks below with the appropriate address 
information. Hint: The default gateway of your computer should be the LAN side interface of your router.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>sudo ifconfig eth0 inet _____________________ netmask _____________________</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The ifconfig command can be used to change a unix system’s IP address or other setting for the network adapter. This command, like most that alter system settings, typically can only be run by root, the system superuser. Prefixing this command with sudo allows regular users to run it with these superuser privileges. Fill in the first blank with the new IP address of the PC. Consult your pod network diagram for this value. Fill in the second blank with a proper netmask for a class-C subnet.</td>
</tr>
<tr>
<td>2</td>
<td>sudo route add default gw _____________________</td>
<td></td>
</tr>
<tr>
<td></td>
<td>We must set a default gateway so network traffic not destined for nodes on the local network segment can find their next hop. Consult your pod network diagram for this IP address.</td>
<td></td>
</tr>
</tbody>
</table>

2. Testing your new LAN

Network connectivity should be restored now that the pod PC is connected to your new LAN. We must verify this to 
ensure proper network configuration and communication across the switch, router, and now the Internet. Execute the 
following commands on your Linux LAN PC. Fill in the appropriate addresses before starting.
### IV. Explore the NAT table/Lab Questions

**Undergraduate students:** Open a web browser on your pod PC and view the translations on the router. Pause now to call your lab proctor to review your ability to connect to the Internet.

**Graduate Students:** Open a web browser on your pod PC and view the translations on the router. Choose one of the translations and document and describe why it is there. Pause now to call your lab proctor to review your ability to connect to the Internet.

Review the translation tables. It should be noticed that each connection uses a port to complete a connection. What might happen if a large number of hosts were “overloaded” on a single IP and all ports were exhausted? What is the number of ports that you would expect to make this happen? What might you do to prevent this from happening?

**ALL Students:**
1. Theorize what would happen if a host outside your network pinged one of your internal hosts. Will that ping be successful? Why or why not?

2. In section III, step c, you were asked to ping fang.cs.sunyit.edu from the switch. Was it successful? Can you ping it now? Please explain what is happening.
3. Does NAT make troubleshooting harder or easier? Justify your answer.

__________________________________________________________________________________________________

__________________________________________________________________________________________________

__________________________________________________________________________________________________

4. What type of impacts does NAT have on security and complexity of a network?

__________________________________________________________________________________________________

__________________________________________________________________________________________________

__________________________________________________________________________________________________

V. Tear down – Return your pod to a clean state

**Warning:** Lab points will be lost for not clearing cables from stations, not resetting configuration files to defaults, leaving hardware in an unusable state, or not properly disconnecting from your serial session. Keep in mind this hardware is being shared.

**A. Return LAN PC to the classroom network**
The pod PC was temporarily moved to a different subnet to verify network connectivity from your pod LAN. Its IP address was then changed in order to communicate on this alternate subnet. Reconnect the 1 foot yellow PC patch cable to the pod PC port and a gigabit uplink port on the patch panel. Execute the following commands to reset the system’s IP addresses and verify network connectivity.

**HINT:** `sudo /etc/init.d/net.eth0 restart`

**B. Restore blank configurations to the pod Cisco devices**
As in past labs, clear the router and switch configurations.

VI. Lab Checkpoint

Upon completion of this lab, call your lab proctor to:

1. View the output of `dir nvram:` on each device
2. Verify your pod PC’s IP address and network connectivity.
3. Review this lab packet, ensuring all fields and commands missing from boxes above have been written in.
This lab will introduce basic Cisco commands to configure the 1921 router and 2960 switch into a small local network, laying the foundation for future lab exercises. At the completion of this lab, each device will be accessible by its assigned IP address, a LAN will be created for each pod, and a node placed within that LAN will be able to connect to the campus network.

I. Complete the network diagram for your pod

The network diagram accompanying this packet displays an overview of the DH 1240 classroom network and the LAN you will be creating over the course of this lab session. Fill in the missing IP address octets on your pod diagram.

1. Consult the lab Pod IP addressing chart and copy the host IP address for the pod PC and external router interface to your pod network diagram.
2. Write your pod number in the top right corner of your diagram sheet and in the missing third octet fields of the three pod LAN components within the dark grey box.

II. Patch in the network

Obtain the following cables from the front of the room.

1. Qty 1 - 1 foot green patch cable
2. Qty 1 - 2 foot green patch cable
3. Qty 1 - 2 foot yellow patch cable

Cable the pod Cisco devices to create the layout in the lab diagram. Only alter cabling, where necessary, for devices within your pod, those items contained within the light grey box on the lab diagram. Be sure to use the appropriate colored cables. Also, do not yet alter the cabling for the PC.

1. Ensure the Pod PC is connected to a patch panel uplink port
2. Attach the Router external interface (interface 0) to a patch panel uplink port
3. Patch the Switch uplink port (gi0/1) in to the router internal interface (interface 1). The switch uplink port is on the far right of the device by itself.
III. Configure the Cisco 1921 router

Upon successful completion of this section, your router will be communicating on the classroom network with an internal LAN created for your pod. Three configuration files will be saved to the TFTP server: a factory default config, a basic networking config, and the final router configuration for this lab.

Objectives:
1. Connect to the router via console
2. Save a clean base configuration to the device
3. Establish a password for the privileged EXEC account
4. Join the router to the network
5. Configure DNS services
6. Enable routing
7. Save your final configurations to the TFTP server

A. Connect to your router via serial console.
1. Connect the blue cisco rollover console cable in to the appropriate patch panel port
2. Access the router via serial console with the picocom utility from your pod PC. Write picocom command string here
3. Access the privileged EXEC mode of the device with the enable command. At this point, you should not be prompted for a password since no password has been set. Inform your lab proctor if you were.
4. The last character of your login prompt should have changed from a > to a hash symbol (#) to denote operation in privileged EXEC mode.

B. Browse the filesystem and save a base config
The dir command can be used to list the files on the devices. Files are organized by the storage medium type, for example internal flash, nvram, system, and usb flash. Saving a barebones configuration will allow us to easily wipe our configuration when this lab session is complete.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>write memory</td>
<td>Save the factory default configuration currently in memory to the startup-config file in NVRAM</td>
</tr>
<tr>
<td>2</td>
<td>dir ?</td>
<td>List the storage medium types available on the device</td>
</tr>
<tr>
<td>3</td>
<td>dir nvram:</td>
<td>Display all files stored in non-volatile RAM</td>
</tr>
</tbody>
</table>
C. Set a password for the privileged EXEC mode.

Cisco devices implement privilege separation and varying access levels to limit functionality to specific users. General access is granted to the device with use of the console cable. In order to change the configuration of the device, the user must enable privileged EXEC mode. This mode and elevated privileges can be obtained with the `enable` command.

This level of access must be secured by password. Note that for today’s lab, no password was required to access the privileged EXEC mode because one had not yet been set. We will now set this password in order to secure our device.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>config terminal</td>
<td>Enter global configuration mode to change the configuration of your router. Your prompt should now resemble <code>Router(config)#</code></td>
</tr>
<tr>
<td>2</td>
<td>enable secret dh1240</td>
<td>Set the enable password to the string <code>dh1240</code>. This password will be SHA 256 hashed in the configuration file to prevent disclosure.</td>
</tr>
<tr>
<td>3</td>
<td>end</td>
<td>Leave global configuration mode.</td>
</tr>
<tr>
<td>4</td>
<td>show running-config</td>
<td>include enable</td>
</tr>
</tbody>
</table>

Output from the above command should be similar to the following:

```
Router# sh run | inc ^ena
enable secret 4 164pc3NLG8oLs1Dr69M1tWFsmmTVrQEQB8zy7kJ//b.
```

The number in the third field above specifies the hashing algorithm used. 4 specifies SHA256, 5 for salted MD5, and 7 for a proprietary Cisco algorithm. The Cisco algorithm can easily be broken and thus should be avoided. By default, the device will use the most secure algorithm available.
D. Enable router network connectivity

Ensure your router is patched in to a network uplink port on the patch panel, as shown in the attached network diagram and enter the following commands to assign an IP address to its external interface.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>conf t</td>
<td>Enter global configuration mode. Note the abbreviated use of the command.</td>
</tr>
<tr>
<td>2</td>
<td>interface gigabitEthernet 0/0</td>
<td>Enter interface configuration mode, specifying the interface to configure. This command can also be abbreviated as int gi0/0. Your prompt should now resemble <code>Router(config-if)#</code></td>
</tr>
<tr>
<td>3</td>
<td>ip address 10.107.5.____ 255.255.<em><strong><strong>.</strong></strong></em></td>
<td>Bind router interface 0 to the appropriate IP address. Consult the IP address table or your network diagram for the fourth octet. Complete the netmask for a class B subnet.</td>
</tr>
<tr>
<td>4</td>
<td>end</td>
<td>Exit interface configuration mode and return to the standard command prompt. Your prompt should again resemble <code>Router#</code></td>
</tr>
<tr>
<td>5</td>
<td>show running-config int gi0/0</td>
<td>Display the configuration for this router interface. We should see our IP address settings. Also notice the shutdown configuration directive. This interface will be disabled while this in place. We must now remove it.</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Enter global configuration mode.</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>Enter interface configuration mode to configure interface gigabitEthernet 0/0. Your prompt should now resemble <code>Router(config-if)#</code></td>
</tr>
<tr>
<td>8</td>
<td>no shutdown</td>
<td>Disable the <code>shutdown</code> configuration directive by prefixing it with the keyword <code>no</code></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>Exit interface configuration mode and return to the standard command prompt. Your prompt should again resemble <code>Router#</code></td>
</tr>
<tr>
<td>10</td>
<td>ping 10.107.0.1</td>
<td>Test connectivity by sending a series of ICMP ping packets to the 10.107/16 gateway. The output of this command will demonstrate proper configuration and cable placement. An incorrect subnet mask above may cause connectivity problems.</td>
</tr>
</tbody>
</table>
E. Enable DNS on the router

Enabling DNS on the devices will allow other nodes to be referred to by host name instead of IP address.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>conf t</td>
<td>Enter global configuration mode</td>
</tr>
<tr>
<td>2</td>
<td>hostname pod___r</td>
<td>Assign a host name to your router. Fill in the blank with your pod number. This command will also change the text displayed in your prompt.</td>
</tr>
<tr>
<td>3</td>
<td>ip name-server 10.107.0.1</td>
<td>Set the DNS server to use for hostname lookups.</td>
</tr>
<tr>
<td>4</td>
<td>ip domain name gw.cs.sunyit.edu</td>
<td>Specify the domain needed to create a FQDN for your device. The domain for our pod devices is gw.cs.sunyit.edu.</td>
</tr>
</tbody>
</table>
| 5    | ip domain list gw.cs.sunyit.edu
ip domain list cs.sunyit.edu | Specify domains to append to host names when a FQDN is not used for connections to other devices. Including gw.cs.sunyit.edu and cs.sunyit.edu in this list will allow us to more easily connect to other nodes within the classroom network and elsewhere on the DogNET. |
| 6    | end     | Leave global configuration mode. |
| 7    | ping gw | Pinging a node by its host name instead of IP address will demonstrate proper function of these DNS settings. Configuring our device to utilize DNS for IP address lookups will simplify connecting to other hosts. |
**F. Checkpoint – Save your current configuration to the device and to the TFTP server**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>write memory</td>
<td>Save our current running-config, the changes we have made to the router thus far, to the startup configuration file. The running-config file is stored in device RAM. Should the device lose power, all configuration changes made to the running configuration since our last save will be lost. We can harness this fact to return our device to the last saved change/working state if something goes wrong. It is important to understand the difference between the startup-config and running-config files.</td>
</tr>
<tr>
<td>2</td>
<td>copy nvram:startup-config tftp://atlantis/201301/ncs_____/p____/router/basic-networking</td>
<td>Save the current startup configuration file to the TFTP server. Fill in the first blank above with your course number (450 or 541) and the second blank with your pod number.</td>
</tr>
</tbody>
</table>

**G. Bring the second router interface online**

Now that our router is online, let's create a small LAN for our pod.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>conf t</td>
<td>Enter global configuration mode</td>
</tr>
<tr>
<td>2</td>
<td>int gi0/1</td>
<td>Enter interface configuration mode for router interface 1. Notice the abbreviated form of this command.</td>
</tr>
<tr>
<td>3</td>
<td>ip address 172.16.<strong><strong>.1 255.255.</strong></strong>.____</td>
<td>Create an internal LAN for your pod with the given IP address as the default gateway for nodes contained within it. The third octet will correspond to your pod number. Complete the netmask for a class C network.</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Disable the <strong>shutdown</strong> configuration directive.</td>
</tr>
<tr>
<td>5</td>
<td>end</td>
<td>Leave global configuration mode.</td>
</tr>
</tbody>
</table>
H. Enable routing
You should be able to communicate from your router to the campus network once the gateway of last resort is configured. Try to ping fang after completing these steps.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Enter global configuration mode</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>ip routing</td>
<td>Enable routing</td>
</tr>
<tr>
<td>3</td>
<td>ip route 0.0.0.0 0.0.0.0 10.107.0.1</td>
<td>Set the gateway of last resort (default route) for your router, so all traffic not destined for a local subnet will be directed to the upstream gateway.</td>
</tr>
<tr>
<td>4</td>
<td>no cdp run</td>
<td>Disable the Cisco Discovery Protocol, a proprietary protocol that Cisco devices use to identify each other on a LAN segment. For our small networks, having this feature enabled may cause conflicts and be detrimental to functionality.</td>
</tr>
<tr>
<td>5</td>
<td>Leave global configuration mode.</td>
<td></td>
</tr>
</tbody>
</table>

I. Save your configurations to the device and to the TFTP server
Saving our configurations to the TFTP server will allow for easy file edits on the system or easy recovery to the devices during our next lab session so we can continue from where we left off.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Save our current running config, the changes we have made to the router thus far, to the startup configuration file.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>copy nvram:startup-config tftp://atlantis/201301/ncs_____/p____/router/lab2</td>
<td>Save the current startup configuration file, containing all changes made to the router for today's lab, to the TFTP server. Fill in the first blank above with your course number (450 or 541) and the second blank with your pod number.</td>
</tr>
<tr>
<td>3</td>
<td>copy nvram:blank-config tftp://atlantis/201301/ncs_____/p____/router/blank-config</td>
<td>Save the factory default configuration file to the TFTP server in the proper location, naming it blank-config.</td>
</tr>
</tbody>
</table>
J. Verify your router settings

Compare the output of these commands to the screenshot below to confirm proper IP addressing and routing. Keep in mind some of the IP addresses will be different for your pod.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>show ip interface brief</code></td>
<td>Display the router interfaces, their assigned IP address, and status. Confirm the IP address for the two gigabit interfaces and ensure their status is up.</td>
</tr>
<tr>
<td>2</td>
<td><code>show ip route</code></td>
<td>Display the routing table for the device. Again, IP addressing may be slightly different for your pod.</td>
</tr>
</tbody>
</table>

```
vienna:~
pod4#sh ip interface brief
Interface     IP-Address OK? Method Status          Protocol
Emmbedded-Service-Engine0/0 unassigned           YES unset administratively down down
GigabitEthernet0/0          10.107.5.97 YES manual up              up
GigabitEthernet0/1          172.16.4.1  YES manual up              up
pod4#
pod4#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area. * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
        + - repicated route, % - next hop override

Gateway of last resort is 10.107.0.1 to network 0.0.0.0
S*  0.0.0.0/0 [1/0] via 10.107.0.1
    10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C   10.107.0.0/16 is directly connected, GigabitEthernet0/0
L   16.107.5.97/32 is directly connected, GigabitEthernet0/0
    172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C   172.16.4.0/24 is directly connected, GigabitEthernet0/1
L   172.16.4.1/32 is directly connected, GigabitEthernet0/1
pod4#  
```
IV. Configure the Cisco 2960 switch

Objectives:
1. Connect to the router via console
2. Save a clean base configuration to the device
3. Join the switch to your newly created subnet behind the router
4. Establish a password for the privileged EXEC account
5. Save your configurations to the TFTP server

A. Connect to your switch via serial console.
1. Patch the blue cisco rollover console cable in to the appropriate switch port
2. Access the switch via console with the picocom utility.
3. Access the privileged EXEC mode of the device with the `enable` command. At this point, you should not be prompted for a password.
4. The last character of your login prompt should have changed from a `>` to a hash symbol (`#`) to denote operation in privileged EXEC mode.

B. Browse the filesystem and save a base config
The commands for this part are similar to the router. Explore the filesystem and save a factory default configuration, named blank-config, to nvram.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Save the factory default configuration currently in memory to the startup-config file in NVRAM</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>List the storage medium types available on the device</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>Display files stored in non-volatile RAM</td>
</tr>
<tr>
<td>4</td>
<td>Fill in missing command</td>
<td>Back up the current running configuration to NVRAM so we have a clean copy saved before any changes are made. Name your backup file blank-config</td>
</tr>
</tbody>
</table>
C. Set a password for the privileged EXEC mode.  
Cisco devices implement privilege separation and varying access levels to limit functionality to specific users. General access is granted to the device with use of the console cable. In order to change the configuration of the device, the user must enable privileged exec mode. Elevated privileges can then be obtained with the `enable` command. This level of access must be secured by password. Note that for today’s lab, no password was required to access the privileged EXEC mode because one had not yet been set. We will now set a password.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Enter global configuration mode. Your prompt should now resemble <code>Switch(config)#</code></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Set the enable password to the string dh1240. This password will be hashed in the configuration file with SHA 256</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Leave global configuration mode</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>The command <code>show running-config</code> will display the currently active device configuration. The <code>include</code> command will display only lines matching a regular expression, similar to the UNIX command <code>grep</code>. By combining these two commands with the pipe (`</td>
<td>`) symbol, only the configuration line containing our hashed enable password will be displayed.</td>
</tr>
</tbody>
</table>

Output from the above command should be similar to the following:

```
Switch# sh running-config | inc ^ena
enable secret 5 $1$BWHk$SffoyNnweXSlefKPlt1ZD/
```

D. Enable network connectivity for the management interface

Ensure your switch is patched in to an uplink port on the patch panel, as shown in the attached network diagram. VLAN 1 is typically used for device management. Creating a switch virtual interface and assigning it an IP address will allow the switch to communicate on the network.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>conf t</code> Enter global configuration mode. Note the abbreviated use of the command.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td><code>interface vlan 1</code> Enter interface configuration mode, specifying the interface to configure. This command can also be abbreviated as <code>int vl1</code>. Your prompt should now resemble <code>switch(config-if)#</code></td>
<td></td>
</tr>
</tbody>
</table>
3  ip address 172.16.___.2  
    255.255.____.___ 
Bind vlan 1 to the appropriate IP address. Consult your network 
        diagram for the third octet. Complete the netmask for a class C 
        network.

4  end 
Exit interface configuration mode and return to the standard 
    command prompt. Your prompt should resemble **Switch#**

5  ping 172.16.___.1 
Test connectivity by sending a series of ICMP ping packets to the 
    internal router interface. The output of this command will 
    demonstrate proper configuration and cable placement.

---

**E. Enable DNS on the switch**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Enter global configuration mode</td>
<td></td>
</tr>
</tbody>
</table>
| 2    | Assign the host name **pod____isw** to your router. Fill in the blank 
    with your pod number. This command will also change the text 
    displayed in your prompt. | |
| 3    | Set the DNS server to use for hostname lookups to the same one used 
    for the router. | |
| 4    | Specify the domain needed to create a FQDN for your device. The 
    domain for our pod devices is gw.cs.sunyit.edu. | |
| 5    | Specify domains to append to host names when a FQDN is not used 
    for connections to other devices. Including **gw.cs.sunyit.edu** and 
    **cs.sunyit.edu** in this list will allow us to more easily connect to other 
    nodes within the classroom network and elsewhere on the DogNET. | |
| 6    | Leave global configuration mode. | |
| 7    | ping gw | Pinging a node by its host name instead of IP address will 
    demonstrate proper function of these DNS settings. Configuring our 
    device to utilize DNS for IP address lookups will simplify connecting to 
    other hosts. |
| 8    | ping fang | Pinging a node further outside the boundaries of your network will 
    ensure wide area network connectivity. |
F. Checkpoint – Save your current configuration to the device and to the TFTP server

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>write memory</td>
<td>Save our current running config, the changes we have made to the router thus far, to the startup configuration file.</td>
</tr>
<tr>
<td>2</td>
<td>copy nvram:startup-config tftp://atlantis/201301/ncs_____/p____/switch/basic-networking</td>
<td>Save the current startup configuration file, containing all changes made to the switch for today’s lab, to the TFTP server. Fill in the first blank above with your course number (450 or 541) and the second blank with your pod number.</td>
</tr>
<tr>
<td></td>
<td>Fill in missing command</td>
<td>Save the factory default configuration file to the TFTP server in the appropriate location and with the file name blank-config.</td>
</tr>
</tbody>
</table>

G. Verify your switch settings

Compare the output of these commands to the screenshot below to confirm proper IP addressing and connectivity. Keep in mind some of the IP addresses will be different for your pod.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>show int status</td>
<td>Display an overview of the switch interfaces. We are looking for gi0/1 to be connected and on vlan 1.</td>
</tr>
<tr>
<td>2</td>
<td>show run int vlan1</td>
<td>Display the running configuration settings for the vlan1 interface.</td>
</tr>
</tbody>
</table>
V. Test PC connectivity from your new subnet

A. Adjust the PC network cable
The pod PC should currently be patched in to an uplink port on the patch panel, which connects it to the DH1240 room switch. Use a 2 foot patch cable to move this PC to your pod LAN. Moving the PC to a different network will cause a temporary loss of network connectivity that may prevent authentication from working properly.

B. IP address changes
Your pod PC has just moved to a different network with a different IP address scheme. IP addressing for this host must now be changed. It will not be able to communicate with the campus network until it is given a proper IP address. Since user authentication comes from the DogNET servers, an extended network outage may negatively impact your ability to work with the system. Execute the following commands on your linux LAN PC to change its network configuration.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>sudo ifconfig eth0 inet _________________________ netmask ________________________</code></td>
<td>The <code>ifconfig</code> command can be used to change a unix system’s IP address or other setting for the network adapter. This command, like most that alter system settings, typically can only be run by root, the system superuser. Prefixing this command with <code>sudo</code> allows regular users to run it with these superuser privileges. Fill in the first blank with the new IP address of the PC. Consult your pod network diagram for this value. Fill in the second blank with a proper netmask for a class-C subnet.</td>
</tr>
<tr>
<td>2</td>
<td><code>sudo route add default gw ___________________________</code></td>
<td>We must set a default gateway so network traffic not destined for nodes on the local network segment can find their next hop. Consult your pod network diagram for this IP address.</td>
</tr>
</tbody>
</table>
C. Testing your new LAN

Network connectivity should be restored now that the pod PC is connected to your new LAN. We must verify this to ensure proper network configuration and communication across the router, switch, and now the LAN PC. When troubleshooting network connection problems, you should start by attempting to communicate with nodes close to you and work further away to help identify the source of the problem. Execute the following commands on your Linux LAN PC.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ping _________________</td>
<td>Ping the IP address for your pod LAN's router interface. We are starting with the first hop leading towards the internet and referring to it by its IP address so any separate DNS problems do not confuse our diagnosis. Press CTRL-C to stop the ping checks.</td>
</tr>
<tr>
<td>2</td>
<td>ping _________________</td>
<td>Ping the external IP address for your LAN router. Again, refer to this system by its IP address</td>
</tr>
<tr>
<td>3</td>
<td>ping _________________</td>
<td>Consult your pod network diagram to determine the IP address of the next hop towards the campus network and attempt to ping it.</td>
</tr>
<tr>
<td>4</td>
<td>nslookup gw</td>
<td>The <code>nslookup</code> command will perform a DNS query on the given host name. A successful conversion of this host name to an IP address will demonstrate communication with the lab's DNS server.</td>
</tr>
<tr>
<td>5</td>
<td>ping fang</td>
<td>Successful pings to fang by its host name will demonstrate proper functioning of the DNS server and our ability to move traffic past our restricted classroom network and to the campus network.</td>
</tr>
<tr>
<td>6</td>
<td>ping google.com</td>
<td>A ping to an external host will show full network connectivity for our pod LAN pc.</td>
</tr>
</tbody>
</table>
VI. Pod LAN connectivity verification

**Undergraduate students:** Pause now to call your lab proctor to review your ability to ping fang from the LAN PC while it is attached to the pod switch and a node on the pod LAN.

**Graduate Students:** Run the command `ifconfig eth0` on your Linux LAN PC and save its output to a text file. Also include the output of 3 successful ICMP pings from the LAN PC to fang.

VII. Tear down – Return your pod to a clean state

**Warning:** Lab points will be lost for not clearing cables from stations, not resetting configuration files to defaults, leaving hardware in an unusable state, or not properly disconnecting from your serial session. Keep in mind this hardware is being shared.

A. Return LAN PC to the classroom network
The pod PC was temporarily moved to a different subnet to verify network connectivity from your pod LAN. Its IP address was then changed in order to communicate on this alternate subnet. Reconnect the 1 foot yellow PC patch cable to the pod PC port and a gigabit uplink port on the patch panel. Execute the following commands to reset the system’s IP addresses and verify network connectivity.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>sudo /etc/init.d/net.eth0 restart</code></td>
<td>This command will restart networking for the pod system and reset all IP addresses to the system default. It is prefixed with <code>sudo</code> since it must be executed with superuser privileges.</td>
</tr>
<tr>
<td>2</td>
<td>Document the command or steps used to verify the pod PC’s proper IP address and network connectivity.</td>
<td></td>
</tr>
</tbody>
</table>
B. **Restore blank configurations to the pod Cisco devices**
This hardware is being shared between two groups for our class and potentially other classes. We must clear our configurations from the devices when we are done. Since our configs were backed up to the TFTP server, they can easily be retrieved at any time to resume our work.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Fill in missing command</td>
<td>Restore the factory default configuration file from the TFTP server to the switch, saving it to the <em>startup-config</em> file on <strong>NVRAM</strong>.</td>
</tr>
<tr>
<td>2</td>
<td>Fill in missing command</td>
<td>Restore the factory default configuration file from the TFTP server to the router, saving it to the <em>startup-config</em> file on <strong>NVRAM</strong>.</td>
</tr>
<tr>
<td>3</td>
<td><code>delete nvram:blank-config</code></td>
<td>Remove this temporary file from each of the devices</td>
</tr>
<tr>
<td>4</td>
<td><code>reload</code></td>
<td>Display the contents of nvram on each Cisco device. Copy this output to a text editor (such as gedit) to be reviewed by your lab proctor. Graduate students must submit a hard copy of this output.</td>
</tr>
<tr>
<td>5</td>
<td><code>reload</code></td>
<td>Restart each of the Cisco devices using the factory default blank configuration.</td>
</tr>
</tbody>
</table>

C. **Log in to rebooted Cisco devices.**
Your devices may take a few minutes to reboot after issuing the `reload` command. Log in to the privileged EXEC mode of each of your Cisco devices once they are back online. The configuration reset was not successfully completed if you were prompted for a password or if the prompt was not reset. Take corrective action if necessary to ensure the configurations have been cleared. Be sure to log out of each device.

D. **Disconnect from the picocom utility**
When the picocom command is launched, lock files are created so no other users can hijack your session. Other users may have difficulty connecting to the serial console if you do not properly disconnect from the serial session.

E. **Return patch cables**
The 1 foot yellow patch cable providing network uplink to the pod PC shall remain connected. Return all others used for this lab session to the appropriate cable box in the front of the room. Be sure to keep the 1 and 2 foot cables separated. Disconnect the blue Cisco rollover console cable.
VIII. Lab Submission and grading

This electronically submitted lab packet, configuration files on the TFTP server as well as the items below will be used to assess a grade for this lab session. Logs may also be consulted.

**Undergraduate students:** Upon completion of this lab, call your lab proctor to:
1. View the output of `dir nvram` on each device
2. Verify your pod PC’s IP address and network connectivity. Have this address displayed on the screen along with proof of successful ICMP pings to fang
3. Approve this lab packet, ensuring all fields and commands missing from boxes above have been written in.

**Graduate Students:** The following will be due by the start of lecture, Monday, February 4:
1. The output of `dir nvram` on each device.
2. The output of `ifconfig eth0` from the LAN PC while it is attached to the pod switch, thus a node on the pod LAN.
3. The output of 3 ICMP pings from the LAN PC to fang while it is attached to the pod switch, thus a node on the pod LAN.
4. This lab packet with all missing fields and commands filled in.
5. Be sure your pod PC is back on the classroom network and able to communicate with external hosts.

Email your completed lab packet and supporting documents to **merantn**

Completion Date: _________________________

Group members in attendance for this lab session:

_________________________________________________________________

_________________________________________________________________

_________________________________________________________________
SUNY Institute of Technology  
NCS 450/541 Lab 3  

During the previous lab session the pod router and switch were attached to the classroom network and brought online. Now these devices must be secured. This lab exercise will explore a basic Cisco configuration in order to secure the management plane of each. Most concepts and commands for this lab were obtained from Chapter 6 of the CCNA Security book. At the completion of this lab, local users will be created and each device will be accessible for management over the network in a secure fashion. Steps will be taken to ensure synchronized time and both devices will log their events to a central server instead of local console.

I. Patch in the network

Obtain the following cables from the front of the room.  
1. Qty 1 - 1 foot green patch cable  
2. Qty 1 - 2 foot green patch cable  

Follow your pod diagram to connect the router to the classroom network and the switch to your pod LAN.
II. Restore router configuration

Configuration settings established during the previous lab were erased due to device sharing. They must be restored to the router before this lab can be completed. Additionally, base configuration files will be transferred from TFTP and stored on the local device to enable a speedy recovery at the start of future labs and speedy teardown at the conclusion of a lab session.

Objectives:
A. Connect to the router via console
B. Establish network connectivity
C. Transfer a clean and a basic networking configuration to the device
D. Load last week’s lab2 configuration file

A. Connect to your router via serial console.
1. Connect the blue cisco rollover console cable in to the appropriate patch panel port
2. Access the router via serial console with the picocom utility from your pod PC.

Write picocom command string here

3. Access the privileged EXEC mode of the device with the enable command. At this point, you should not be prompted for a password since no password has been set. Inform your lab proctor if you were.
4. The last character of your login prompt should have changed from a > to a hash symbol (#) to denote operation in privileged EXEC mode.
5. Notify your lab proctor if the router was not left in a clean state so corrective steps can be taken. Being prompted for an enable password or having a custom prompt are signs of an existing configuration.

B. Establish network connectivity to the router
The configuration that allowed this router to communicate on the network was previously erased from the device. You have two options to bring your device online:

1) Refer to last week’s lab (section IID) and type in the networking commands to reestablish connectivity
2) Follow these steps below to copy and paste the configuration from atlantis, our TFTP server.

Method 2:
Since our configuration files are plain text, the following steps will retrieve the backup file from atlantis and paste it in to the router. This will offer an alternative to retyping last week’s commands.
Open a second terminal window and connect to atlantis via SSH to retrieve your configuration file. Type the following commands into your atlantis SSH session:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ssh atlantis</td>
<td>Connect to atlantis via SSH</td>
</tr>
<tr>
<td>2</td>
<td>cd /opt/tftp/201301/ncs____/p____/router/</td>
<td>Change directory to the router configuration storage location. Fill in the first blank above with your course number (450 or 541) and the second blank with your pod number.</td>
</tr>
<tr>
<td>3</td>
<td>cat basic-networking</td>
<td>Display the basic-networking text file to the screen.</td>
</tr>
</tbody>
</table>

Highlighting the text of the basic-networking configuration file will copy it to the clipboard. Be sure to only highlight the text of the configuration file.

Enter terminal configuration mode on your Cisco device and paste in the configuration text. You may receive some errors. These can be safely ignored.

C. Verify online status of router interfaces
Verify network connectivity to the router. You may also need to enable the router’s ethernet interfaces. Last week’s lab will contain necessary commands. Document any steps taken to verify or restore network connectivity.
D. Transfer configuration files from TFTP server

Now that the router is online, transferring configuration files from atlantis via TFTP will bring the router fully online and simplify future access to the configuration files. Copies of the basic-network and blank-config files generated last week will be saved to the router for future loading.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Copy the lab2 configuration file from atlantis via TFTP. This file should be stored on the SYSTEM: device with the file name running-config. Transferring the remainder of the lab2 file will merge in the entire configuration and fix the errors we just saw.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Copy the lab2 configuration file from atlantis via TFTP. This file should be stored on NVRAM with the file name ncs_____net. We will use this file to quickly get our devices online next time. Fill in the blank with your course number (450 or 541).</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Copy the blank-config configuration file from atlantis via TFTP. This file should be stored on NVRAM with the file name ncs_____clean. We will use this file to quickly reset the router configuration during tear down. Fill in the blank with your course number (450 or 541).</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>List the nvram: device and verify these configuration files are present</td>
<td></td>
</tr>
</tbody>
</table>

If you do not have either the basic-network or blank-config files on your TFTP directory:

**Undergrads:** Request assistance from your lab proctor if you are unable to rectify the problem.

**Graduate Students:** Copy the files stored in NVRAM: by the undergraduate students.

Ensure router interface gi0/1 is online!
III. Secure the router management plane

Objectives:
A. Enable the network time protocol
B. Adjust system logging
C. Enable SSH remote access
D. Create local users
E. Set a login banner
F. Save new configurations to TFTP

A. Enable NTP
It is important to have synchronized clocks when forensically examining log records from different sources in order to piece together a timeline of events. The Network Time Protocol utilizes a central atomic clock to keep time synchronized across systems.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Enter terminal configuration mode on the router</td>
</tr>
<tr>
<td>2</td>
<td>ntp peer 150.156.192.2</td>
<td>Enable Network Time Protocol (NTP) updates from the CS NTP server. This statement will keep time synchronized across all of our devices.</td>
</tr>
</tbody>
</table>
| 3    | clock timezone EST -5  
clock summer-time EDT recurring | Set the time zone for our area. |
| 4    |         | Leave terminal configuration mode |
| 5    | sh ntp association | This command will verify our NTP settings and communication with the DogNET NTP server |
| 6    | sh ntp status | This command will show whether our clocks are synchronized. It may take a short amount of time for the synchronization process to complete. The first line of output will state synchronization status. |
| 7    | sh clock | Display current date and time |
B. Disable console logging and enable logging to syslog

Logging to the console quickly becomes annoying and disruptive while serving little purpose. Console logs are those being displayed to the screen as we are entering commands. Local logs are still available on demand by using the `show logging` command. In addition to storing local logs, most systems are capable of sending log events to a remote server. If an attacker gains access to a device clearing logs to cover tracks is often an easy task. Sending these log events to a remote server allows a safe storage location for this data. Central log storage also facilitates log analysis either manually or through automated tools to notify an administrator of abnormal events.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Enter terminal configuration mode on the router</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>no logging console</td>
<td>Disable logging to the console. This will prevent log lines from interrupting our typing.</td>
</tr>
<tr>
<td>3</td>
<td>logging 10.107.0.25</td>
<td>Specify the server to send log events</td>
</tr>
<tr>
<td>4</td>
<td>logging facility local6</td>
<td>Specify the syslog facility to use. Syslog messages are broken down into facilities and severities to direct different types of messages to different locations. Typically, a facility refers to the software type generating the message. Examples of built in facilities are authentication, kernel, or mail events. Eight local facilities exist for customized use. By using the local6 facility we can direct all Cisco log events to a single file.</td>
</tr>
<tr>
<td>5</td>
<td>login on-success log login on-failure log</td>
<td>Also log successful logins and failed login attempts to our device</td>
</tr>
<tr>
<td>7</td>
<td>archive</td>
<td>Enter archive configuration mode</td>
</tr>
<tr>
<td>8</td>
<td>log config</td>
<td>Enter configuration change logger configuration mode</td>
</tr>
<tr>
<td>9</td>
<td>logging enable</td>
<td>Enable the logging of configuration changes</td>
</tr>
<tr>
<td>10</td>
<td>notify syslog</td>
<td>Send notifications of configuration changes to a remote syslog server</td>
</tr>
<tr>
<td>11</td>
<td>hidekeys</td>
<td>Suppresses recording of password information in log files.</td>
</tr>
<tr>
<td>12</td>
<td>Leave terminal configuration mode</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>show logging</td>
<td>This command can be used to display current logging settings and view the most recent log events.</td>
</tr>
<tr>
<td>14</td>
<td>show archive log config all</td>
<td>List all the records in the config log</td>
</tr>
</tbody>
</table>
C. Enable SSH remote access to the router

Enabling SSH on the Cisco devices will allow secure management connectivity through the network. Serial connectivity requires proximity to the device, limits the number of devices that can be accessed at one time, and limits the number of connections to the device to 1. Typically, network gear is managed through the network. Once SSH is enabled we won’t have to utilize the console cable and can interact with both devices at once.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Enter terminal configuration mode on the router</td>
</tr>
<tr>
<td>2</td>
<td>crypto key generate rsa</td>
<td>Generate the public/private keypair used by SSH. Answer yes if prompted to replace the keys. Use a modulus bit size of (2048). Respond yes if prompted to replace an existing keypair.</td>
</tr>
<tr>
<td>3</td>
<td>ip ssh version 2</td>
<td>Require the more secure SSH version 2 protocol</td>
</tr>
<tr>
<td>4</td>
<td>ip ssh time-out 60</td>
<td>Disconnect during authentication if no response if received from a client within 60 seconds.</td>
</tr>
<tr>
<td>5</td>
<td>ip ssh authentication-retries 3</td>
<td>Disconnect after 3 login attempts</td>
</tr>
<tr>
<td>6</td>
<td>line vty 0 4</td>
<td>Enter line configuration mode</td>
</tr>
<tr>
<td>7</td>
<td>login local</td>
<td>Instruct the device to obtain remote authentication from local user lists instead of a remote source</td>
</tr>
<tr>
<td>8</td>
<td>session-timeout 15</td>
<td>Set the login session to time out after 15 minutes of inactivity</td>
</tr>
<tr>
<td>9</td>
<td>transport input ssh</td>
<td>Disable insecure telnet, only allowing remote connections over SSH.</td>
</tr>
<tr>
<td>10</td>
<td>Leave terminal configuration mode</td>
<td></td>
</tr>
</tbody>
</table>
D. Test SSH connectivity to the router
The `telnet` command can be used to test the availability of basic TCP services and display information about what is listening on that port. Use the `telnet` command as shown to verify communication with your device on the standard SSH port. This can be done from the Linux command line of your pod station.

```
meranth@tequila:~$ telnet pod88 22
Trying 10.107.5.161...
Connected to pod88.gw.cs.sunyit.edu.
Escape character is '^^]'.
SSH-2.0-Cisco-1.25
^^]
telnet> close
Connection closed.
meranth@tequila:~$
```

Press CTRL-] and type `close` to end the telnet session.

E. Create local users on the router
Local user accounts must be created on the device in order to access it remotely.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Enter terminal configuration mode on the router</td>
</tr>
<tr>
<td>2</td>
<td>username ncs____ password dh1240</td>
<td>Create a local user account with the password dh1240.</td>
</tr>
<tr>
<td>3</td>
<td>do sh run</td>
<td>inc username</td>
</tr>
<tr>
<td>4</td>
<td>service password-encryption</td>
<td>Enable basic password encryption for these user passwords.</td>
</tr>
<tr>
<td>5</td>
<td>username ncs____ password dh1240</td>
<td>Observe the difference. This configuration statement will update the user account to scramble the password with the default Cisco algorithm, as denoted by the number 7 in the fourth field. The Cisco algorithm is easy to crack</td>
</tr>
<tr>
<td>6</td>
<td>do sh run</td>
<td>inc username</td>
</tr>
<tr>
<td>7</td>
<td>no username ncs____</td>
<td>This algorithm is reversible, thus still not secure. Remove the current username so the password can be encrypted using a stronger algorithm.</td>
</tr>
</tbody>
</table>
Recreate the login account using the **secret** keyword instead of **password**. Using secret will cause the password to be hashed using the most secure algorithm available to the device.

Leave terminal configuration mode

Observe the difference. The password now should either be SHA256 (router) or MD5 (switch) hashed, much more secure algorithms.

Create user accounts for each member in your group. Each username shall correspond to your campus login ID and each password will be dh1240.

**F. Log in to your router with SSH**

If SSH access was properly enabled, you should be able to connect to your router via the network instead of the console cable. Open a new terminal window and run the following command: `ssh pod____r -l username`

Replace username with a valid username on the device. The Cisco command `who` will display a list of users currently logged in.

**G. Access atlantis to view logs**

The router should now be sending log events to atlantis, a remote server for central log storage. Access atlantis via ssh and view the aggregated cisco log file for all pods. Syslog log files on unix systems are typically stored in the `/var/log/` directory. View the `/etc/syslog.conf` configuration file and observe the destination file for the `local6` facility.

Write the full path to the Cisco log file here:

```
```

Standard unix tools can be used to view this file:

```
less – view a file one page at a time
grep – searches a file and displays lines which match a particular pattern.
```

Use the `less` command to scan the contents of the Cisco log file.

It is often useful to be able to extract from a log file only the data of interest to us. Extracting log events for a specific host from a large log file of events may make it easier to paint a picture of what has been occurring on that system.

Using the `grep` command, write the command string which will display log events for only your pod devices (both router and switch):
H. Set the device login banner
This message will be displayed prior to SSH login. It is often more difficult to prosecute unauthorized access of computing equipment if such notifications prohibiting unauthorized use is not displayed. It is generally not wise to put too much identifying information in such messages. Doing so will facilitate an attacker’s mapping of your network infrastructure.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Enter terminal configuration mode on the router</td>
</tr>
</tbody>
</table>
| 2    | banner login ;  
     SUNYIT - DogNET ---> ***** RESTRICTED NETWORK *****  
     Management Access Only Unauthorized Use Prohibited!  
     Access Logged Violators will be PROSECUTED!  
     ************************* ; | Set the device login banner to be displayed prior to SSH login. |
| 3    |         | Leave terminal configuration mode |

I. Save your new configuration to the device and to the TFTP server
Save the current configuration to the TFTP server for quick recovery next week.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Save the current running config, changes made to the router thus far, to the startup configuration file.</td>
</tr>
<tr>
<td>2</td>
<td>copy nvram:startup-config tftp://atlantis/201301/ncs____/p____/router/lab3</td>
<td>Save the current startup configuration file, containing all changes made to the router for today’s lab, to the TFTP server. Fill in the first blank above with your course number (450 or 541) and the second blank with your pod number.</td>
</tr>
</tbody>
</table>
IV. Checkpoint: Verify configuration and connectivity

**Undergraduate students:** Log in to the router via ssh and display the contents of **NVRAM:**, the output of the **who** command and the NTP synchronization status. Time should be synchronized with the server by this point. Pause now to call your lab proctor to review this output.

Paste the output into this box:
V. Restore switch configuration

Objectives:
A. Connect to the switch via console
B. Establish network connectivity
C. Save a clean and a basic networking configuration to the device

A. Establish network connectivity to the switch
Either by entering each command manually or copy and paste from the configuration file stored on atlantis.

B. Transfer configuration files from TFTP server
Save the blank-config and basic-networking files to NVRAM; naming them ncs----clean and ncs----net

VI. Secure the switch management plane

The switch must now be configured in a manner similar to the router. Refer to section III for a step by step list of commands.

A. Enable NTP

B. Disable console logging and enable logging to syslog

C. Enable SSH remote access to the switch

D. Create local users on the switch

E. Test SSH connectivity to your switch
Use the host name pod____isw when connecting to the switch via SSH.

F. Set the device login banner

G. Save your new configuration to the device and to the TFTP server
VII. Tear down – Return your pod to a clean state

**Warning:** Lab points will be lost for not clearing cables from stations, not resetting configuration files to defaults, leaving hardware in an unusable state, or not properly disconnecting from your serial session. Keep in mind this hardware is being shared.

**A. Restore blank configurations to the pod Cisco devices**

This hardware is being shared between two groups for our class and potentially other classes. We must clear our configurations from the devices when we are done. Saving our basic startup and teardown configuration files to local NVRAM: will facilitate quick retrieval.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Restore the clean configuration file to the startup-config of the switch. This clean configuration file should currently be located on NVRAM: and named <code>ncs_____ -clean</code></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Restore the clean configuration file to the startup-config of the router. This clean configuration file should currently be located on NVRAM: and named <code>ncs_____ -clean</code></td>
</tr>
<tr>
<td>3</td>
<td><code>reload</code></td>
<td>Restart each of the Cisco devices using the clean configuration.</td>
</tr>
</tbody>
</table>

**B. Log in to rebooted Cisco devices.**

Your devices may take a few minutes to reboot after issuing the `reload` command. Log in to the privileged EXEC mode of each of your Cisco devices once they are back online. The configuration reset was not successfully completed if you were prompted for a password or if the prompt was not reset. Take corrective action if necessary to ensure the configurations have been cleared. Be sure to log out of each device.

**C. Disconnect from the picocom utility**

When the picocom command is launched, lock files are created so no other users can hijack your session. Other users may have difficulty connecting to the serial console if you do not properly disconnect from the serial session.

Write picocom disconnect key sequence here

**D. Return patch cables**

The 1 foot yellow patch cable providing network uplink to the pod PC shall remain connected. Return all others used for this lab session to the appropriate cable box in the front of the room. Be sure to keep the 1 and 2 foot cables separated. Disconnect the blue Cisco rollover console cable.
VIII. Lab Submission and grading

This electronically submitted lab packet, configuration files on the TFTP server as well as the items below will be used to assess a grade for this lab session. Logs may also be consulted.

Upload your completed lab packet to the Angel dropbox created for this lab. Be sure to name your files according to the following naming scheme:

Undergrads: Lab3-P#U.pdf
Grads: Lab3-P#G.pdf

Replace # with your pod number. Add your username at the end of the file name if you did not attend the lab session and are submitting your own packet.

Completion Date: ____________________________

Group members in attendance for this lab session:

____________________________________________________

____________________________________________________

____________________________________________________