Enhanced Methods of Coursework Provision in Computer Networks

Dr WJ Buchanan
School of Computing
Napier University
Edinburgh, UK
w.buchanan@napier.ac.uk

L Saliou
School of Computing
Napier University
Edinburgh, UK
l.saliou@napier.ac.uk

Abstract **Many computing-related programmes and modules have many problems caused with large class sizes, large-scale plagiarism, module franchising, and an increasing requirement from students for increased amounts of hands-on, practical work. This paper presents a practical computer networks module which uses a mixture of on-line examinations and a practical skills-based test to assess student performance. Many academics have observed a strong link between examination and coursework performance, where academically strong students generally work hard for both examinations and courseworks. This paper shows that it is possible to design hand-on practicals which have a strong correlation between performance in practical skills-based tests and in examination-type assessments. The paper shows results which grades students for two on-line, multiple choice tests, and then analyses the average time these students take to complete a practical on-line test. It shows that, on average, the average time taken to complete the test increases for weaker students. Along with this the paper shows that female students in the practical test out-perform male students by a factor of 25%.**

Index Terms— **Computer Networks, Courseworks**

I. Introduction

The growth in student numbers on computing programmes has increased over the past decade. Most departments have seen an increase in the number of students on both their programmes and on modules. This increase has typically increased the staff-student ratio for academics, and the increase in student numbers has not led to a corresponding growth in the number of academics used to support the modules. The area of computing has also seen new areas of technology grow, especially in computer networks and Internet technologies. Unfortunately these factors have caused a rise in the amount of time taken to properly assess the performance of students. Thus, typically, the time spent preparing material, lecturing and taking tutorials has generally decreased as a portion of the module time, and the time spent assessing the module has increased. This is inefficient as a better model would be to spend more time designing assessments, than marking them. Along with this, academic testing has generally moved from exam extensive assessment into an assessment based more based on courseworks. Unfortunately the move to coursework-based testing has proven difficult in the computing discipline. Reasons include:

We would like to acknowledge Prof. Jon Kerridge in the School of Computing at Napier University for his support for academic development.

- **The large sizes of classes**. Computing-related programmes often have class sizes measured in 100s, which makes the marking of courseworks difficult, as it is often a time-consuming task. For example, an academic will take over 40 hours to mark 250 courseworks (assuming that they spend 10 minutes reading each coursework). Thus, this time can thus often be as long as the total time spent on teaching the module.
- **Large-scale plagiarism**. Unfortunately, most academics in computing have seen extensive examples of plagiarism. In the past this was mainly focused on copying from fellow students, or small-scale copying from books. With the Internet, there has been a massive growth in verbatim copying from WWW pages. This is now a problem that is made worse with the large-scale variation of the rules relating to plagiarism across institutions, and even across academic departments within the same institution.
- **Plagarism due to weak coursework specifications**. For courseworks a common complaint is that the coursework specification is not well defined, and does not reward students for well-constructed research.
- **Franchising of modules**. Many universities have franchised modules for HND and degree-level modules. These can cause problems, especially as franchised colleges adjust to marking to the same level as their parent institutions, and acquire confidence in certain technical areas. In many cases the quality of a module at a franchised college can be based on the one-to-one interface between the module leader in the parent institution, and the module leader in the franchised college. If this academic relationship does not work, or if there is not regular communication, the module can hide problems that are not uncovered until it is too late. Problems can also occur when modules are franchised in another country. A good example of this is in modules run in Far East countries, which have different time zones, and typically run a semester system which is offset by many weeks as from a UK-based institution.
- **Relevancy of exams/courseworks**. Many computing-related students now demand practical skills relevant to the jobs market-place, and may devalue academic skills, such as research skills, analysis and design, in favour of the practical skills, such as configuration and programming. Networking is a good example of this, as many students demand a good deal of hands-on experience, such as in the programming of routers and switches. The Cisco Academy [1] has shown that there is

a strong need for skills-based training in an academic environment, but it struggles to deliver the high-level of academic rigor that is required at Level 3 and 4, of degree programmes.

- **Time factors in coursework completion**. Many academics have observed that some students spend far too long in the creation of courseworks, and others spend very little time. It is, of course, difficult to define how much time a student should spend on a coursework. Deadlines have often been extended on coursework submissions due to students spending too much time on their reports (or through bad planning from the student). In a practical assessment students will know themselves when they reach the required level and, when ready, will take the assessment.
- **Support-time for courseworks.** There is typically a large investment in time in both the support of a coursework and in its marking. The more challenging the coursework, typically, the more time that an academic will spend with an individual student discussing its specification and implementation. A practical test which can be practiced in the student's own time would obviously bring benefits in time saved.

An assessment model which allows these problems to be solved yet still achieves a good standard of academic rigor is obviously required. This paper outlines an example of a module which has been refined over the years, and shows that it is possible to create a skills-based assessment which has a strong correlation with academic assessments.

II. COMPUTER NETWORKS MODULE

The School of Computing at Napier University has run a successful Level 3 module in Computer Networks [2] over several years. It is taken by over 250 students, of which around 200 are Napier-based and 50 are from other UK-based institutions, and Malaysia. The module has been refined over the years, so that there is no final exam, as this did not give students much opportunity to gauge their performance on the module, before it ended. Along with this, courseworks were not seen as the most efficient method of assessing student skills, as many students simply used the Internet to get their material, especially in the theory and background sections of the coursework, or cribbed results from other students. Thus it was thus decided that the module would be based on three assessments.

With such numbers of students it was decided that the tests would be taken as on-line tests, that allowed more work to be put into the design of the tests, and less into their actual marking. This also mechanised marks, so that students could not debate the actual marks allocated to their assessment. As a result, there were two on-line tests and a practical skills-based test [3] for network configuration. The first two assessments were carefully created with a defined mix of questions that were graded for difficulty.

III. ACADEMIC TESTING

The students took on-line examinations in weeks 6 and 10, and these were used to grade the students for the final mark. A template of questions was used for the assessment with 35 questions of a defined specification. The questions that the student receive were of the same difficulty level, but each question differs in a key element, such as asking the student to calculate a bit rate for a defined number of transmitted bits over a given time. The number of bits and the time varied over the questions given to the student; but they all have the same level of difficulty.

With the results of the tests it was observed that students who did well in the first assessment also did well in the second one. The students were then graded using: A+ (above 85%), A (76-84%), A- (70-76%), B+ (65-69%), … C- (40-45%) and FAIL (less than 40%). The number of students in each of the bands was:

A+	3	A	22	A–	18
B+	25	B	29	B–	24
C+	33	C	22	C–	16
Fail	30				

which shows a good spread of marks [4]. After interviewing students it was observed that many who had failed the first two tests had worked very little and had not asked for help with problems. A common finding was that they had not read the additional material that they were directed towards. The early assessment of their performance obviously gives the opportunity for students to be reassessed before the end of the module.

IV. SKILLS-BASED ASSESSMENT

The third test was designed as a practical hands-on assessment taken in Week 13 of a 15-week semester. With such large numbers on the module it would not be possible to allow each student to be assessed in a practical way, as Cisco Academy students are, thus an emulator was created to assess practical skills. Figure 1 shows the interface that the students used. The emulator produces a unique test with differing parameters to set each time it is run (this stops students from learning the material in a parrot fashion). The student then configures three networking devices: a router, a switch and a wireless access point. Examples of configuration include:

- Setting the network addresses for all the ports of the devices.
- Setting the routing protocol.
- Setting device names.
- Setting up virtual networks.
- Setting up wireless parameters.

The system then detects if the student has correctly answered, and after 25 correct answers it stops and registers a pass. The time taken to complete the test is registered. This is then used for research purposes to analyse performance, related to the first two on-line tests.

To reduce stress, students can use the emulator to practice on, and take the test only when they are ready. An instructor then enters a password and when they complete the test the time taken and their configuration commands are stored. Students are informed, immediately, that they have passed the test in order to receive quick feedback. Only the top times are published so that high performing students can feel proud of their achievement.

V. RESULTS

The results from the first two tests were graded and were assessed against the average time to complete the on-line skills-based test. The results of this are given in Table 1. While there is variation within the grades for the test, the trend can be easily seen when the A, B and C grades are taken as average values, as given in Table 2. It can thus be seen that there is now a strong correlation between the performance on the academic tests, and the skills-based ones. It can be seen that students who failed the first two tests took nearly 80% more time to complete the skills-based test, on average, as an A-rated student. The largest gap is observed between the B-rated student and the C-rated student, where there is a 45% increase in the average time. This again is an observable attribute, as B-rated students tend to understand many of the principles involved, but maybe do not work as hard and are as motivated, on average, as A-rated students. Students who score C's often struggle to understand the basic concepts of the subject, and do not ask questions, until it is too late (such as after the assessment).

Another observation is that the variation in times is the largest at the lower grades. For example, the FAIL, C-, C and C+ grades have a standard deviation of around 600 seconds, while the A and B grades have a standard deviation of 240 seconds.

Another trend, which is seen from the practical test, is that female students did, on average, much better than male students, where the average time for students was:

Male students (217 students): 898 seconds
Female students (28 students): 716 seconds

This can be contrasted with the average marks for male and female students for the first two tests which were:

Male students: 55.9%
Female students: 55.6%

It can be seen that this is a similar mark. Thus female student outperformed male students by more than 25% in the skills-based test, which is an encouraging result, especially in attaching female students to Information Technology/ Technical subjects.

Table 1: Average time to complete practical test

Grade two tests	*Average Time for practical (sec) [students]*
A+	685 [3]
A	683 [22]
A-	648 [18]
B+	713 [25]
B	793 [29]
B-	760 [24]
C+	1056 [33]
C	1179 [22]
C-	1052 [16]
Fail	1200 [30]

Table 2: Average times for grades

Student grade	*Average time (sec) [min:sec]]*
A	672 [11:12]
B	755 [12:35]
C	1096 [17:55]
Fail	1200 [20:00]

VI. CONCLUSIONS

Computing modules in Higher Education have many challenges, such as the growth in student numbers, and in the large-scale plagiarism from the Internet in courseworks. This has caused academic departments to search for an optimal way to assess students, especially in creating assessment strategies which allow more time to be spent in creating assessments and in teaching students, than in marking student scripts. The move from examinations to coursework-based assessments has not really helped this, as it can take just as long to properly assess courseworks as it does to mark exam scripts. In contrast, the move to on-line examinations brings obvious benefits, as more time can be spent on the design the questions, and virtually no time taken to mark student's work. It also allows more diagnostic work to be completed where weaker students can be quickly identified and given help. On-line tests must, though, be carefully designed and should not fall into the trap of vendor-provided assessment strategies which tend to focus on factual knowledge as the main assessment criteria. If they are not designed correctly, students can learn the answers to the tests by gaining the standard answer, and spend very little time doing background reading. Many vendor-provided trainers have been known to tell their students that they should accept the answer, as it's the one that gives the correct one. The model presented here uses a template of questions, which define each of the questions, each question differing in a key element, such as asking the student to calculate a bit rate for a defined number of transmitted bits over a given time. The number of bits and the time varies over the questions given to the student, but they are all at the same level of difficulty. Thus more time is spent designing the range of questions and on the range of difficulty.

The model defined in this paper is a helpful as it allows students to be graded at an early stage, where the rest of the module can be used to develop practical skills which students find useful in gaining employment. A key element is that the skills-based part is assessed in a formal way with some form of challenge to allow students to stay motivated throughout the module. One thing that was observed was that many students were continually scanning the top times for the skills-based test to check to see if they still had a top time. This generated positive competition where no student felt a fear of failing the assessment, but instead actively competed against others in the time taken.

The results also show that on-line practical tests can be made challenging so it is possible to create assessment strategies which have a strong correlation to student performance in academic tests. From these tests it can be seen that a benchmarking process can be used to grade students for the practical test and these can then be applied for future tests. Continual refinement is obviously required and the tests should be contrasted against traditional exam-based testing.

The use of emulators is beneficial in assessments and in allowing hands-on experience but they cannot fully replace real hands-on with wires and boxes and, if possible, students should also be given some time on the configuration of real-life devices. Also, while determining the time taken to complete an activity is possibly not the most efficient way to grade a student in this area, it at least proves that it can be used as a metric in the testing. At present work is being completed on analysis in the flow of commands that a student enters and marks are allocated to the most logical approach. Along with this a fault emulator has been developed which generates a number of unique faults (Figure 3) [5]. The time taken to debug the faults will also be a good measure of the student's performance on the practical side of networking. Faultfinding and debug/testing are two of the most important skills in the computing industry. Academia typically avoids any mention of them, especially in faultfinding.

While the model presented is not suitable for every subject, it seems to be useful in an area of computing which depends on both academic and practical study. Most academics could identify practical elements in their subject area that could be used for a practical test, such as in a programming exercise for the time taken to determine the fault in a computer program.

Overall, students seem to enjoy the challenge of the on-line practical test and a degree of positive competition was observed. Whether this positive competition would still be relevant if the students were graded for their performance is another area of research.

REFERENCES

[1] http://cisco.netacad.net

[2] http://www.dcs.napier.ac.uk/~bill/cnds.html

[3] http://www.dcs.napier.ac.uk/~bill/network_emulator2.htm

[4] http://www.dcs.napier.ac.uk/~bill/cnds_student_test00.html

[5] http://www.dcs.napier.ac.uk/~bill/network_emulator1_fault.htm

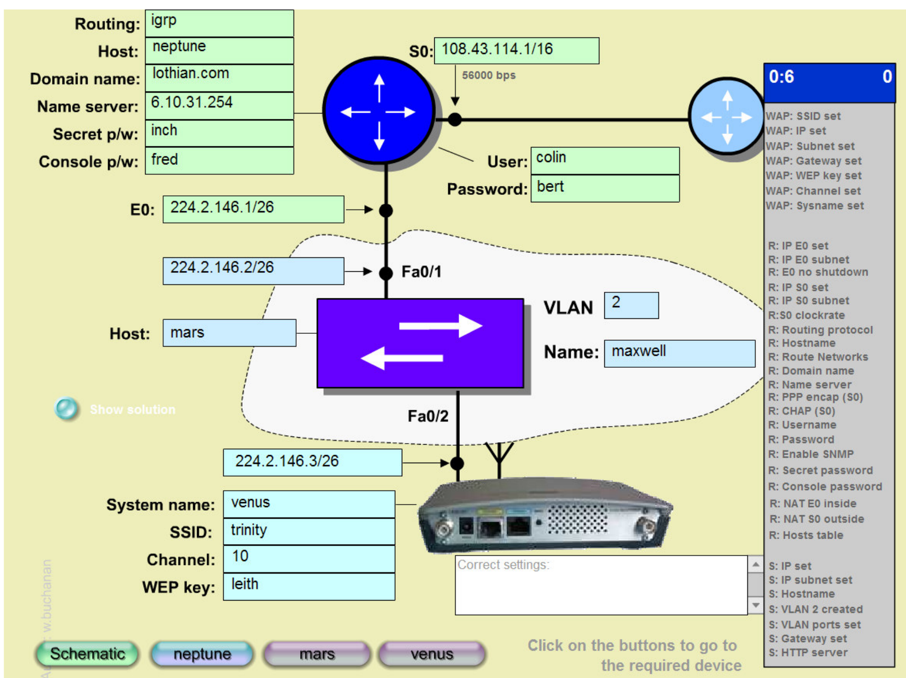

Figure 1: Skills-based networking tests

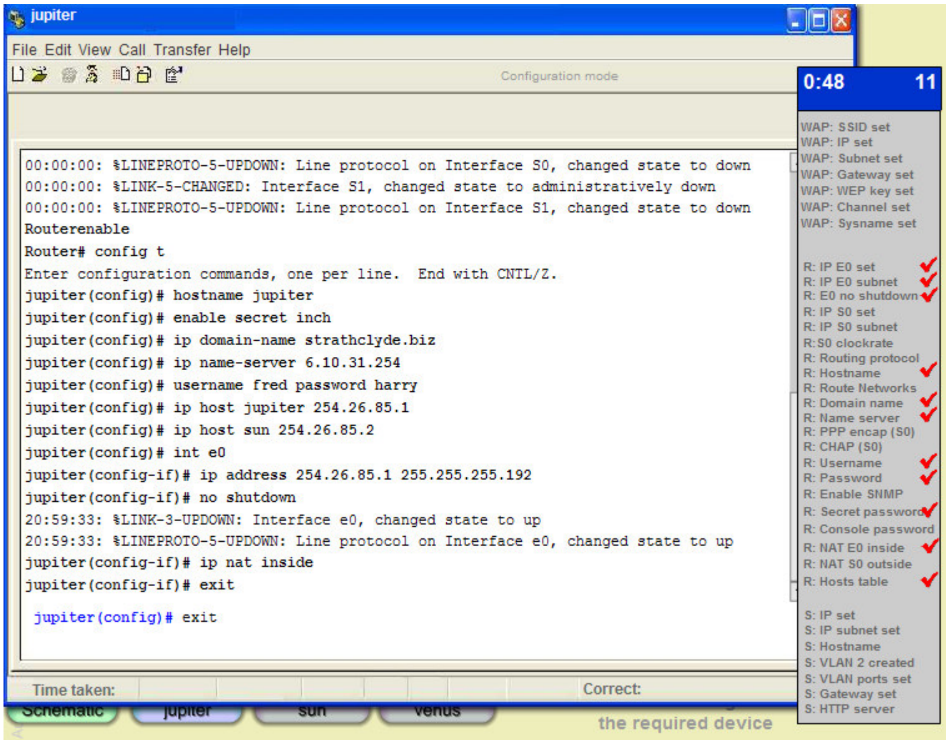

Figure 2: Example of the completion of the test

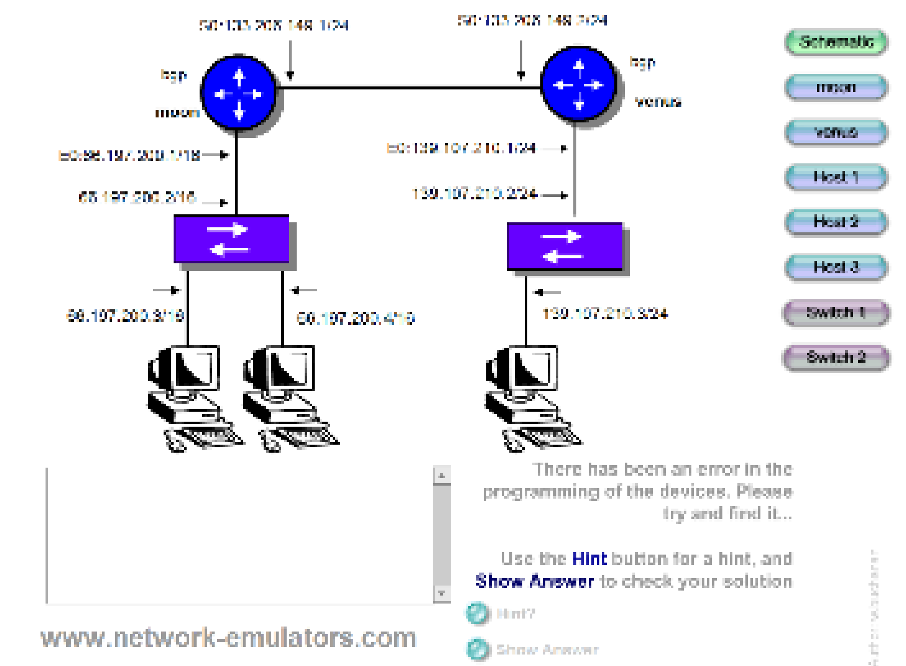

Figure 3: Example showing the variability of the test