Content

How do students understand network protocols? A phenomenographic study

Anders Berglund

Abstract. University students' understanding of network protocols is in focus in this report. With an overall aim to improve learning and teaching in computer systems at a university level, an empirically based study has been performed. In the study, the different ways in which students understand three specific network protocols – TCP, UDP and RMI – as well as the general concept of a network protocol have been investigated with a phenomenographic research approach.

Qualitatively different ways of understanding or experiencing network protocols are discerned. The identified critical differences between the understandings are "how" or "as what" the protocols are understood, "as a part of which framework" the protocols exist, and "in what way" the protocols are described. Although experienced as different, the three protocols are understood as being parts of similarly frameworks.

Recommendations for teaching of computer systems in distributed projects are made, based on the results. Universities should teach computer networks in a way that encourages students to understand network protocols in these critically different ways, and that stimulates them to shift between these ways depending on the task at hand.

1. Introduction to this study

The work presented in this report is a phenomenographic study of students' understanding of network protocols. In this section I will briefly describe the purpose of the study and give an outline of the content of the report

1.1 Purpose of this study

The purpose of this study is to explore university students' understanding of advanced computer science concepts in an internationally distributed project-centred course. In the present report, I will focus on the understanding of network protocols[1], and I will analyse and describe the variations in the ways that the protocols are understood.

The overall objective of my research is to learn about students' learning in computer science, in order to offer possibilities and tools for students, teachers and the universities to improve learning and teaching. While the central issue in this report is variation in *what* the students learn, in my future work, I will study the variation in *how* they learn, and the interaction between these two aspects.

1.2 Structure of this report

The work presented in this report is a phenomenographic study of students' understanding of network protocols. The content and the structure of the report are as follows.

The project-centred course the students are taking as well as its context within the universities, the Runestone initiative is described in section 2. There are technical and pedagogical descriptions of the project the students perform and a brief overview of the computer science concepts that will be in focus of the investigation.

The objectives for using a phenomenographic research approach together with a short overview of some key aspects of phenomenography in this study is presented in section 3 that also points out some methodological issues.

Section 4 presents, based on the data collected, an analysis of different ways of experiencing the three standard network protocols TCP, UDP and RMI in the group of students.

Results on students' understanding of the general concept of a network protocol are presented in section 5. The analysis is based on the results presented in section 4 as well as the empirical data.

In section 6 case studies of individual students' learning are studied in the light of the results presented in earlier sections. These empirically based case studies form the basis for a discussion about students' learning and implications for teaching.

Finally, the report is summarised in section 7.

2. The Runestone initiative, its content and objectives

The research presented in this report is performed within an international networked project, the Runestone initiative. The Runestone initiative, and the research performed within its framework, is centred around an internationally distributed project-centred course in computer systems. This section will focus on the project-based course.

[1] A (computer) *network protocol* is a set of rules that enable communication between computers. See section 2.5 for a further discussion about this concept. The terms *network protocol*, *communication protocol* and *protocol* are used as synonyms in this report.

2.1 *The aims of the student project within Runestone*

The learning that is investigated in this research takes place in a course about distributed computer systems and real-time programming in the Runestone initiative (Daniels, 1999). The students, who are majoring in computer science, take the Runestone course during their third or fourth year at Uppsala University, Uppsala, Sweden and Grand Valley State University, Allendale, MI, USA.

During the course, the students work in internationally distributed teams to jointly develop a software system that is intended to solve a technically advanced computer science task. In the spring term, 2001, when the data was collected for the research that is presented here, the task was to write a program that gives an end-user the possibility to "play" with a Brio labyrinth (see Figure 1).

The labyrinth is a Swedish toy, the aim being to manoeuvre a steel ball from a starting point to a final point on the board, by tilting it so that the ball moves without falling into any of the holes. The original labyrinth has, as is shown in the left picture of Figure 1, knobs that are used to control the angle of the board. The labyrinth that was used was modified to have motors to control the board and a camera to give feedback to the controlling software system, as in the right picture of Figure 1. There were 14 groups of five or six students, each group comprising by students from both universities, collaborating mainly by e-mail and Internet Relay Chat, IRC[2].

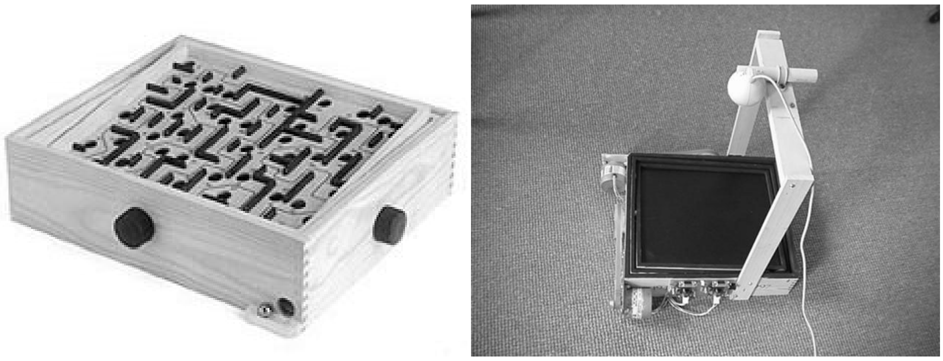

Figure 1 A Brio labyrinth, and a modified version with a camera and motors added

On the Web-page related to the course[3] the students' project was described in the following way:

> This project involves designing and implementing a distributed, real-time system to navigate a steel ball through a board by tilting the surface of the board via positioning motors. The board and ball are a modified version of the well-known Brio Labyrinth game. A monochrome digital video camera focused on the board is available to aid in navigation. The user interface is presented through a web browser. Users who play the game specify a path for the ball to follow, then get feedback on the result of their run.

---

[2] *Internet Rely Chat*, *IRC,* is a system for human communication over Internet. A computer running an IRC program can be used in a way similar to a text telephone and offers to the user a possibility to communicate with any other IRC user in the world.

[3] `http://www.csis.gvsu.edu/class/brio/BrioProject/ProjectDesc/BrioProjectOverview.html`

> This project has elements of real-time control (the Brio game), low-level distributed systems (multiple CPUs to gather data, drive motors), and high-level distributed systems (web interface, network programming), in addition to some demanding requirements on the language used to implement portions of the project (dynamic code loading, security).

As should be clear from the description above, it is a rather large and complex project that the students were given to solve. Several smaller sub-problems had to be solved in order to create the software system that was needed. The results of these smaller tasks were to be integrated to form a working software system. The time for the full task was limited to approximately 8 weeks to fit the universities' requirements on exam periods etc. This period is too short for the students to create a well-functioning software system. Different groups managed to finish different sub-tasks, a result that was expected by both teachers and students.

During the spring term of 2001, the students were given code that had been produced in the previous spring term and were asked to improve it by making three major changes of their own choice. Year 2001 there was one group who managed to complete the task and that produced a working software system that in large corresponded to the specifications, while the other groups presented results that still were not judged to be complete by the teachers.

2.2 The learning objectives from the universities' perspective, the official "what"

Looking at the official documentation at the two universities, descriptions of the course content can be found.

At Grand Valley State University (GVSU) the course is the senior project course for majors. The following course objectives are described:

1. Experience software maintenance and development phases.
2. Integrate experience and knowledge from other courses and apply them to a project.
3. Experience working in a distributed team.

At Uppsala University the Runestone project course is part of a large course that spans over three-quarters of the academic year. The project corresponds to one third of this course, and comes at the end of the full course. It is preceeded by coursework on computer networks, real-time systems and distributed systems.

The aim of the full course is described thus:

> The course provides basic knowledge of the design of distributed systems and their underlying communication subsystems with special focus on real time and embedded applications and control systems.

When the project starts, the students have encountered the teaching about the theoretical aspects of the course content, and have done several smaller practical labs. The course content is described in the following way:

> [...]. Methods for achieving user transparency, eg synchronization, interprocess communication, distributed control and consistency primitives. Time handling, fault tolerance, language support and scheduling for real time control. Case studies.

Neither of the two course descriptions specify the content of the project in any detail. In fact, in the Swedish course description it is not mentioned explicitly, but looking at other official documentation it becomes clear that a project is required, though it is not specified what kind of project is expected.

The educational framework into which the project should fit is set by these descriptions. It should be a senior project for majors, where a software system should be developed that should, according to the GVSU specification, require the application of experience from earlier courses. Uppsala University is more explicit on the content of the project: Computer networks, distributed systems and real time control.

2.3 The collaboration from the universities' perspective, the official "how"

The course objectives, as they are presented by the two universities, do not specify the technical content of the course in detail, and are still more open when discussing how the international project should take place.

The web-site for the course, that was used jointly by the two universities, gives more detailed information about how the course was planned for the spring of 2001, and applies to both universities. It states some major aspects[4]:

> There are two major aspects of this project.
>
> - Developing the software.
> - Building a virtual work team.
>
> Software development involves splitting up the work and allocating it to members of the group, and making sure that your group understands what is happening in the project. Consequently one of the major features of this project is for each group to have a regular contact with one of the teaching staff to report on the progress they are making and to ask questions that might develop.

A total of 96 students participated in the course. All groups except one (that only had Swedish participants) consisted of two to three students from each university, making up to a total of five to six students in each group. Two teachers, one from each university, taught the course in collaboration. There was also technical support with issues like operating systems and practical questions concerning the functioning of the Brio-board. At GVSU this service was offered by the technical staff of the department, while it in Sweden was given as a task to the group that was formed only of Swedish participants.

All interaction with the teachers, whether local or not, as well as the interaction with group members at the other university had to be made using forms of ICT[5], such as chat and e-mail. An initial physical meeting was arranged in Sweden, in the US a few meetings were arranged, mainly to teach Java. Each group of students was assigned a teacher, either in Sweden or in the US. It was decided to keep regular weekly meetings with the teachers, where the groups should report the progress they had made, and discuss problems and other issues that had risen during the week.

A general overview of the planning, as it was expected to be done by the students is described at Figure 2, taken from the web-page of the course.

---

[4] `http://www.csis.gvsu.edu/class/brio/BrioProject/`

[5] *ICT* is an abbreviation for Information and Communication Technology

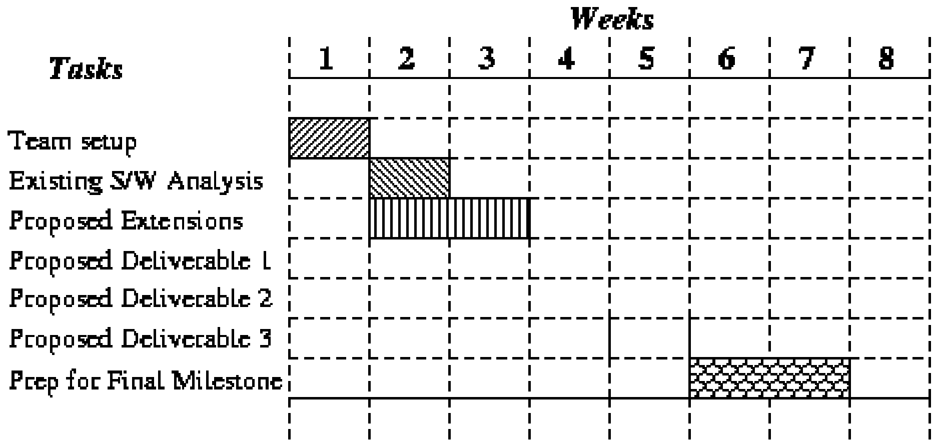

Figure 2 A time plan for the students' work in the project

The gradings were based both on the process the students went through, mainly evaluated through the weekly meetings and the outcome of the work. There were both individual and group-based components in the grading. The grading systems and the related issues are further discussed in Pears, Daniels, Berglund and Erickson (2001).

2.4 A technical description of the student project

On the web-site of the student project[6], a technical description of the setting and the requirements for the results produced by the students is available. As was pointed out earlier in this section, the task that is given to the students is to produce a software system that makes it possible for an end-user to control the labyrinth using any web-browser on Internet.

In Figure 3 the main components of the project are found. The system as a whole consists of some inter-connected sub-systems that might run on the same computer or on separate computers. The hardware, operating systems, standard communication solutions etc are supplied by the two universities, while the task of the students is to write the required software.

The end-user should have the possibility to draw a path that he or she wants the ball to follow. He or she should then be able to follow the movements on the screen that the ball makes on the physical board. The client[7], written in Java, offers this visual interface to the end-user.

---

[6] `http://www.csis.gvsu.edu/class/brio/BrioProject/`

[7] The concepts of a *clients* and *servers* are fundamental within the field of computer communication. A client, as an active participant, is a computer that initialises a dialogue by sending a request for data or for another service. The request is sent to a passive server that answers requests to provide services and normally sends data or offers a service as a reply to the request. The two concepts are normally referred to as a pair.

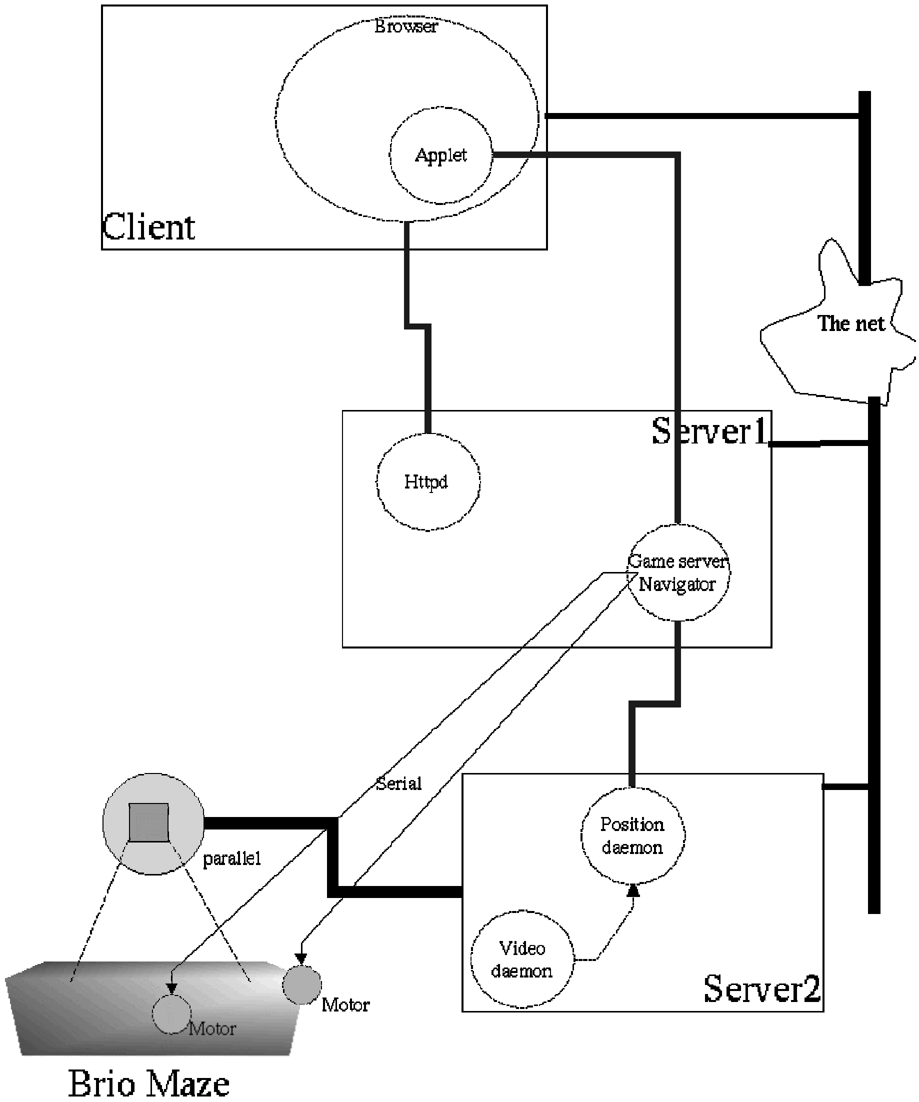

Figure 3 The architecture of the Brio system

The movements of the board are controlled by step-motors[8] that have a serial connection[9] to the game server (marked as server 1 in Figure 3), which can be seen as the centre of the system.

The system needs to keep track of the movements of the ball in order to be able to control the motors in a relevant way. This requires feedback, which is fed into the system through a camera that constantly supervises the board, as seen in Figure 1. The camera is connected to the video server (server 2 in Figure 3) through a parallel connection.

---

[8] A *stepmotor* is a type of electric motor, which is fed with electric pulses. For each pulse, the motor turns a certain angle. This feature makes stepmotors useful in computer-controlled devices.

[9] Data that is transmitted in *serial* mode, in contrast to parallel mode, is transmitted one bit (or unit) at the time. All bits use the same (physical) connection. Data that is transferred in *parallel* mode is sent in parallel simultaneously on several lines, one bit (or unit) on each line.

The purpose of the video server is to interpret the images from the camera and transfer the information the camera offers into information about the ball: its position, speed and direction of movement etc.

The game server acts as the coordinator of the system, getting information from the camera through the video server and information on user's demands from the applet[10]. From this information the game server should calculate how the motors should move, and send the required information to the motors for these movements to take place. This server should also provide the information about the movements of the ball and the status of the system to the applet.

The hardware is available at each university, and except for the cameras, where different brands with different characteristics are used, it is basically the same at both places. The Brio boards and the physical equipment are getting old, and the variations between the different boards are considerable. These variations between the boards add a difficulty to the task: The programme system ought to be written in a way that makes it work correctly on most of the boards.

As should be clear from this description, there are several places within the system where communication between two computers or virtual machines[11] or a hardware controller and a computer takes place. The focus of this study is the students' experiences and understanding of the network protocols that can be used in this project, and as will be shown in this report, there are a number of possible ways of understanding them.

2.5 The network protocols taught

As noted previously different aspects of computer and data communication are basic components of the Runestone course curriculum. Many different network protocols have evolved over the years with different properties and for different applications. However, a consensus has emerged on a few of all possible protocols that could be used being relevant for education and their roles in education (Derrick & Fincher, 2000). This section briefly describes three standard communication protocols that are frequently taught, and that can be used by the students in their project work to different degrees. For deeper understanding of the different protocols, as well as the context to which they belong, refer to standard literature in the field, such as Stalling (1997) and Tanenbaum (1996).

To communicate between computers a set of rules is needed, specifying what should be sent and how it should be interpreted. Stallings (1997) defines network protocol in the following way:

> Set of rules that govern the operation of functional units to achieve communication (p. 780)

---

[10] *Applets* are Java programs that are intended to be run in a web browser, such as Netscape, or by a dedicated appletviewer. Applets are frequently used to implement graphical interfaces.

[11] A *virtual machine* can be described as a simulated computer that runs on another computer. In other words, a virtual machine is a program, that, when executed, behaves as a computer with well-defined properties. Virtual machines are one of the underlying techniques for platform-independent programs. Java that can be run on different kinds of computers and in different environments has a virtual machine. Java's virtual machine is (at least in theory) the only program that has to be rewritten to run Java in a new environment.

The protocols contain information about message formats, formats for control information, responses to messages, timing requirements as well as information about how errors or other unexpected events should be handled.

Network protocols are standardised. Software that offers the programmer the routines that are needed to handle the communication has been developed for the important protocols. As an example can be mentioned that TCP software (Transmission Control Protocol) offers, among other routines, procedure that listen for incoming messages, that sets up connections, sends messages and closes down connections.

This report mainly concerns the students' understanding of some standard communication protocols: TCP, UDP and RMI[12]. To different degrees it is possible to use these protocols in their project and they are normally described as important protocols for computer communication.

TCP, UDP and RMI are all *end-to-end protocols*. This means that they offer services that a program developer or programmer can use when developing application programs for end-users. A programmer who is a user of these end-to-end protocols does not normally need to think about the underlying layers. For example, he or she can completely disregard the physical appearance, as voltages used or frequencies used by the communicating computers to transmit a message. In other words, one could say that a protocol "is" a set of rules, but the concept of network protocols also captures some elements of data semantics.

TCP, *Transmission Control Protocol*, is undoubtedly the most widely used of the three protocols, since it forms the basis for data transfer over Internet and other networks based on the same technology[13]. TCP offers connection-based services.[14] This means that a programmer who uses TCP needs to make his or her program establish a connection in order to communicate with another computer, similar to when we need to dial a number and wait for an answer to be able to communicate over the telephone. It is a reliable protocol, in the sense that the sending program gets a confirmation or acknowledgement from the receiver that the information has arrived.

UDP, *User Datagram Protocol*, is a connectionless protocol. This means that a sending computer does not get any confirmation if the data sent has reached its destination or not. This makes it unreliable, or unsafe. Still, it is useful in applications where some losses can be accepted, but where speed is important, such as video conferencing.

---

[12] *TCP*, *UDP* and *RMI* are abbreviations. Within the field of computer science the abbreviations are usually used instead of the full names.

[13] There is a common convention of writing *Internet*, with a capital *I*, when referring to the global Internet, and of writing *internet* when discussing independent internets. An independent *internet* or *internetwork* is a network that in its turn consists of a set of networks that are connected in a way so that they interconnect to form a whole network. The parts that make up an internet can be Local Area Networks (for example within a building) or other internets.

[14] The word *connection* can have different meanings. As I use the word in this report, a connection has to be created or "set up" between two of the many computers on a network. Setting up a connection resembles in many ways making a telephone call.

RMI, *Remote Method Invocation*, is closely related to the object-orientation[15] in Java, since its purpose is to allow Java objects, distributed over a network, to communicate. To a programmer, RMI provides access to routines (called methods with the terminology used) on remote machines as if they were available on the local computer.

Data that is transferred using UDP or TCP does not have any kind of meaning assigned to it. These protocols are restricted to the transmission of data, and leave issues concerning its interpretation to be implemented by the application programs; that is, to the programs that use TCP or UDP. RMI, on the other hand, is intended for transfer and manipulation of objects, entities that have intended meanings or interpretations. This higher level of abstraction clearly makes RMI a more complex protocol. Consequently security policies and related issues become more demanding for the programmers.

3. The study

I have chosen to use primarily a phenomenographic approach to address my questions about students' learning. The results of a phenomenographic research project are, as is argued by Marton and Booth (1997), insights into qualitatively different ways in which a phenomenon is understood.

A phenomenographic research project thus aims at analysing and describing the variation in ways in which central concepts of the subject matter are understood or experienced by the learners. In my study, university students' experience of computer networks is in focus, and phenomenography offers the possibility for me, as a researcher, to investigate the students' own experience of network protocols.

One of the keystones of phenomenography is that phenomenographic researchers can arrive at a limited number of qualitatively distinct categories of description which succinctly and adequately cover the countless ways in which a phenomenon can be experienced, or understood. The results are articulated in a set of qualitatively distinct categories of description that express the variation in how a phenomenon is experienced by the learners, called the outcome space. The outcome space that I arrive at thus gives me, as a computer scientist and phenomenographer, the possibility to relate the students' understanding of computer network to the goals of the education, as it is expressed in the course descriptions and as I, as computer scientist, understand the protocols. The strong focus on the object of the students' learning is an important feature of phenomenography as a research approach in the research that I present in this thesis, maintaining as it does the subject matter that is of prime interest to me as a teacher.

In the next sub-section, I will discuss aspects of the students and the interviews, while the larger part of the section will focus on the aspects of the use of phenomenography in this research project. Finally, I will discuss some methodological decisions that I have taken.

---

[15] *Object-orientation* is based on the idea that a program consists of a set of communicating entities. The execution of the program takes place within these entities, and in the interaction between them. Java and C++ are programming languages that support this style of design and programming. There is a vast literature on object-oriented programming and object-oriented programming languages. Budd (1999) discusses the ideas behind object-orientation and Java.

3.1 The interviews

Ten students have been selected as candidates for interviews in the US and nine in Sweden on two occasions during the spring of 2001. The students were selected to obtain a variation in backgrounds, earlier study results, gender, age, motivation to take this course as indicated in a background questionnaire etc. The students participated on a voluntary basis in the study. They did not get any credit for participating, but got two movie tickets each as a sign of recognition. With a few exceptions, the students attended both interviews. Those who did not attend all had different reasons for this, including illness, exchange studies abroad, and shortage of time.

The interviews were carried out by the author of this report, in Swedish with the students in Uppsala and with a Swedish exchange student at Grand Valley State University and in English at Grand Valley State University and with an exchange student from a European country studying in Uppsala. The first interview was made a few weeks after the course had started, while the second was carried out after the end of the course. The interviews have been transcribed by native Swedish and English speakers respectively. Excerpts of interviews in Swedish are translated into English by the author of this report.

The results presented in this report are based on the first interviews in both countries as well as the second interview in Sweden. Nine of the interviews from the US have been used and five from the first set of interviews in Sweden. From the second set of interviews in Sweden four interviews are used. Some of the interviews from Sweden have been impossible to use, due to poor quality of the recording, while other interviews have been judged as less interesting for the research project. The second set of interviews in the US is currently being transcribed and will, together with all relevant Swedish interviews, be available for future research.

Excerpts from interviews are presented in this report to illustrate different aspects of the categories created and to make the results open to inspection. When referred to in the text, the students are assigned names by me that are not related to their real names. All students who are currently studying in Sweden have been given names that start with an "S", while names that start with an "A" indicate that the student is currently studying in the US. Alec is thus an American student, while Samuel and Sven are Swedish students. In the excerpts of the interviews the statements of the students are preceded by their name and the number 1 or 2, indicating if the statement is from the first or the second interview. Staffan2 is thus intended to indicate a statement made by Staffan during the second interview.

Since there are considerably fewer females than males taking the course, I have chosen not to indicate if any particular quote is from an interview with a male or a female. Four of the students interviewed were female. Instead I refer to all students by "he". The purpose of this is to respect the anonymity of the students. The females could, since they are few, easily be recognised by schoolmates or teachers. The research design and the way of presenting the data that I have chosen make it impossible to address gender issues in the current report. Since the data collected contains statements from individual students, it could be possible to address gender issues in my later research in this project.

3.2 The use of phenomenography as a research approach in this study

The process of a phenomenographic research project is not algorithmic, and does not in any way follow a specific pre-defined path. The result does not describe causal relationship, but instead focuses on more complex understandings and relations.

Phenomenography as such offers a framework of theoretical and practical tools, but leaves to the researcher to design his or her research. In the coming sub-sections I want to draw attention to some specific aspects that have become important during this phenomenographic research project. Marton and Booth (1997) give deeper insights into phenomenography in general and discuss aspects that are not touched on below.

3.2.1 Collecting data

As stated by Adawi, Berglund, Booth and Ingerman (2002) a goal of the data collection is to maximise the variation in the pool of meaning. In the current study this means that I, as a researcher, am eager to collect material that can support a creation of a rich and expressive picture of the different understandings of network protocols. Two obvious ways of getting a large variation is by selecting students in a way that you can hope that they will express different understandings, and by interviewing the students in a way that encourages richness in their answers.

When making a selection of students, as a researcher I have tried to construct a sample, where students with different interests, backgrounds, earlier results, attitudes to their studies etc are represented. Data for this selection were taken from different sources. Important information came from a questionnaire before the start of the course, where the students were asked about their expectation. Previous study results in computer science, as well as study results from other subject areas that were available in the records of the two universities, were also used.

As a researcher, you start an interview with a set of open questions, based on your ideas of what you want to learn from the interviewee. To the interview you bring your understanding of the topic for the interview, in this case computer networks, as well as your understanding as a phenomenographer. During the interview, which normally is semi-structured, the researcher interacts with the interviewee in order to explore his or her understandings of the phenomenon in focus. The researcher can ask follow-up questions, nod or in other ways encourage the interviewee to reveal his or her experience of the phenomenon that is in focus during the dialogue.

The outline for the semi-structured interviews can be found in appendices of this report. It is worth pointing out, that the ourlines only served as a framework and as a guideline for what issues that should be discussed. There were many improvisations during the interviews, in order to follow up statements by the students and in other ways try to increase the variation in the ways of speaking about the phenomenon as much as possible. As can be seen from the scripts, several issues were discussed that are not further elaborated in this report. These topics are open for future analysis.

The openness to the experiences of the interviewee, which is an important feature of a phenomenographic research project, creates possibilities for the researcher to draw new conclusions about the interviewees' ways of experiencing a phenomenon, but does not in any way imply that the researcher can get a full picture of the understandings of an individual. Instead as a researcher you only get limited insights into the interviewee's experience of the phenomenon. To draw this conclusion still further: The interviews are learning occasions not only for the researcher, but also for the student. This is a dynamic process, where the interview influences the interviewee's as well as the interviewer's experience of the phenomenon.

3.2.2 Analysing data

The goal of the analysis, as well as the phenomenographic research project as a whole, is to reveal the experiences of the phenomena (here network protocols) in focus of the study at hand. The non-algorithmic structure of a phenomenographic research puts the interaction between the researcher and the data in the centre of such a process. Statements made by the students are decontextualised, that is, taken out of the context[16] or meaningful background in which they were originally uttered, to be recontextualised, that is, put into another context. The categories of description that are created in this way are the researcher's recontextualisation of the data. This recontextualisation is made in an iterative process, where the researcher starts with a tentative understanding, and then through reconsiderations and refinement reaches a description that he or she finds relevant to address the research question and honest to the data. In a sense, the process can be seen as a discussion between the researcher, the pool of meaning and the developing categories of description. In other words, when I, as a researcher, create categories of description my goal is to describe the different ways of experiencing network protocols that I have met in the group of students.

The results, as categories of description of certain phenomena, are to be interpreted at the collective level. A tool for the researcher, when studying the collective level, is to look for logical consistency for the categories created. As pointed out by Marton and Booth, such a logical consistency often takes a hierarchical form: An advanced way of experiencing something can be "more complex, more inclusive (or more specific)" (p. 107) than another, less advanced way of experiencing the same thing. The framework in which the network protocols are experienced will be described and studied as structures in this report. It is worth noting that the categories and the hierarchy are interpretations made by the researcher, and are expressions of the researcher's view on data, combined with his/her understanding of the subject matter and of phenomenography as a research approach. That is, the analysis is an aspect of the researcher's experience of the students' experiences. A further discussion on issues concerning the perspective of the researcher can be found in Adawi, Berglund, Booth and Ingerman (2002).

I also want to draw the attention to an aspect concerning the translation and interpretation of the interviews. In the excerpt below Sven has misunderstood, or does not remember, the correct meaning of the abbreviation RMI.

Sven2:	Remote, and then there is Indication, but what is the other, in the middle then. I am not so ....
Interviewer:	Method
Sven2:	Remote Method Indication
Interviewer:	And what's that?

[16] The word *context* has many uses in educational research. For a discussion of its use in phenomenography, and especially aspects on who is experiencing the context, see Adawi, Berglund, Booth and Ingerman (2002). Throughout this section of this report the word context can be read as "meaningful background"

This can have several explanations, and as a researcher one should be careful not to draw any firm conclusions. Apart from indicating unfamiliarity with the concept, it can be a matter of the language: Since his mother tongue is Swedish, it is not obvious that the word "invocation" is a part of his normal English vocabulary. A possible Swedish translation is "anrop", a word that does not resemble it. The other two words offer less difficulty: The Swedish equivalence of "method" is "metod"; here there are clearly similarities. The word "remote" is frequently used within the field of computer science and is most probably part of the English vocabulary of an advanced undergraduate student in computer science.

Another language related issue, that has appeared as a small problem is the distinction between a generic *internet*, as a set interconnected network and the global *Internet* (see footnote on page 13 for a discussion about the difference between the two words). Neither in Swedish nor in English is it possible to hear a difference between the two in the statements made by the students. The context in which the word is used only occasionally offers help for an interpretation. As a consequence, the distinction between *Internet* and *internet* has to a certain degree to be guessed. I have preferred to use the word *internet* in cases when I have hesitated about the interpretation, since *Internet* is one specific *internet.* This means that the word *internet* covers both generic internets and the global Internet.

3.3 Some methodological decisions

In this section I will discuss some of the methodological decisions I have taken during this project and their implications for the results I present, based on the insights I have gained during the work, my considerations of the research questions, and discussions with colleagues within computer science.

In the analysis I decided not to discuss those few excerpts from interviews where students gave the plain answer "I do not know" as an answer. They are excluded, since they do not express a specific way of experiencing the protocols.

Differences between expressions *within* the categories have also not been explored. Differences within categories could relate to how well students articulate their understandings or how sure they are of the relevance of what they say, as the following example (that will be further analysed in section 4.1.3) illustrates. One could speculate about possible differences in the understandings expressed by Albert and Allan in the following interview extracts:

> Interviewer: Um, what is TCP?
> Albert1: TCP, um, it's um, part of the internet protocol. It's used with part of the internet protocol typically. Um, it's one of the methods of communications, I don't know a whole lot about it, as far as the whole, um, design construction behind it.

> Interviewer: Um, you've talked about TCP. What is TCP?
> Allan1: Basic concepts.. it's a protocol language, I guess you can call it, that you just put your data in and it's sent across the network using the different protocols you want to use, like IP or.. I can't think of any other protocols off the top my head. But it is more or less a packet that you put your data in and you send across and it has some features such as, keeps things in order when you, um, when you get to the, um, when it gets to the server you want to go to.

In section 4.1.3 I develop my arguments for interpreting these excerpts as expressing an understanding of TCP that I describe as "a connection over a network".

The categories of description that are created describe *qualitatively* different ways of experiencing a phenomenon, and are the smallest unit of the analysis. I use the term *category of description* as it is presented in Marton and Booth (1997). When discussing the phenomenographic research they state that:

> [...] the individual categories should each stand in clear relation to the phenomenon of the investigation so that each category tells us *something distinct* about a particular way of experiencing the phenomenon. (p. 125, my italics)

This use corresponds well to the definition of the word *quality* that is found in Webster (1979) "that which belongs to something and makes or helps to make it what it is; characteristic element; [...]". It would be possible to argue that one of the students above expresses this quality "better" than the other, with regard to some criteria. However, in this phenomenographic research project I have decided not to study this kind of difference; it is the quality of the *categories* that is in focus, and not the quality of the ways in which individuals express themselves.

Nor have I made any quantitative analysis of the answers the students gave. A phenomenographic research project is designed to be understood across a collection of people, a population of interest, and the results do not describe individuals. A statistical analysis would demand that individuals or individual expressions be counted. As already described, the selection of students and the interviews aim at obtaining rich data for a phenomenographic study, and a different approach would need to be taken to get a good sample for a statistical study.

4. Students' understanding of individual network protocols

The students were asked during the interviews to describe what they meant by TCP, UDP and RMI. When opening the subject of discussion, the three protocols were treated as three different topics by the interviewer. Later in the conversation about specific protocols comparisons were frequently made, often on the initiative of the students.

The opening question was normally "What is TCP?" followed by similar questions for the other protocols. There was no particular order in which the protocols were introduced by the interviewer. On the contrary, often the decisions about the order of the protocols were taken as a consequence of the flow of the conversation, for example by the interviewer referring to earlier statements made by the student.

This section focuses on the students' understanding of TCP, UDP and RMI. Aspects that will be discussed are students' understandings of the meaning or use of the protocols, their technical characters, and the framework to which they belong.

4.1 Students' understanding of TCP

Three qualitatively different ways of experiencing TCP have been identified within the group of students. Table 1 gives an overview of the categories of description. Differences between the categories are found in the framework of which the protocol is experienced as a part, as what the protocol is experienced and in what way it is described.

Table 1. Categories of description for TCP.

	As what is TCP understood?
1.	TCP is understood as safe communication between two specific computers and is described in a concrete way
2.	TCP is understood as a connection over an internet and is described in an abstract way
3.	TCP is understood as a standard communication tool in a framework that includes and goes beyond computer networks, and is described at a meta-level

A fundamental difference between the three ways of experiencing TCP is the framework of which the network protocol forms a part. Similar experiences of frameworks have been identified for the students' understanding of UDP and RMI, as well as for their understanding of the general concept of a network protocol. In the coming sections I will explore the understandings of TCP, that I have identified.

4.1.1 An overview of different ways of experiencing TCP

The first category describes an understanding where TCP is related to an experienced framework that consists of two specific computers, where data is transferred between these two computers. The network only exists as a background to this transfer. In the second category the framework is broader: TCP is experienced as integrated with and a part of an internet. Finally, in the third category, the framework has its limits outside an internet and the world of computers, and also takes human decisions into account. Here TCP is the result of decisions taken by a committee.

TCP is, in all the three categories, experienced as an inseparable part of the framework to which it belongs. The protocol is integrated with specific computers, the network, or the world outside the network. A protocol needs its surroundings for its existence, and could thus not exist without the world of which it is a part. Neither would its surroundings be the same without the protocol.

The experiences of what TCP "is" or "means" differs between the three categories. The "meaning" of the protocol is closely related to the experienced framework of which the protocol is a part. When TCP is related to two specific computers, it is understood as a safe[17] way of communicating, that is, a user can know that no data is lost during the transfer. In the second category TCP is understood as a connection over the network. A connection has to be created or "set up". Once the connection is there, it offers safe communication. In category three TCP is experienced as a part of a framework that reaches outside the world of computers. TCP is a standard tool for communication; as a standard it is decided by a committee. The fact that it is a standard is what makes it useful.

When talking about TCP, as well as the other network protocols, the students frequently referred to the technical characterisation, or technical properties, of the protocol, telling the interviewer "how the protocol works". No variation in the understanding of this technical characterisation for TCP has been found in data. TCP is experienced as a protocol with

[17] In the analysis presented in this report, I use the word *safe* as synonymous to the word reliable.

acknowledgement in the three categories. Rather, the technical characterisation is thus what gives a specific protocol its character that makes it possible to recognise TCP as TCP or UDP as UDP etc.

An analysis of the different aspects of the categories of description for TCP is summarised in Table 2. The first column indicates what TCP is experienced as. The second column focuses on the world or framework in which TCP is experienced. The next column shows the technical characteristic that is understood for TCP. Finally, the last column indicates in what way TCP is described.

Table 2. Aspects of the different categories of description of TCP

<table>
<tr><th></th><th>As what is TCP experienced?</th><th>As a part of which framework is TCP experienced?</th><th>What is the technical character of TCP?</th><th>How is TCP described?</th></tr>
<tr><td>1.</td><td>Safe communication</td><td>A framework of two specific computers</td><td rowspan="3">TCP is a protocol with acknowledgement</td><td>In concrete terms</td></tr>
<tr><td>2.</td><td>A connection</td><td>A framework of an internet</td><td>In an abstract way</td></tr>
<tr><td>3.</td><td>A standard for communication</td><td>A framework of a world outside the network</td><td>On a meta-level</td></tr>
</table>

A relation between the experienced frameworks expressed in the three categories can be identified: The framework is wider in category 2 (an internet) compared to 1 (two computers), and still wider in category 3 (a world outside the network). Thus, there is a hierarchical structure that relates to the framework in which TCP is experienced as a part. A similar structure can be found when looking at how TCP is described: The level of abstraction increases from category 1 over category 2 to category 3. In the higher categories, the TCP is experienced as a part of a wider framework and in a more abstract way. In a way, the categories can be seen as inclusive: 1 is included in 2, and 2 in 3.

It is often the case that the students shift between two (in rare cases three) ways of experiencing TCP, as will be discussed later in section 6.1. However, many of the students do not shift between different ways of experiencing the phenomena.

In the following sections I will show some data and present the arguments that made me draw these conclusions. Since the material from the interviews is rich, only a small selection of excerpts can be presented in this report[18].

4.1.2 TCP as a safe communication between two computers

This category of description describes an understanding of TCP where the protocol is experienced in a framework of two specific computers that communicate. The protocol is explained in concrete terms and is experienced as safe communication.

Andy's statements during the first interview clearly show the focus on two computers:

[18] The author can on request supply the complete interviews subject to guarantee that the integrity of the interviewees will not be abused by any form of unauthorised publication

Interviewer:	What is TCP?
Andy2:	That is ... you communicate with .. between client and server with TCP packets.

Here, Andy describes TCP as communication between a server and a client. The explanation given refers to two specific computers: a client and a server. The concept of client-server implies that the issue of communication is integrated with the computers, since a server and a client could not be imagined without communication between the two in the field of computer science. The communication is the basis for the existence of a server and a client.

In the continuation of the dialogue, the issue of safe communication is raised by Andy:

Interviewer:	What is a TCP packet?
Andy2:	That's a type of packet, that one sends, that contains also ... so that one can get.... one must. It is a safe communication so that one knows ... three-way, so that one always knows it arrived or not, in contrast to UDP.

Andy here points out that TCP is a safe communication and says that TCP informs whether data, in the form of a TCP packages, has arrived or not. Also, by mentioning "Three-way" he indicates that there is an acknowledgement sent by the receiving computer. [19]

Staffan also talks about his understanding of TCP as a safe communication between two specific computers during the second interview:

Interviewer:	What is TCP IP then?
Staffan2:	It is sort of ... a safe connection we send a stream of data back and forth and it's checked that there is no errors and suchlike [....] we've read about this

.

Staffan answers the question concerning TCP by saying that there is a stream of data "back and forth" in order to check that there will be no errors. The expression "back and forth" indicates that there are two well-defined end-points, between which data is sent. He also expresses the view that it is a safe protocol by saying that TCP checks that the stream of data "is no errors". He uses the expression "safe connection" to explain that TCP offers safe communication. By the expression "safe connection", I understand that he points out that the communication is safe. As was mentioned in section 2.5, my use of the word "connection" is somewhat different in this report. The words "back and forth", as well as his discussion of errors, clearly indicate to me that he uses the word in a way that differs from the usage I have chosen in this report.

Sebastian explains safely during his second interview his understanding of TCP and particular the use of acknowledgements to make sure that information arrives. In the following excerpt he starts by telling in what parts of the project his group has used TCP:

Sebastian2:	No, down from the server and down to the hardware, the bits where we use TCP/IP.
Interviewer:	What is that?
Sebastian2:	It is...it is a communication protocol which uses...ack?
Interviewer:	Acknowledgement?

[19] *Three-way* indicates in fact that there is an acknowledgement sent to confirm the arrival of the first acknowledgement. This technique is used when setting up a TCP connection between two computers to make it possible for the computers to agree on different parameters for the communication.

Sebastian2: Yes, an acknowledgement, That is, that I know that the information I send has arrived correctly, and what comes back has also arrived. There is a bunch of other stuff that I have to look out for.

He says that an acknowledgement is sent by the TCP[20] protocol, and that there is a control performed by the protocol if the confirmation arrives or not. By the words "I know that the information I send has has arrived correctly", he indicates the reason for the acknowledgement: To get a safe communication between the two specific, communicating computers that he is focusing on.

Since the interview with Sebastian is made in Swedish and later was translated into English, a word about the hesitation in the third statement of the excerpt of the interview might be required. When explaining what a communication protocol is Sebastian first has difficulties finding a Swedish word (bekräftelse), so he turns into English and starts saying "ack...", for acknowledgement in a hesitant voice. As an interviewer, I then present the Swedish word to him, which he uses in the next statement. It is worth noting that Sebastian studied computer networks and TCP as an exchange student in another language before taking this course. His hesitation can therefore be interpreted as a question of language and terminology and probably not as related to the concept as such.

As we have seen in this section, this category of description describes an understanding of the TCP, where the protocol is used for transferring data between two specific computers. TCP uses acknowledgements to verify that the information arrives safely at the destination. Clearly, all descriptions relate to concrete entities, like specific computers, packages of data or confirmations.

4.1.3 TCP as a connection over a network

This category expresses an understanding where TCP is related to and integrated with an internet as a whole. TCP offers a possibility to create connections over a network. The understanding of the protocol is expressed in abstract terms.

When Albert was asked what TCP is during the first interview, he talks about TCP as a part of an internet:

Interviewer: Um, what is TCP?
Albert1: TCP, um, it's um, part of the internet protocol. It's used with part of the internet protocol typically. Um, it's one of the methods of communications, I don't know a whole lot about it, as far as the whole, um, design construction behind it.

Albert talks about TCP as an internet protocol and mentions that it is a part of an internet.

Axel also expresses a way of experiencing TCP as a part of the Internet, that is as a part of an internet, as can be seen the following excerpt of the first interview:

Interviewer: [...] Um, I want you to talk about TCP.
Axel1: TCP/IP?
Interviewer: Ya.

[20] IP, Internet Protocol, is an underlying protocol, providing the basic services used to implement TCP. TCP, together with the underlying internet protocols are often referred to as *TCP/IP* or the TCP/IP stack

Axel1:	TCP/IP is how almost everything on the Internet communicates. IP addresses and everything, and that's um, one of the fundamentals behind RMI also. One could give it the address where the object is [...] the IP address [...]

Beginning by saying that it "is how almost everything on the Internet communicates", he indicates that he regards the protocol as very important and as a part of Internet as a whole. The importance of the protocol is emphasised by his reference to IP-addresses[21] and to RMI.

Another student, Allan, also stresses that TCP is a part of an internet:

Interviewer:	Um, you've talked about TCP. What is TCP?
Allan1:	Basic concepts.. it's a protocol language, I guess you can call it, that you just put your data in and it's sent across the network using the different protocols you want to use, like IP or.. I can't think of any other protocols off my head. But it is more or less a packet that you put your data in and you send across and it has some features such as, keeps things in order when you, um, when you get to the, um, when it gets to the server you want to go to.

He says that TCP is a protocol language[22] that is used for sending data across a network. In this way, he clearly indicates his view that TCP is an integrated part of the network. He then explains its main feature, as he sees it: The order of data is kept when sent to the application program through the TCP socket[23], although data physically might have arrived to the server in any order. This makes the protocol safe.

In the same interview with Axel that was mentioned above, he talks about TCP as a connection:

Interviewer:	But what are the specifics about the TCP protocol, some characteristics of it?
Axel1:	Ah, some characteristics of it. Well, I don't know a lot of the underlying characteristics of it [...] numbers with dots [...]
Interviewer:	You can't tell me, you can't say anything about the differences between UDP and ...
Axel1:	I don't really.. UDP and TCP are different in that TCP is a connection protocol and UDP is connectionless. Um, I've never quite completely understood exactly how one's connected and what is not. So, that is the most I can really say.

Axel tells the interviewer that he does not know any technical details, and continues by pointing out the important difference between TCP and UDP: TCP is a connection-based protocol, in contrast to UDP.

The entities Axel mentions are described in an abstract way: He refers to other protocols in a comprehensive line of reasoning, instead of talking about packages or flows of data.

[21] An *IP address* is a unique 32-bit number that is assigned to computers on an internet. This address is used for all communication with the host. IP addresses are written as four decimal numbers with dots between. As an example 130.238.8.89 is address of the computer used by the author of this report.

[22] The term *protocol language* refers to a formal language, not to a natural ("spoken" "human") language, within the field of computer science. A formal language is used to express statements about calculations in a general sense as for example giving instructions to a computer.

[23] A TCP *socket* is an endpoint of a connection between two computers originally created in a Unix environment. It is used by a programmer as a mechanism for transferring data.

In this section, a way of experiencing TCP has been described where the protocol is seen as a connection over an internet, and at the same time, as an integrated part of the network. The protocol is build on a technology with acknowledgements, and is experienced as a connection.

4.1.4 TCP as a standard communication tool

This category of description expresses an understanding where TCP is seen as a part of an experienced framework that goes outside networks to humans decisions about the protocol.

In the following part of the first interview, Adam expresses such an understanding:

> Interviewer: So, what is TCP then?
> Adam1: Well that I have studied in some networking classes um, Transfer Control Protocol, something along those lines. Um, that is just a protocol for computers to communicate with each other. That's a standard that was created by a committee somewhere, sometime, and it's just a, it's a protocol, meaning that it's, it specifies um, the layout and the size and what's in the header and footer of packets being sent across networks and things like that. So it's, it's a standard communication tool

He argues that TCP is a standard that is created by a committee. The form of the packages sent is the result of conscious decisions, taken by the committee.

Later, when the choice of TCP instead of RMI as the principal protocol for their project is discussed, he continues:

> Interviewer: Yes, but can you tell why you have chosen TCP?
> Adam1: Right, it's for one thing it doesn't require this registry running in the background. It's sort of a universal standard so that, you know, our applet can be run on any computer anywhere and still communicate with the game server running on Linux or whatever. Um, so I guess just being a standard and being more flexible than RMI.

TCP has two advantages over RMI, according to Adam. One is technical: TCP is simpler since it does not require a complicated background program to be run. The other advantage is that TCP as a well-defined standard increases the flexibility.

Adam compares the use of RMI and TCP on several occasions during the whole interview. From his remarks above and comments in general, it is clear that he takes for granted that TCP offers safe communication. It is never spoken out aloud, rather it can be seen as a condition for the rest of the conclusions.

Adam reasons about TCP without making direct references to the technical structure or the entities in the communication process. Rather he talks about standards, flexibility, and tells the interviewer that size and design of packages are decided, without mentioning whatthe packages look like. In this way, he talks about properties of the protocol in an indirect way, from a position outside the two protocols, telling the interviewer how decisions about the protocols are taken and what the consequences of the decissions are. This can be seen as reasoning at a meta-level[24] about the protocol.

---

[24] I use the term *meta* in words like meta-level to indicate a reasoning that goes "beyond, higher, transcending" (Webster, 1979). A meta-level reasoning about a protocol goes beyond discussions of the properties of the

In this category of description, TCP is seen as related not only to computers, but to human decisions as well. The discussion is mainly focused on how decisions are taken and the consequences of the design, and is thus held at a meta-level.

4.2 Students' understanding of UDP

Three qualitatively different ways of experiencing UDP have been identified. The categories of description that have been created for UDP, and that are presented in Table 3, share its structure and main properties with the findings for TCP. The categories of description as they are presented in Table 1 for TCP are thus basically the same as for UDP. The important difference that has been identified is that UDP is recognised as an unsafe or connectionless protocol without an acknowledgement.

Table 3. Categories of description for UDP

	As what is UDP understood?
1.	UDP is understood as an unsafe communication between two specific computers and is described in a concrete way
2.	UDP is understood as connection-less communication over an internet and is described in an abstract way
3.	UDP is understood as a standard communication tool in a framework that includes and goes beyond the network, and is described at a meta-level

During the interviews, a large number of students spontaneously compared the two protocols and pointed out differences between them.

Since the similarities between the two protocols are strong, I will only briefly sketch the experience of UDP that has been discerned within the group, without presenting the full analysis that I have made, nor the data.

Table 4 gives a more detailed picture of the experience of UDP. The first column indicate as what the UDP is experienced, in analogy with the corresponding table for TCP, Table 2. In category 1, UDP is experienced as an unsafe protocol. For communication that uses unsafe protocols, the sender does not get any confirmation from the receiver if information has arrived. With this protocol the sender does not know if data is lost and if it therefore needs to resend data.

The understanding that is expressed in category 2 is of UDP as a protocol that does not set up a connection. A connection demands that the sending computer gets a confirmation that data arrives to the receiving computer. Without the possibility to obtain a confirmation, data can be sent without ever be received. In the concept of a connection lies an interaction that is not fulfilled in protocols that does not have an acknowledgement

As is indicated in the third column, UDP is experienced as a protocol without acknowledgement within the three categories. This is the technical aspect that characterises

protocol, and takes the protocol as a whole as an object of discussion, that is open to variation in other dimensions.

UDP, and makes UDP recognisable as UDP. Comparing with TCP, it is notable that TCP is recognised as a protocol with acknowledgement. This way of characterising the two protocols, as having or not having an acknowledgement, is thus aspect that takes different values for the two protocols.

Table 4. Aspects of the different categories of description of UDP

	As what is UDP experienced?	**As a part of which framework is UDP experienced?**	**What is the technical character of UDP?**	**How is UDP described?**
1.	Unsafe communication	A framework of two specific computers	UDP is a protocol without acknowledgement	In concrete terms
2.	Connectionless communication	A framework of an internet		In an abstract way
3.	A standard for communication	A framework of a world outside computer networks		On a meta-level

4.3 Students' understanding of RMI

Different ways of experiencing RMI have been discerned in the group of students. In many important ways the identified understandings resemble the structure that was described for TCP and UDP. However, the picture of the students' experience of RMI is somewhat more complex than the pictures given in the previous sections. A possible reason for the complexity found in the data is the purpose, design and function of RMI. RMI gives a programmer a possibility to create a program which, in its turn, can start other programs on other computers or machines[25]. That is, RMI offers not only a possibility to transfer data between different computers (as do UDP and TCP), but also to transfer and execute code on other computers. This means that RMI from the programmer's perspective offers more possibilities than the other two protocols, but becomes at the same time more complex and thus more complicated for him or her to handle.

4.3.1 An overview of different ways of experiencing RMI

Three different categories of description that together express the students' experience of RMI have been identified in the data. A critical difference between the categories of description has been found in the frameworks in which the students experience the protocol: The protocol is experienced as a part of an environment that consists of two computers, as a part of an internet, or as belonging to a world that goes beyond computers. This difference is closely related to how, or as what, the protocol is experienced.

To describe the increased complexity, I have created three subcategories of the first category that relates to two specific computers. The subcategories differ in the roles the two computers play in the communication: Undefined roles in the first subcategory, different but not clearly specified roles in the second, and finally, differentiated and well defined roles in the third. Other aspects that have been found, and that differ between the subcategories, are in the understanding of the function or the purpose of the RMI, whether it is used for data transfer or

[25] By *computer* I refer to a physical computer, a computer that actually can be touched. The term *machine* refers to a *virtual machine*. For a further discussion on *virtual machines*, see section 2.4.

something more than data transfer, or if its purpose is to use resources on different machines. The categories that have been discerned are summarised in Table 5.

Table 5. Categories of description for RMI

	What is RMI understood as?
1a	RMI is understood as data transfer between two specific computers and is described in a concrete way
1b	RMI is understood as something more than data transfer on two specific computers and is described in a concrete way
1c	RMI is understood as being for using resources on specific computers and is described in a concrete way
2.	RMI is understood as being for sharing resources over an internet and is described in a abstract way
3.	RMI is understood as a standard tool in a framework that includes and goes beyond a computer network, and is described at a meta-level

The categories identified correspond in large to the descriptions made of the students' understandings of TCP (see section 4.1). In the project the students could use both TCP and RMI, and the two protocols could to a certain degree be substituted for each others. The fact that the protocols are experienced as being an integrated part of a similar environment can thus be seen as a confirmation of the soundness of the result. There are no internal contradictions in these results.

As mentioned briefly above, three subcategories have been identified within the first category of description. The framework in which the protocol is experienced is the same: Two communicating computers. Also, the protocol is described in a concrete way in the three sub-categories. The roles of the two computers, and the interpretation of what RMI "is" differs between them.

In the first sub-category, 1a, no differentiation or clear roles between the two computers are articulated. Here, RMI is experienced as data transfer. Within the field of computer science, the point of using RMI is to have a possibility to call methods[26] on other computers or virtual machines. The understanding expressed in this category is thus in a technical perspective too simple or even incorrect. For file transfer, there are simpler protocols.

The second subcategory, 1b, describes an understanding of RMI goes beyond the pure idea of transfer, but its features are not clearly articulated. RMI is experienced as something more than transfer, and the two computers involved are assigned different but still unclear roles.

The roles of the two computers or machines are well-defined in the third sub-category, 1c, where RMI is understood as being a tool for using resources that are located at another computer or on another virtual machine. In the second category of description an understanding is expressed where RMI is integrated with an internet as a whole. The purpose

[26] *Methods* are functions or procedures that exist within Java objects. Executing a Java program corresponds to calling or executing methods.

of the protocol, what it "is", is understood as a tool for sharing and using resources over a network.

There is a close correspondence between category 1c and category 2: An understanding of RMI is expressed, where RMI is for using resources in both of them. The critical difference lies in how RMI is experienced: In the second category RMI is understood as an abstract method of sharing resources and as an integrated part of an internet, referring not only to the two concrete computers or machines that communicate, as in category 1c.

Finally, a third category of description has been identified, in which the protocol is experienced as a standard in a framework that extends beyond the network, and that is discussed on a meta-level.

It is worth noting that the experience of the internal technical characterisation is different between RMI on one hand, and TCP and UDP on the other. Although the protocols are recognised by the students as a part of, and integrated with, the same environment, differences in their internal structure are experienced. While TCP and UDP were described in similar ways, the descriptions of RMI differ from this in important ways. In computer science this makes sense: While UDP and TCP are mainly used for data transfer, RMI is used for transferring and executing both code and data, and is thus considerably more complex.

In categories 1a and 1b, where RMI is described as a tool for data transfer or as a communication tool that goes beyond data transfer, the internal technical structure is not clearly articulated by the students. In the second category, the technical structure of RMI is described as an interaction between objects on virtual machines, which is clearly an abstract view of the protocol. For the third category there is not enough empirical material to draw a full picture of the different aspects that are imlied for RMI; although it is possible to identify a way of experiencing RMI in a framework that goes outside computer networks, and thereby to create the category of description, there is not enough data to completely inspect the technical characteristic. The aspects of the understandings of RMI are summarised in Table 6.

Table 6. Aspects of the different categories that describes RMI

	As what is RMI experienced?	**In which framework is RMI experienced?**		**What is the internal technical characterisation?**	**How is RMI described?**
1a	RMI is related to data transfer	Two specific computers	Two computers with undefined roles	Not clearly articulated	In concrete terms
1b	RMI is something more than transfer		Two computers with different roles	Not clearly articulated	
1c	RMI is for using resources		Two computers with well-defined roles	Methods on another computer that is called	
2.	RMI is for sharing resources on an internet	An internet		Interacting objects virtual machines	In an abstract way
3.	RMI is a standard tool	A world outside computer network		Not articulated	On a meta-level

The three sub-categories of category 1 show a clear hierarchical structure. The understanding expressed in category 1a is not very useful for solving computer science problems, and can be seen as an uninteresting special case of the use of RMI or even as incorrect. 1b indicates a somewhat more relevant understanding, since it is closer to the understanding within the field of computer science, but is still not rich enough to be useful to solve technical problems using the features of the protocol. In 1c an understanding is expressed that helps solving practical programming, since RMI is understood as a tool for sharing of resources, and in this sense 1c expresses a better understanding than 1b, which in its turn is better than 1a.

The hierarchical structure that is formed of categories 1, 2 and 3 is somewhat more complex. There is a hierarchical structure in the framework as a part of which the protocols are experienced, although this does not imply that an understanding that is expressed in a higher category, with a broader framework, by necessity includes understandings expressed in lower categories. The structure can be understood in another way as well: An understanding expressed in a higher category is needed to evaluate and judge decisions that are taken based on an understanding that is expressed in a lower category.

4.3.2 RMI in a framework of two computers

In this first category of description we meet an understanding where two communicating computers form the experienced framework of which RMI is an integrated part. But there are differences in how the roles of the two computers are experienced, and these form the basis for the distinction between sub-categories. The experience of what RMI "is" also differs between the sub-categories.

The subcategories are:

- 1a Two computers with unidentified roles. RMI is for data transfer
- 1b Two computers with different, but unspecified roles. RMI is for more than data transfer
- 1c Two computers or machines with different, well defined and specified roles. RMI is for using resources

The communication that takes place and the entities involved are described in concrete terms in all three sub-categories.

<u>File transfer between two computers with undefined roles</u>

In this sub-category RMI is experienced as a method for file transfer in a framework that consists of two computers, where their roles or the functions are not articulated.

An illustration can be found in the second interview with Sven.

Interviewer:	[...] what is RMI? [...]
Sven2:	That is, it is a ... one moves files between, yes....for instance if I were to use RMI that was sort of ... I have the game server and a file that had marbleinfo and so the information on [...] speed so then I want to move over to mine...and then I should use RMI, hard to explain, but ...
Interviewer:	But there are lots of ways to move information what is the thing that is typical for RMI?
Sven2:	Now I am stuck....

Sven talks about RMI as a tool for transferring files. He refers in a very concrete way to the project he is working on and gives an example referring to a specific file that had to be transferred. The term ”Game server” refers to the program module that controls the whole software system in the project, while ”marble info” refers to some specific information about the ball, possibly its speed and position. Sven states that he should use RMI to move a file, containing ”marble info” to ”mine”, most probably referring to the module that he was working on. There are two specific computers in his argument: The computers the file is moved between. Other computers, or a network, are not mentioned.

Anthony also describes transfer of data between computers in his first interview:

> Interviewer: You talked about Java RMI. What is RMI?
> Anthony1: I don't even know. I know it's a type of protocol used between, um, talking between two machines.

The discussion continues and UDP and TCP are discussed. The interviewer returns to the subject of RMI:

> Interviewer: Could you relate RMI to this?
> Anthony1: Um, no. I'd probably say that RMI is just another version of TCP.

Anthony, as Sven, talks about RMI in a framework of two communicating computers, and does not assign them any different roles. Instead, he refers to TCP, see section 4.1.

The understanding expressed in this sub-category is an extreme simplification of the normal use of RMI. It is not a proper way to characterise RMI from a computer science perspective, since it does not capture the specifics of RMI, the features that distinguish RMI from most other protocols, such as TCP or UDP, and makes it possible to recognise RMI as RMI.

<u>Something more than file transfer between two computers with different roles</u>

This sub-category differs from the previous one, since the two computers are described as having different, but yet not clearly defined roles.

Samuel focuses on the communication between two computers, the server and the computer his program is executing on in the following excerpt:

> Interviewer: Can you explain to me what Java RMI is?
> Samuel1: Yes, exactly, but also wants to have some some ... one wants ... what does one say ... one will order or order ... one wants to make a request so to speak, they, that is pretty good, that is sort of, I don't really know ... I now now sit here and speculate here ... I ... I... I don't know so much Java either and I am totally new to this Javathingumy all the time really, but I never worked with Java so that, um, I believe that it is like like some type also, um, ah, protocol to communicate with with servers and such.

He talks about requests and communication with a server. Since he mentions a server, one can deduce that he considers the other of the two communicating computers as a client[27], and in

[27] As mentioned earlier, the active part in the communication between two computers is often called a client while the server is a passive part.

this way assigns them different roles. He does not mention file or data transfer, nor does he have a well-articulated advanced understanding of the protocol. His expressions "order" and "to make a request" as well as his discussion of a server (and implicitly about a client) are relevant for the normal use of RMI as a tool for computer communication. At the same time, the explanation he offers of what RMI "is": "... type of [...] protocol to communicate with servers and such" is unspecific and does not indicate an understanding of how RMI is intended to be used.

During the second interview, when the interviewer returns to the subject of RMI, Samuel expresses a similar understanding:

Interviewer:	[...] What is RMI?
Samuel2:	Remote Method Invocation
Interviewer:	Yaa
Samuel2:	It's something one uses if one wants to find some sort of address which doesn't exist in its own own frame for it for this code which one makes. It it ... it is a concept that understand, but here its used in Java., eh...and Java I don't know anything about actually.
Interviewer:	You have not used that?
Samuel2:	No [...]

He expresses an understanding of RMI, where the protocol is used to find an address, which is not within the frames of the code currently being executed, in order to access external code or other objects via this address.

This sub-category describes an understanding of RMI as a protocol between two computers, where the two computers have different, but undefined roles. The interaction between the computers is understood as going beyond a pure file transfer. In a computer science perspective, this understanding still does not capture the particular properties of RMI.

Using resources on two computers with well-defined roles

In this sub-category RMI is experienced in a framework of two computers, as a tool for executing programs on another machine and in that way to use the resources of another computer.

Staffan gives a description in concrete terms on his view of RMI. During the first interview he says:

Interviewer:	RMI?
Staffan1:	Oh, that's Java's version of client server, it has a stub and a skeleton which one uses. You send you from your client ... you can fetch and allow to execute things from the server via. It feels as if they are local on your ... on your client, but you execute from the server actually.

RMI is used on two computers, a client and a server. The client can execute a program on the server, according to the explanation offered by Staffan. This program is used as if it were residing on the client. In the beginning of his explanation, he talks about a stub and a skeleton[28]. This, together with the fact that he talks about the role of client indicates that he

[28] A *stub* that resides on the client offers the same interface as the server object to a program on the client. It takes the call and passes it to its corresponding server object. The *skeleton* that resides on the server side takes the call of the stub, and forwards it to the server object, waits for an answer, and sends this answer back to the stub. Stubs and skeleton together form a layer in the architecture of RMI.

experiences RMI as integrated with the two computers that are used in the communication process.

He expresses a similar understanding during the second interview:

Interviewer:	[...] What is RMI?
Staffan2:	It's a sort of Java client server model..you can execute. It has something to do with stubs and skeletons. You execute ... I'm not sure how it works ... we had nothing of that kind in our project as it is right now, but ...
Interviewer:	What is stub? What is skeleton?
Staffan2:	Ah, it it generates something ... you have the production on the server anyway and I think that is the skeleton and then this is generated in a way that I don't understand how it works but anyway you can execute those methods, those functions, which ... which are on this or that computer over there even though it seems like they are on your own computer. It something like that.

After stating that he is not sure how it works, he says that it can be used for executing methods or functions on another computer as if it were on your own.

In the following excerpt of the discussion the interviewer starts by referring to an earlier statement made by Stig, where he says that the group was reading about RMI in order to learn more about the protocol:

Interviewer:	[...] We start by RMI, which you had read about. What is that?
Stig1:	It is ... it is Remote Method Indication means that that is with Java in order to...sort of as server client they will be able to communicate with each other. One should be able to use a client, should be able to use code that is on another computer or machine by setting up a connection just ... sort of ... like ...a shell so that one should be able to. It looks like as though one can use, that one like has all the information there, but... communication fetches somewhere else.
Interviewer:	You said as client server: What does client server mean in that case?
Stig1:	Oh, it is that hard to explain ...such things.... you know what it is, but um...it's like a a client mostly program wants to get information from a server.... then they have to communicate with each other in some way and then one can decide...like sort of connect to each other in some way ....it is a special port or something...and that is, yes, that is used ...so we have one of these to the server in ours...in this project.
Interviewer:	And RMI is kind of special case, or?
Stig1:	Yes, I think so. Used when it has to do with Java ... a Java client

Stig clearly states that the purpose of RMI is to offer one machine the possibility to use code on another machine. He also indicates that this has applications, namely when a program wants information from a server. It is clear that he sees RMI as a way of using resources that are available on a computer other than the one you are currently executing your program on. In his explanation he talks about "communicate with each other" and "setting up a connection" on a "special port", clearly focusing on the two communicating machines. Stig's words "on another computer or machine" I understand to refer to a physical computer, but then he broadens his view to include a virtual machine in his explanation. However, he still only refers to *two* computers or machines, not to any network as a whole.

In this sub-category, RMI is understood as a tool for using resources on a different computer. RMI is understood and described in a concrete way in the experienced framework of two communicating computers.

4.3.3 RMI for sharing resources on an internet

In the previous section, a category of description where RMI is experienced as a part of a framework that consists of two computers has been described. The communication that took place between the two computers is understood in three different ways, as transfer, as something more than transfer, and as using resources on another computer.

In this section, the framework that forms the basis for the category of description is different from that of two computers of the previous category: RMI is seen as a part of an internet as a whole, and is experienced as a way of using or sharing resources on the network.

In the excerpt below, taken from the first interview with Abraham, this perspective is clearly visible.

> Interviewer: [...] Um, what is RMI? Java/RMI?
> Abraham1: Ah, Remote Method Invocation.
> Interviewer: Ya, OK
> Abraham1: Very nice. It allows two Java virtual machines to talk to each other. They, an object on one machine could instantiate an object that lives on another machine and use that one's methods. That's how RMI is useful.

Abraham talks about RMI as a tool that offers the possibility for two Java virtual machines to communicate. He then talks about objects that "live" on another machine and possibilities to use the methods of this object. Although he talks about two machines, he does not give any reference to two specific machines. Instead, he focuses on the object, and experiences the machines as a place where the object "lives". In this rather abstract perspective the focus is clearly not on the physical computers, and not even on the virtual machines. Instead, they create the space where the objects live. Thus, the framework is an internet, a broader framework than the one presented in category 1.

Axel, during his first interview, expresses a similar understanding, but is more explicit on the usage of RMI:

> Interviewer: We have talked about RMI?. OK what is RMI?
> Axel1: RMI is Remote Method Invocation which is basically, you have a Java object on one machine somewhere, it doesn't matter where, and then you have a Java object on another machine somewhere, it doesn't matter where. And then you can, either one can call the other, or they can each call each other um. It's, basically, you have to register the object in the RMI registry and then essentially it works just like the other object on the same machine. It is a little bit slower than maybe a socket would be, but it's fairly stable if you can get the security issue right.

Axel explicitly says the objects that call each other may be on any machines. It is not important to him where they are. Having one object call another, or having two objects call each other, implies that they use each other's methods, meaning using or sharing resources. Axel shows an understanding of RMI where the protocol is seen as a way to use resources in a framework of the internet.

Sebastian opens the discussion about RMI by a general description, and then continues his reasoning by going deeper into his explanation during the first interview:

> Interviewer: [...] What is RMI?
> Sebastian1: Yes, that combination of letters stands for Remote Method Invocation, which means that one can call a command from one virtual Java machine on another.

Interviewer: Can you please say a bit more on that?
Sebastian1: Yes ... No ... but that is roughly the picture I have of it actually, I don't know exactly what happens then what ... where the command went.... where it gets executed somewhere, which processor it is that will work on it, which of the two virtual machines it is that...
Interviewer: ... that executes.
[...]
Interviewer: Hmmm, what, what could one use RMI for both in the project and in general. What is the point of the concept?
Sebastian1: No, no, but it feels like I can win something by that ... that ... if I for example if I have a server and a client, so if I should execute a command that is on the server if I can execute it here without having the thing itself so that it executes here so the server gains from that. [...]

Sebastian starts by giving a ”school book” explanation to RMI. However, immediately after giving this explanation, he talks about an aspect of the concept that he does not grasp: He does not understand on which machine the code is executed. The question he raises is, seen in a technical perspective, relevant and can be taken as an indication that the first explanation, although it has a ”school book style” had a meaning for him and was not only a quote that he had memorised from a book. He later gives an argument for using RMI ("it feels like I can win [...] if I can execute it here"), that is consistent with his explanation.

An understanding of RMI, where RMI is experienced with an internet as a framework has been met in this category of description. The protocol is used for sharing resources over the network and is described in abstract terms.

4.3.4 RMI in a framework that goes beyond a network

In section 4.1.4, evidence was presented that TCP could be experienced in a framework that went beyond a computer network, and that also considered human decisions. TCP was discussed as a standard communication tool, and the students expressed an understanding of TCP at a meta-level. A similar way of experiencing RMI can also be identified.

In the excerpt below, Adam discusses the choice of TCP, instead of RMI, for all communication throughout the code of the project:

Adam1: Between, like the game server and the video and motor, you mean? [...]
Interviewer: And you will just accept that they are TCP. So what you do is that you go for overall a TCP solution. OK Ya.
Adam1: Right. And it's my impression that it doesn't matter what one part communicates in, because if it is communicating with RMI to the client, but with TCP to the motor, I mean it's just different ways of formatting the information, in a sense, so..
Interviewer: Ya, ya.
Adam1: If it isn't TCP, you know, it doesn't really affect..

He argues that RMI and TCP can be seen as different ways of formatting the information, where different protocols can be used to solve different sub-problems. He mentions communication with the motor and the technical communication between the server and the client. The choice between the protocols, seen in this perspective, is not important, continues his argument. He talks about the two protocols as two different ways of formatting data and as two different instances of the same phenomenon. The comparison that he makes requires him to reason about the protocols from an outside position, where properties of individual protocols are abstracted; that is, he talks about RMI at a meta-level.

Alec also uses a meta-level reasoning to express his understanding of RMI:

Alec1:	[...] But it's very lengthy and verbose, as far as a lot of work, and this RMI is quick and concise, but it seems to take away some of the flexibility.
Interviewer:	Uhum.
Alec1:	There are probably ways to do things that I'm not talking to.., that I'm unable to do now, that I'm not aware of but, um, as of now it seems to take away some of the flexibility. I've also discovered that there's, like you were discussing, the security, which has to do with a..
Interviewer:	Uhum
Alec1:	[...] a policy file that, um, that I have little or no knowledge of, just discovering it, but that I've begun some research on it and, um, as far as how that works. [...]

This is a meta-level discussion of RMI. Alec says that there might be solutions he does not know at the moment of the interview. His judgement, that the solutions he has found are inflexible, and that there ought to be other solutions, demands that he takes a position outside RMI, where he can abstract and talk about what properties he expects the protocol to have to be a good standard tool. Also, this argument requires that he is consciously aware of the fact there are decisions taken on the design of RMI.

In this category, RMI is understood as a standard tool and is experienced as a part of a framework that goes beyond a computer network and that is described a meta-level.

5. Students' understanding of the general concept of a network protocol

In the previous section an analysis of how students understand TCP, UDP and RMI as individual network protocols has been made. A question that naturally arises in the context of this section is what common properties of network protocols are experienced, and what could be said about students' ways of experiencing the general concept of a "network protocol"? In this section I will explore this issue further.

5.1 Different ways of experiencing network protocols

An analysis of the students' understanding of the concept of network protocols as a whole could be made in several ways. An obvious alternative option would be to ask the question to the effect: "What is a network protocol?" during the interview. However, such a question was not asked, and the issue of the general concept of a network protocol was not raised explicitly during the interviews.

Another possibility would be to re-analyse the interview extracts concerning the individual network protocols in the light of the analysis made for the individual protocols and the categories of description that were created. This re-analysis could form the basis for a possible creation of categories for the concept of network protocols. Yet another possible attempt is to go directly to the interviews to look for statements about protocols in general during the interviews about the specific protocols. In this section, I combine the two latter approaches. Thus the original analysis form a background to the interview extracts that in different ways address the general concept of a network protocol. The statements are, in other words, recontextualised at a collective level, as is proposed in Adawi, Berglund, Booth and Ingerman (2002).

An important aspect of the categories of description that were created for TCP, UDP and RMI is the framework of which the protocols are experienced as parts. As was pointed out in earlier sections, the protocols are experienced as integrated with their environment. They would not exist without the environments to which they belong, and the environment in which they can be found would not be the same without them. In other words, there exists no computer communication without some kind of communication or network protocol.

The frameworks that have been identified for the individual protocols are
1. two communicating computers
2. an internet
3. a world beyond computer networks

The framework can be seen as one aspect of the ways that protocols are experienced, and since the analysis has given similar results on the framework for all three of them, it can be assumed that this aspect is also relevant for the experience of frameworks or backgrounds for the idea of a network protocol. I use this as a starting point, and I will explore this question further by considering these categories of description alongside some interview extracts on the general concept of a network protocol.

The findings for the general concept of a network protocol are summarised below and in Table 7. The next sections present the evidence leading to these results.

Table 7. Ways of experiencing the general concept of a network protocol

	What is the general concept of a network protocol experienced as?	**Which framework is the concept of network protocols experienced as integrated with?**
1	A protocol is a way of talking/communicating between two machines	One (or more) specific computers
2	A protocol is a method of communication on an internet	An internet
3	A protocol is a set of rules that are used on an internet	An internet
4	A protocol is a standard	A world that goes beyond computer networks

Four categories of description have been identified. The critical difference between the four is the qualitative ways in which the general concept of a network protocol is experienced. Two of the protocols are experienced in the framework of an internet. As I will argue in the coming sections, the qualitative differences between the categories are important, and as a consequence, separate categories do more justice to the data.

The first category of description expresses an understanding where a network protocol is experienced as a way of talking or communicating between two computers. The critical difference between categories 2 and 3 is how, or as what, network protocols are experienced. Category 2 describes an understanding where a protocol is a method of communication over an internet, while 3 expresses and understanding where a protocol is experienced as a set of rules. The general concept of a network protocol is experienced as a standard, and as such related to human decisions in category 4.

As was the case for the individual protocols, a hierarchical structure can be identified. In the higher levels of the hierarchy the protocol is experienced in a wider framework. It can also be argued that the understanding of what a network protocol "is" shows a hierarchical structure, where the level of abstraction increases from category 1 to 4. In category 1, a concrete understanding is described, with communication between two specific machines. In 2, the protocols are understood as methods of communication, which represent a more abstract understanding. With the experience of network protocol that is described in 3, the protocol is understood as a set of rules. A set of rules, is, by its pure definition, an abstract entity. And finally, in category 4, the understanding of a protocol can be described as a standard. This understanding demands, as was mentioned earlier a possibility to reason about the properties of the rules, and from where they stem, not only to see the rules themselves.

5.1.1 Network protocol as a way of communicating between two computers

In this category of description an understanding is expressed where network protocols are experienced as methods of communication, or methods of talking, between two computers.

Anthony articulates such an understanding during his first interview:

> Interviewer: You talked about Java RMI. What is RMI?
> Anthony1: I don't even know. I know it's a type of protocol used between, um, talking between two machines.

He says that he only knows that RMI is a protocol that is used for communicating between two computers. The interviewer continues by about TCP:

> Interviewer: Uhum. What is TCP?
> Anthony1: TCP is another type of protocol .. used between two machines. There is TCP and there's UDP that's one of the things that I actually do remember from ah, networking class. And I believe TCP sends packets to one machine and then there is some sort of response saying that they got the packets or not.

He talks about TCP as another type of protocol, which is also used between two machines. He spontaneously mentions UDP, as another protocol. By referring to the three protocols in this way, it is clear that Anthony experience that properties are shared between protocols: The three protocols mentioned are for communication or talking, and are experienced in a framework of two computers or machines.

5.1.2 Network protocol as a method of communication over an internet

The general concept of a network protocol is experienced as a method of communication over internet in category of description 2.

During the first interview with Albert, he expresses an understanding of TCP as a method of communication over an internet in a statement that has in part been discussed in section 4.1.3:

Interviewer: Um, what is TCP?
Albert1: TCP, um, it's um, part of the internet protocol. It's used with part of the internet protocol typically. Um, it's one of the methods of communications, I don't know a whole lot about it, as far as the whole, um, design construction behind it. Um, I'm learning it pretty deep, in depth in my.. I'm going to learn it pretty in depth in my network class, but um, I can't think of what TCP stands for right now.. Transfer Call Pad .. I can't remember off hand. But it, it is used as one of um, the other, as one of the connections as also, as is UDP, a TCP/IP protocol.

In this dialogue about TCP Albert mentions UDP as a protocol with similar characteristics, without being sure of the exact differences between the two. TCP is, according to him, a part of the Internet protocol, and is one of the methods of communication. In other words TCP is a method of communication that is related to an internet, and not only two communicating computers.

When prompted on the differences between the two protocols, Albert stresses the similarities in his answer:

Interviewer: O.K What is difference?
Albert1: I don't know (laughter).
Interviewer: That's fine, that's fine.
RedU1: I know, I know that it's part of it and it's separate. But it's just a different type of protocol that you use to communicate. I know that, but..

Later during the interview, Albert mentions RMI when answering a question about sockets:

Interviewer: There is another word you mentioned there, and that's socket. What is a socket?
Albert1: A socket is pretty much like a, a port that is opened up on the server, or that is requested by the client and, it's assigned a number. And it's just sitting there and listening and um, it's just an open port and that port is um, designed to use a specific type of protocol, you know whether it be TCP, um, or the RMI. And it's opened up to listen on that and once it receives that connection you know, it connects on that port. So it's like an outlet socket, you know, you connect it in, you communicate and then when it's down it gets turned off and then that port is either closed or it stays open if it's required by the server.

On reading Albert's statements, it is clear that he experiences UDP, TCP and RMI as being protocols that share important properties and that are basically experienced in a similar way, and a part of the same framework. In the interview with Albert there are similar statements indicating a relationship between the protocols. There are differences between the three, but they are all closely related, as he experiences them. From this I deduce that my interpretation of his understanding of what the general concept of a network protocol is, is relevant also when related to UDP and RMI. Protocols are methods of communication.

Axel experience TCP and RMI as methods of communication in a framework of an internet as well:

Interviewer: OK That's fine, that's fine. Um, I want you to talk about TCP.
Axel1: TCP/IP?
Interviewer: Ya.
Axel1: TCP/IP is how almost everything on the Internet communicates. IP addresses and everything, and that's um, one of the fundamentals behind RMI also. One could give it the address where the object is [...] the IP address [...]

As was shown in a previous section (section 4.1.3) he understands UDP as a protocol that is similar to TCP. The three of them are clearly experienced as integrated parts of Internet. He talks about the protocols as "how almost everything [...] communicates". I interpret this, together with his statement that TCP is a fundamental protocol, as an expression of an understanding of network protocols as a methods of communication over an internet.

In this section, a category of description has been identified, where the general concept of a network protocol is experienced as a method of communication in a framework of an internet.

5.1.3 Network protocol as a set of rules

The experience of the general concept of a network protocol is related to an internet in category 3 and is understood as a set of rules.

Allan says that it is a protocol language used for sending data across a network, in an excerpt that was discussed in section 4.1.3

> Interviewer: Um, you've talked about TCP. What is TCP?
> Allan1: Basic concepts... it's a protocol language, I guess you can call it, that you just put your data in and it's sent across the network using the different protocols you want to use, like IP or.. I can't think of any other protocols off my head. But it is more or less a packet that you put your data in and you send across and it has some features such as, keeps things in order when you, um, when you get to the, um, when it gets to the server you want to go to. [...]

His statements that TCP is "a protocol language", that you "put your data in", and that different protocols might be used for the actual transfer, mean to me that he experiences protocols as a set of rules, since a protocol language in computer science is a formal language or a set of rules. He does not regard his answer as only valid for TCP, since he talks about different protocols, without wanting to mention or without being capable of mentioning others by name. By mentioning IP, Internet Protocol, Allan relates to an internet.

Adrian tells the interviewer during the first interview that his group plans to remove RMI:

> Interviewer: Um, what is RMI? What is Java/RMI? The thing that you're removing?
> Adrian1: I don't know, and that's why we're removing it.
> Interviewer: OK
> Adrian1: 'Cause we don't know enough about it. I, it's.. I've read briefly whole paragraphs about it. It's basically enabling it to get around security features that TCP/IP wouldn't allow. Um, or standard HTTP protocols. Um, like RMI, I guess allows complete access to certain files. Whereas if you go to HTTP, it's going to be a little bit slower, and you, there you have to worry more about the security issues, what you want to have access to.

The group plans to remove RMI to get around certain security features, and profit from, as they understand it, the less severe rules of TCP. To get around security features is to avoid certain rules, since the security features mainly consist of rules that govern certain operations that guarantee security. From this argument and his discussion about allowing access, I interpret that he experiences the protocols as sets of rules, that is, rules that are somewhat different for different protocols.

In other words, he expresses an understanding of network protocols as a set of rules experienced in a framework of an internet.

5.1.4 Network protocol as a standard

Adam explains during the first interview what a network protocol is, in a statement that has earlier been discussed in section 4.1.4:

> Interviewer: So what is TCP then?
>
> Adam1: Well that I have studied in some networking classes um, Transfer Control Protocol, something along those lines. Um, that is just a protocol for computers to communicate with each other. That's a standard that was created by a committee somewhere, sometime, and it's just a, it's a protocol, meaning that it's, it specifies um, the layout and the size and what's in the header and footer of packets being sent across networks and things like that. So it's, it's a standard communication tool

He starts by saying that the purpose of a protocol is to get computers to communicate. He then points out that TCP is a standard for a protocol, which was created by a committee. A protocol, in its turn, specifies the format on data sent across the network. TCP is, with this understanding, one of many protocols. A standard is, according to Adam, a set of rules that are created by a committee, that is, a result of human decisions.

In this category of description, we have met an understanding where a network protocol is experienced as a standard in a framework that goes beyond a computer network.

6. A discussion on learning and teaching

As has been stressed throughout the report, the objective of this phenomenographic research project as a whole is to gain insights in the students' learning of computer communication when taught in an internationally distributed project-oriented course. This report focuses on variations in the students' experience of network protocols, while my future work will study variations in learning in the context of the course and the interplay between their experience of learning and the context they experience.

Different ways of experiencing the concept of network protocols in general as well as the three specific network protocols TCP, UDP and RMI have been identified and presented. A network protocol is, of course, understood in a context by an individual. This means that an individual experiences the protocol against the background of and interacting with a specific environment. In the analysis (see section 3.2.2) this background is stripped away; in other words, the statements made by individuals are decontextualised. The decontextualisation is an analytical tool for the researcher to draw conclusions about the distinctly different ways a phenomenon, as for example RMI, is experienced within the group. The individual statement is then, as has been described earlier, recontextualised through a dynamic process into a context at a collective level that is created by the researcher: the outcome space of the categories of description. The coming sections will explore and develop the results presented in earlier sections and related them to learning and teaching.

6.1 Shifts between different ways of experiencing network protocols

In order to address the issues of learning and of what constitutes a "good understanding" of computer networks, I consider the results in the context that they originally stem from, that is from the interviews. I will show that there are shifts between different ways of experiencing a specific phenomenon, and from this I will draw conclusions about learning.

Categories of description can only be created by the researcher for a group, at a collective level. Individuals experience particular phenomenon differently at different moments, which is to say that shifts can occur spontaneously and rapidly. With a distinction that was articulated by Pong (1999), shifts in focus can occur as inter-contextual shifts, when the context shifts, that is when a new subject is discussed, but also as intra-contextual shifts within the same context, either spontaneously by the student or as a part of a conversation.

Many intra-contextual shifts have been identified in the data that forms the basis for this paper. The students in this study are advanced students in computer science in their third or fourth year, and as such they have had the opportunity to meet different views from their teachers, books etc on computer science. This might be a reason why they take different stands on various computer science issues throughout their studies.

6.1.1 Case studies on shifts between different ways of experiencing network protocols

An example of such an intra-contextual conceptual shift from experiencing TCP as communication between two computers, expressed in concrete terms, to experiencing TCP as related to an internet, expressed in abstract terms, can be found in the following part of the first interview with Anthony:

> Interviewer: Uhum. What is TCP?
> Anthony1: TCP is another type of protocol .. used between two machines. There is TCP and there's UDP that's one of the things that I actually do remember from ah, networking class. And I believe TCP sends packets to one machine and then there is some sort of response saying that they got the packets or not [...]

Here, in the first part of the discussion of TCP, Anthony tells the interviewer that he understands that TCP is used between two machines, for the concrete purpose of sending packages. TCP has, according to him, a kind of response that indicates whether a package has arrived or not, that is, TCP has an acknowledgement.

The dialogue continues:

> Interviewer: So what's the implications of this?
> Anthony1: Um, it, it all depends on how you're coding it. It depends on how secure the network you're on. And if you actually trust just sending it out and just assuming that it gets there.

When the discussion continues Anthony gets a question about the implications. He argues that the implications depend on how "you are coding it", that is what your program actually does, and your understanding of the quality of the network.

His focus changes thus from experiencing packages sent between two machines to experiencing TCP as a part of a network that he discusses in abstract terms and assigns properties, like trust. In this case, the shift was triggered by the interviewer asking a question that encouraged the student to reflect further on the subject. .

Another example of an intra-contextual shift that a student spontaneously made during the interview can be found in the continuation of the extract of Sebastian from section 4.1.2. He says:

Sebastian2: Yes, an acknowledgement, That is, that I know that the information I send has arrived correctly, and what comes back has also arrived. There is a bunch of other stuff that I have to look out for. That the communication really works as it should, yes, between two software-created gadgets, that are sockets and ports.

By the end he mentions sockets and ports as "software-created gadgets". Here he shifts his focus and talks about abstract items, and thus expresses another way of experiencing the protocol.

Similarly several shifts between different ways of experiencing RMI have been identified. An example of an inter-contextual shift can be found when comparing the following two excerpts of the first interview with Albert. In the first excerpt, the discussion is about the changes the group has decided to make to their project (see section 2). The interviewer introduces the question of the changes, but the concrete change that gives the direction to the continuation of this part of the interview, comes from the student.

Interviewer: Ya, exactly.
Albert1: Um, the client and server separation, um, is going to involve a little bit more. In fact it will probably involve quite a bit more. The reason for that is because it looks like they're really, really close on the way it was structured using RMI and the, the layout of the classes and the way that the classes used each other, but it was really kind of odd when we started looking into it and the way that they structured it and the way that they're trying to send information back to the client. Um, the way that they currently send, like the path that was run, back to the client so you can see the path that was run, um, was that the navigation class sent it directly to the client. And the way it receives the client object as it's passed from the server, the client calls the method and it passes itself to the server and then the server passes it to the navigation. And then the navigation class uses that object, the client object, to call a function to send the path that was run back directly to the, excuse me, client.
Interviewer: Back to the client?
Albert1: Yeh. Back to the client, which, um, the way that we've been reading about RMI, is not the way that it should be done. [...]

In this context, the client-server separation, which is one of the changes that the group has decided to make, Albert discusses in detail the interaction between the server and the client. He mentions data that is sent, and discusses which methods that are called and on which objects they can be found. He clearly expresses an understanding where RMI is used as a tool for using resources and is seen in relation to two specific machines: the client and the server.

Later during the interview, the interviewer asks him about RMI:

Interviewer: You have mentioned some here. What is RMI, could you explain that to me please?
Albert1: Um [...] But it stands for remote method invocation and what it is, is you have an interface that is, um, that a class, um, uses this interface and the client also uses this interface. And the way that that happens is that, um, the server implements the methods to be used remotely. So there is only a few methods that are being used remotely, and it registers in the RMI registry the name of the object. And so when the client wants to use those methods, what it does is it does a look-up on that server. You pass up the server and you pass up the object that you want to look up. And what it returns is that it returns that object, and then, in this client,

you use that object as you would a local object. You could call functions on it, stuff like that, um..

Interviewer: You use it as if, as if it was a local object although it is an object on the server.
Albert1: Correct.
Interviewer: OK
Albert1: So you can call methods that are actually located on the server and it does things like that. And that's the basic concept of it. It gets pretty involved if you do call back, um, which I haven't been able to find too much on..

This time he explains the function of RMI, without referring to any specific machines or computers. He does not mention explicitly that objects can be on any machine. However, his use of the words "remote" and the general attitude in his explanations clearly indicate that he experiences RMI in a framework of an internet.

Another case of shifts can be found in the first interview with Alec:

Interviewer: You are going to Java RMI, what is RMI?
Alec1: It's um, a remote method communication. Um, Java sets up interfaces between two, let's see, um, classes, objects, and in the interface are methods that are available to the other class. And nothing else within the class. [...]

Here, he expresses an understanding where RMI is related to an internet (category 2), especially by mentioning "remote method communication". He articulates his understanding in an abstract way using words like "class" and "objects" and talks about the methods in the interface. He continues:

[...] RMI starts a connection on the port. It's not really a port, it's a registry number, and between, on that registry number they can communicate but only in the interface between the two. Um, I found, right off the bat, that you can't just compile these classes regularly. There is a RMI compiler. The RMI compiler creates two classes, a stub-class and a skeleton-class and these are needed for the communication between the interfaces. These are set up, um, separately to the communication. Um, I found that particularly interesting because it takes a lot of the work out. The hard coding I know and C++ I've seen the coding, I've never actually coded it. [...]

The discussion here moves towards coding, how to make a particular connection, between two specific machines or computers to work, and expresses a way of experiencing RMI, that is described in category 1c. He chooses words that have concrete denotations.

The continuation is interesting:

[...] But it's very lengthy and verbose, as far as a lot of work, and this RMI is quick and concise, but it seems to take away some of the flexibility.

Interviewer: Uhum.
Alec1: There are probably ways to do things that I'm not talking to.., that I'm unable to do now, that I'm not aware of but, um, as of now it seems to take away some of the flexibility. I've also discovered that there's, like you were discussing, the security, which has to do with a..
Interviewer: Uhum
Alec1: .. a policy file that, um, that I have little or no knowledge of, just discovering it, but that I've begun some research on it and, um, as far as how that works. [...]

This excerpt can be understood as a meta-level discussion about RMI. He says that there might be solutions he does not know at the moment of the interview. His argument is that the solutions he has found are inflexible. To make this judgement, that the solutions are inflexible, and that there, as a consequence, ought to be other solutions, demands that he takes a position "outside" RMI, where he can talk about what properties he expects the protocol to have. This is an indication of a shift to experiencing RMI as related to a world that goes beyond computer networks (category 3).

Alec has made spontaneous inter-contextual shifts from 2 to 1c and further to 3.

6.1.2 Implications of shifts in ways of experiencing a protocol

In the discussion about shifts between different ways of experiencing network protocols three qualitatively different types of shifts have been identified: spontaneous intra-contextual shifts, triggered intra-contextual shifts, and inter-contextual shifts. These results harmonise well with the results about inter-contextual and intra-contextual conceptual shifts that are articulated by Pong (1999).

The cases of shifts that have been studied are summarised in Table 8.

Table 8. Cases of shifts between different ways of experiencing network protocols

Name of student	Type of shift(s)	Categories of descriptions for the shifts
Anthony	triggered intra-contextual	$1 \rightarrow 2$
Sebastian	spontenous intra-contextual	$1 \rightarrow 2$
Albert	inter-contextual	1, 2
Alec	spontenous intra-contextual	$2 \rightarrow 1 \rightarrow 3$

For the the intra-contextual shifts, the table shows the order in which the students expressed a certain way of experiencing the protocol, since the shift happened during a single episode of the interview. In the case of inter-contextual shifts, the order is not relevant, since the different ways of experiencing the protocol were expressed during different parts of the conversation.

Although there are many cases of shifts within the data, this does not imply that all students shift between all understandings. For each individual, it is possible to identify the most advanced understanding he shows during the interviews. With some rare exceptions all shifts found in the data are between categories 1 and 2.

Also, there are students who, although provoked by the interviewer, just express one way of experiencing the protocols. As examples of students who do not shift during the parts of the interviews that are analysed for this report can be mentioned Sven, who expresses an understanding that can be identified as category 1 (see section 4.3.2), and Adam (see section 4.1.4 and 4.3.4) who expresses a stand that is described in category 3.

6.2 Conclusions about learning

It has been said earlier in this report, but it is worth repeating: The way(s) a certain student experiences a specific phenomenon, such as RMI, can change. Nor does a student have a given limit for the capacity to reach an advanced understanding. Rather, the student interacts with the phenomenon. His understanding of the phenomenon is then shaped by the phenomenon in the context in which the phenomenon is experienced, in the environment where the learning takes place, and the student him- or herself with his or her interests and previous understandings. Thus, it is worth studying what constitutes a good understanding, and how the universities can act to promote good understanding among their students.

Marton and Booth (1997) argue that relevant or meaningful learning is a change in the learner's capability of experiencing something in a new or different way. This definition of learning does not only indicate that some learning is meaningful, but also points out that there are less relevant forms of learning. Pure rote-learning without a related different or deeper understanding, or the learning of a new program construct that is not related to or does not offer any new possibilities to develop thinking or programming, are, according to this argument, not examples of meaningful learning.

They also discuss good learning and argue that the ways in which learning is experienced "differ in richness (different aspects of learning that are discerned and held in focus simultaneously) and situational appropriateness (which particular aspects held in focus under the prevailing conditions)." (Marton & Booth, 1997, p. 55). I will take this as a starting point for a discussion about situational appropriateness and richness of the students' experience of network protocols.

6.2.1 Situational appropriateness of ways of experiencing network protocols

A conclusion that can be drawn from the argument of Marton and Booth, mentioned above, is that the task at hand indicates which way(s) of experiencing a protocol are the most fruitful.

Relevance for programming

A way of experiencing a protocol in a framework of two computers and described in a concrete way is closely related to programming. The descriptions made by the students resemble the terminology that is used in different programming situations that relates to communicating computers or machines. It can be assumed that this perspective is fruitful for solving concrete programming issues.

A quote from Sebastian can illustrate this. On a question from the interviewer about UDP, he compares UDP and TCP:

> Interviewer: UDP?
> Sebastian1: UDP.... but that is another form of communication. TCP/IP is set up ... like TCP, in contrast to UDP, TCP sets up communication between two points, and they talk to each other and make sure that they don't drop anything sort of.

As was explained in section 2.2, TCP and UDP offer procedures, or operations, to a programmer who writes application programs. The procedures for TCP offer services like setting up a connection or sending data. The statements by Sebastian above can directly be related to programming issues for using TCP in an application program. Similarities between his statements and some basic operations on TCP sockets are shown in Table 9.

Table 9. Similarities between Sebastian's statements and basic TCP operations

Sebastian's statements	Basic TCP operation
set up	Connect to a remote machine
talk to each other	Send data
talk to each other	Receive data
implicit, a connection that is set up, also has to be closed	Close a connection

In the continuation, Sebastian returns to UDP. His way of talking is still close to the issues of programming:

> Sebastian1: UDP is [...] that the client asks what does this mean. Or what is this, or any question, whatever, and, so the server answers. And the server doesn't care in the end if the answer gets there or not. It is only a question and an answer, and then it is up to the client. If it feels that I didn't get any answer, it gets to ask again.

Here he talks about what a client that uses UDP has to do: If no data has arrived, the client has to repeat the question. This line of reasoning is close to the steps taken by a program that uses UDP.

Relevance to program design

A framework where the protocol is experienced as an integrated part of a network is useful for discussing the properties of protocols or which protocol to use in a particular situation. Issues like in what situations and in what way a protocol is useful come into focus here. It can thus be assumed that this way of experiencing a protocol is fruitful for design purposes.

An excerpt of the first interview with Abraham (see section 4.3.3) can serve as an example:

> Interviewer: [...] Um, what is RMI? Java/RMI?
> Abraham1: Ah, Remote Method Invocation.
> Interviewer: Ya, OK
> Abraham1: Very nice. It allows two Java virtual machines to talk to each other. They, an object on one machine could instantiate an object that lives on another machine and use that's one methods. That's how RMI is useful.

Abraham explicitly discusses the advantages of RMI when asked what it is. He clearly has an understanding of what the purpose of RMI (use objects on another machine as a resource). This understanding is useful for deciding when to use RMI, and when to choose another protocol.

Relevance to policy issues

The meta-level discussions that concern what possible protocols there could be, and what properties they could have, characterise an understanding that is described in the third category. This understanding is useful for policy discussions.

This position is clear in the quote below (see section 4.3.3), where Alec argues that he is not aware of all features of RMI:

> [...] But it's very lengthy and verbose, as far as a lot of work, and this RMI is quick and concise, but it seems to take away some of the flexibility.

Interviewer: Uhum.
Alec1: There are probably ways to do things that I'm not talking to.., that I'm unable to do now, that I'm not aware of but, um, as of now it seems to take away some of the flexibility. [...]

Alec argues that RMI, as he understand it, is quick and concise, but it is not as flexible as he thinks it ought to be. His conclusion is that he does not know the features of the protocol well enough. Reasoning thus, he discusses what properties a protocol should have. This line of argument is relevant when considering policy questions, as how to design network protocols.

6.2.2 Richness in ways of experiencing network protocols

In the previous section I have argued that different ways of experiencing network protocols are useful for different tasks at hand. I have pointed to the need for a fruitful variation, by showing examples of the relevant ways of experiencing the specific network protocols. Different ways of experiencing a network protocol are shown to be useful for solving different kind of practical problems. The examples given above are intended to illustrate the relevance of being capable of experiencing a phenomenon in different ways. I do not argue that the examples show the only or not even the principal situation when a particular way of experiencing a phenomenon is useful.

Another argument for different ways of understanding network protocols being useful is presented in section 4.1, where it is argued that an understanding expressed in a higher category of description offers the broader perspective needed to inspect and evaluate an understanding expressed in a lower category of description. An example can serve to illustrate and concretise the reasoning. To evaluate the solution to a problem solved in a concrete way concerning two interacting computers, as for example the coding of a program using TCP, it is necessary to shift to an understanding where the program is experienced in the framework of a network, and is discussed in an abstract way. By "stepping outside" the original reasoning to look at the problem as an issue of design instead of as an issue of coding, questions about the relevance of the solution can be discussed.

For solving complex or new problems it is thus necessary to shift between different ways of experiencing a protocol, since problem-solving involves different sub-tasks. To shift perspective, whether a shift is intra- or inter-contextual, triggered in a discussion or spontaneous, is not alone sufficient for problem-solving. Shifts have to be made in a relevant way, and the student needs to be capable of evaluating when and why a specific way of understanding a protocol is fruitful.

6.3 Implications for teaching

In section 7.1.1, we have seen that there are students who shift in situationally relevant ways between different ways of experiencing network protocols. It is also argued in the previous sections that an objective for a teacher is to promote variation in the ways network protocols are experienced. In this section I combine these two conclusions and discuss implications for teaching and for future research concerning teaching of network protocols.

Marton and Pang (2001), Pong (2001) and Lo (2001) among others have studied the relation between teaching and students' understandings. They argue, based on empirical research, that a meaningful variation in the presentation of a phenomenon in a teaching situation improves learning. Thus, a first implication of their conclusion is that a teacher of computer networks should create a variation in how he or she presents the concepts he or she wants the students to understand. However, in situations where the students work in projects, this kind of advice, although good, is hard to implement directly.

Still, since a systematic variation in the students' experience of a phenomenon is desirable, it is worth further developing the issue of variation. A point of departure is Adawi, Berglund, Booth and Ingerman (2002), where we argue that variation in the context in which a phenomenon is experienced during a research situation supports that the phenomenon is experienced in new or different ways.

In our work we make a distinction between two different meaning of context in phenomenographic research projects. The *prepared context* is "defined by or observed by, or indeed experienced by, the researcher; that is, what the researcher considers to be relevant for the interviewee to make sense at the situation at hand", while the *experienced context* is "experienced by the participant; that is, what the participant experiences as being relevant for making sense of the situation at hand". The two terms do not denote two different contexts, but express two different ways of seeing and analysing a context and serve in this way as an analytic tool.

A similar distinction can be made in a teaching situation: A teacher presents a phenomenon in a context, which he or she prepares for the discussion with the students. We do not know how an individual student experiences either the phenomenon or the context. In a phenomenographic research project we can discern different ways of experiencing the phenomenon within a group, but we can only get glimpses of the ways the context is experienced. Still, the phenomenon is not experienced in a vacuum by a student. The phenomenon is experienced in the context the student experiences. We can only speculate on what constitutes this experienced context. The student's study objectives, his previous understanding of the phenomenon, discussions with other students, and the physical learning situation are some factors that to different degrees, together with the prepared context as offered by the teacher, form the experienced context for a student.

Thus our work in Adawi, Berglund, Booth and Ingerman (2002) offers another approach to the issue of variation in how a phenomenon is understood. A teacher can, and ought to, create a teaching situation in a way that promotes variation in how the context of the phenomenon taught is experienced. For a project course, like Runestone, where no lectures are held, this approach is promising.

In my future work concerning the Runestone initiative, I will study the relation between the learning and the context of the learning in order to identify factors that promote variation and thereby good learning. My belief is that the project work in Runestone, through the interaction between the participants of the project group jointly aiming at attaining a shared goal, will support that phenomena being experienced in different contexts and thus in different ways. I will then use the results presented in this report, as well as the full pool of meaning that has been collected. Focus will be on how variation is created and how contextual factors affects variation. However, it remains to be seen if the belief holds, and in that case, in what way variation is encouraged.

7. *Conclusions and summary*

In this report, I have presented university students' understanding of network protocols. The students, who are advanced students in computer science, have taken part in an internationally distributed project course that is jointly taught by two universities. The aim of the student project is to produce a software system to control the movements of a computer-controlled mechanical toy.

Qualitatively different ways of understanding or experiencing network protocols are discerned in this study, which has been carried out with a phenomenographic research approach. Based on these results a discussion about learning and teaching is given. It is argued that a variation in the context in which the protocol is experienced promotes good learning, since different ways of experiencing a protocol are useful with different tasks at hand. A student with a good understanding of network protocols can choose in a situationally relevant way between different ways of experiencing a protocol.

The ways in which students understand three specific network protocols – TCP, UDP and RMI – as well as the general concept of a network protocol have been studied. Although the protocols are experienced as different by the students, the three protocols are understood as being parts of similarly experienced frameworks. The three qualitatively distinct frameworks consist, respectively, of two communicating computers, a computer network, or a world beyond computer networks.

When TCP is seen in a framework of two computers, it is understood as a safe way of communication, while RMI seen in the same framework is understood: as a tool for data transfer, as something more than data transfer, or as a tool for using resources. This way of understanding is discussed by a student in a concrete language, and is found to be useful for solving programming problems.

In a framework related to an internet, TCP is experienced as a connection, while RMI is a tool for using resources. Discussions related to this way of experiencing protocols use abstract concepts and can be shown to be fruitful for taking design decisions.

When the two protocols are seen in a framework that includes, and extends outside, computer networks they are experienced as standards. Human decisions are taken into account in meta-level discussions about the networks. The understanding that is expressed in this category of description is relevant for policy discussions.

Different ways of experiencing the general concept of a network protocol are also discerned, related to the three frameworks described above. A network protocol can be understood as a way of communicating between two computers, a method of communication, or set of rules or a standard in a world that extends beyond computer networks.

Based on the results presented in the report ways to improve teaching of computer networks are discussed. It is proposed that, universities should teach computer networks in a manner that encourages students to understand network protocols in different ways, and that stimulates them to shift between these ways depending on the task at hand.

8. References

Adawi, T., Berglund, A., Booth, S. & Ingerman, Å. (2002). *On context in phenomenographic research on understanding heat and temperature*. Revised paper presented at EARLI 2001, Fribourg, Switzerland, 2001. Submitted to Learning & Instruction.

Booth, S. (1992). *Learning to program: A phenomenographic perspective*. Gothenburg, Sweden: Acta Universitatis Gothoburgensis.

Budd, T. (1999). *Understanding Object-Oriented Programming Using Java*. Reading, MA, USA: Addison-Wesley.

Daniels, M. (1999). *Runestone, an International Student Collaboration Project. NyIng report No 11*. Linköping, Sweden: Linköping University 1999. Also available at `http://www.docs.uu.se/~matsd/NyIng.html`

Derrick, J. & Fincher, S. (2000). Teaching Communication Protocols. *Computer Science Education,* 10(3). 195 – 202.

Lo, M.L.(2001). *Variation in Practice*. Presented at EARLI 2001, Fribourg, Switzerland.

Marton, F. & Booth, S. (1997). *Learning and Awareness*. Mahwah, NJ, USA: Lawrence Erlbaum.

Pang, M.F., Marton, F. (2001). *Bringing Learning about*. Presented at EARLI 2001, Fribourg, Switzerland.

Pears, A., Daniels, M., Berglund, A. & Erickson, C. (2001). *Student Evaluation in an International Collaborative Project Course*. Presented at the Wise workshop of the SAINT conference, San Diego, CA, USA.

Pong, W.Y. (1999). *The dynamics of awareness*. Presented at the 8th EARLI conference, Göteborg, Sweden.

Pong, W.Y. (2001). Making Conscious Use of Variation. Presented at EARLI 2001, Fribourg, Switzerland, 2001.

Stalling, W. (1997). *Data and Computer Communications*. Upper Saddle River, NJ, USA: Prentice-Hall.

Tanenbaum, A. S. (1996). *Computer Networks. Third edition*. Upper Saddle River, NJ, USA: Prentice-Hall.

Webster. (1979). *Webster's new twentieth century dictionnary of the English language.* Unabridged. Second edition. NY, NY, USA: Simon&Schuster.

9. *Appendices*

The questions that are written within parenthesis are follow-up questions that I had prepared to use if needed.

Outline for interview 1

Test recorder

Introduction
(Feel free to say anything, not being capable of giving an answer is OK, do not wait for questions - just say what you have in mind, nothing will reach anyone outside the research team).

About yourself and your group
For the recorder, what is your name?
In which group are you
Who are the other members in your group?
Who is group leader? How did he/she become the leader?
What would you say is the function of the group leader?
Do you agree on this? (the others in the group?)
How have you divided the job between yourselves?
(How did this happen? How do you think the others regard this?)
Do you have a clear task? What is your task, as you see things?

About the project
Which code have you selected?
Tell me about this code?
(Why did you choose this particular code?)
(Which strengths and weaknesses does it have?)
How did you make the choice?
(Personal opinions? Criteria based on the code? Group leader?)
How do you want to develop this code?
(Why do you want to develop these particular issues?)
How did you take this decision?
What expectations do you have on your solution?
(Have you talked about possible problems? Or brilliant solutions?)

About knowledge of the subject area and computer networks
Would you say that you in the group, together, know enough computer science to solve the problem?
(If not, what is missing?)
(If anything is missing, how do you plan to go about to learn this?)
(Do you think you have enough practical experience in the group?)
Who knows what? How does this influence your collaboration?
(What re your strengths and weaknesses?)
Where can you contribute yourself?
Are you good at computer networks? (In theory? Hands-on?)

I am going to ask you to tell me about some concepts. Talk freely, I am not looking of anything in particular.
What is RMI?
What is UDP?
What is TCP?
What is Client/Server?
What is sockets?

Some of these you will use in your project. Tell me which ones?
(Where?)
(Why?)

Communication between humans
How would you say that the communication works
with the others in Sweden?
with the rest of the groups in the US?
with other groups?
with teachers and support?

About this way to work
How would you say it works?
Do you have any worries about problems that might arise?
(Technical?)
(Course related? (Milestones, assessment rules)
(Different ways of working? Language etc?
(What is your role in all this?)

Sum up
What would you say you would learn on this?
(In Computer Science?)

Outline for interview 2

Introduction
(as last time: say what you want, not being capable of answering is fine, do not wait for questions - just talk, noting will be forwarded to the teachers)

About you
For the tape, what's your name?
To which group do you belong?

About the project
Which code did you choose?
How have you developed the code?
How do you feel about the result?
What do you think the others think?
Does it meet your expectations?
Which problems and good experiences have you had with the collaboration?
(Different cultures/languages? course rules, like Milestones or meeting rules? different grading systems? different teachers?)
Which problems and good experiences have you had with the communication issues?
Which problems and good experiences have you had with the technical/CS issues?

Your part of the project
Which were your tasks, as you see it?
How would you judge your result?
Did your own work meet your expectations?
How have you distributed the work between yourselves?
(How did this happen?
What do you think the others would say?
How well does this go with your planning?
What became as you had planned?
What became different?
Why?)

Communication between humans
(You said earlier ....

Can we go deeper into ...)

How would you say that the communication has been
with the others in the group in the US?
with the others in the group in Sweden?
other groups?
teachers and support team?
How does this fit your expectations? Problems? Good points?

About learning
About what you have learned?
What would you say you have learned from all this (in CS?)
(in computer networks)
(about communication between humans)
(about projects)

About knowledge of the subject area and computer networks
I will ask you about some concepts, related to computer networks.
Please talk freely - there is not anything special I am looking for.
(have you used this concept? How?)
What is RMI?
(How has it worked out for you?
Have you had problems?
Have you used the registry? What is it ?
Have you had security problems? What is security manager?
Tell me more
How come these kind of problems appear?
How did you work to solve them?)
What is UDP?
(Why did you use this protocol?
Have you had any problems?
Why did they appear?
How did you work to solve them?)
What is TCP?
(Why did you use this protocol?
Have you had any problems?
Why did they appear?
How did you work to solve them?)
What is client/server programming?
What characterise a client? a server?
What is sockets?
What is ports? (Is the concept the same thing as serial parallel
ports?)
(What is the relation between these?
Have you used them?)

Round up
What would you say are the advantages and disadvantages to work in
this way?
(What has felt "good"? What has felt "wrong"?)
What would you say as a conclusion that this has meant for you?
CS knowledge?
Technical?
On a human level?