Network Science in Your Pocket

Toshihiro Tanizawa

1 Introduction

Network science has its root in mathematical graph theory, which began from the famous problem "Seven Bridges of Königsberg," solved by a mathematician, Leonhard Euler, in 1736. Erdös and Rényi combined classical graph theory with modern probability theory and initiated random graph theory in 1959 [1]. Starting in 1998, it became evident that mathematical graph theory could be applied for understanding real-world phenomena, to develop into "network science" in the present context [2, 3]. Network science has now become one of the most active research fields, and a vast amount of knowledge has been accumulated.

Any system that can be mapped onto a structure made from nodes and links is considered to be a network. This ultimate simplicity is the most valuable asset for the concept of "networks" since, in this sense, networks are everywhere. By representing a complicated system as a network, we can analyze the problems relating to the system theoretically and/or numerically. The results are compared to the observed data generated by the system, and the validity of the network representation of the system is verified quantitatively. The methods in network science have thus become powerful tools for understanding various phenomena produced by complex systems.

One of the remarkable features of network science is that, though the cutting-edge results are obtained by highly mathematical and statistical analysis, the most basic concepts, such as nodes and links, degree of nodes, average path length, clustering, and small-world properties, are conceivable even for elementary school children through vivid network visualization without any mathematical expressions. Thus, we believe that the basic knowledge of network science can be included in

T. Tanizawa (✉)
National Institute of Technology, Kochi College, Kōchi, Japan

C. B. Cramer et al. (eds.), *Network Science In Education*,
https://doi.org/10.1007/978-3-319-77237-0_12

core curricula at every educational level, from preschool to undergraduate. The concept of "network literacy"[1] embodies this situation [4].

Since most of the experts in network science at present are confined in research universities, institutes, or corporation laboratories, increasing the number of people familiar with the concepts of network science outside academia is the first important step toward the goal of network literacy. In this regard, outreach education activities for K-12 students and/or their teachers led by network scientists are crucial for spreading the concept of network literacy. Moreover, it is preferable that the activities be performed at the places where the teachers are engaged in their everyday teaching, since, by doing so, teachers could realize that network science is a subject that can be taught outside research settings.

However, preparing for outreach activities outside campus can be tedious and troublesome. In the first place, instructors have to prepare various kinds of lecture materials, such as viewgraphs, images, and charts. Since the networks dealt with in network science are dynamic objects on which various phenomena occur, the presentation of the contents should also be dynamic and, preferably, interactive. If the attendees have sufficient skills for software programming, it would be a good option that the lectures include a short hands-on course for developing small applications of network science using various programming languages, such as C, C++, Java, Python, and so on. The instructors, however, would not be able to expect such programming environments at off-campus activities. Connection to the Internet is also preferable, since students may look up various technical terms online during the lectures. The instructors may also use online contents during the lectures. To deal with all of these problems coherently would be a burden for the instructors in preparation of the lectures.

To make this easier, it would be very helpful if instructors have the necessary tools and materials on a stand-alone environment in one tiny box that can be brought anywhere easily, which has become possible through recent advancements in tiny single-board computer technology. The project, "Network Science in Your Pocket" (the Pocket project, hereafter), intends to realize such a "tool box" that enables us to practice outreach activities for network literacy in everyday teaching in the classroom.

This chapter is organized as follows. In the next section, we clarify the requirements for such a toolbox, which is called the Pocket Server in this chapter. Section 3 describes two operation modes of the Pocket Server. In Sect. 4, concrete implementation processes for building the Pocket Server are described. Section 5 reports the details of a lecture for network science using the Pocket Server, held in a technical college in Japan. Section 6 summarized this chapter.

[1] A PDF of Network Literacy: Essential Concepts and Core Ideas, in the original English as well as over 20 additional translations, can be downloaded from: https://sites.google.com/a/binghamton.edu/netscied/teaching-learning/network-concepts/.

2 Requirements for the "Tool Box"

In Fig. 1, we show the education levels, which the Pocket project intends to cover. In preparation for a series of lectures in an outreach activity, instructors have to take into account prior knowledge and skills of the students. For instance, instructors would not be able to expect primary school students to have sufficient skill in fluent keyboard typing. The contents of the lectures would thus be preferable if they are presented with a lot of graphics on-screen and proceed by mouse clicking and/or dragging. The most convenient interface for such a style will be a web browser, so we refer to this style as the "browser-oriented" style. As the education level becomes higher, the typing skill of the students improves. Instructors are even allowed to expect that the students might have some knowledge or experiences in software programming. In this case, the contents of the lectures are allowed to contain more materials that require inputs from the students through a character-based user interface, such as a terminal window provided in almost all operation systems presently available. We refer to this style as the "command-line oriented" style. The toolbox of the Pocket project is thus required to be able to deal with both browser-oriented and command-line oriented styles.

To make lectures interactive, the toolbox should accept various types of communication channels and protocols over the local area network (LAN) and the Internet alike. For effective storage of a large amount of data used in the course, a suitable database application might also be required. If a set of lectures for high school or undergraduate students contains assignments of writing progress reports, it might be preferable that the tool box provides a suitable document processing suite such as LaTeX that can handle mathematical expressions, reference citations, indexing, and so on, easily and coherently. For efficient code writing in programming, good old editors such as Vim or Emacs should not to be omitted from a fundamental set of requirements for the toolbox.

Figure 2 is a schematic picture of the toolbox of the Pocket project, the "Pocket Server" that fulfills all of these requirements. In this picture, the Pocket Server is intended to work in two different operation modes alike; one acts as a server that

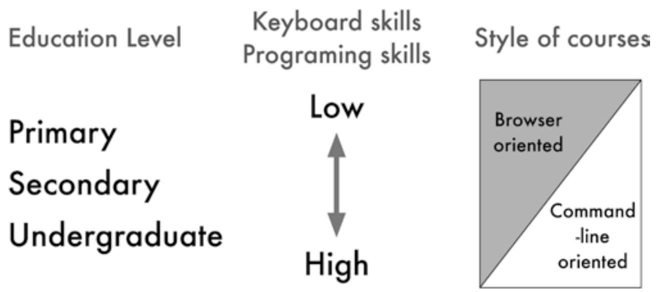

Fig. 1 Education levels from primary to undergraduate over which this Pocket project intends to cover

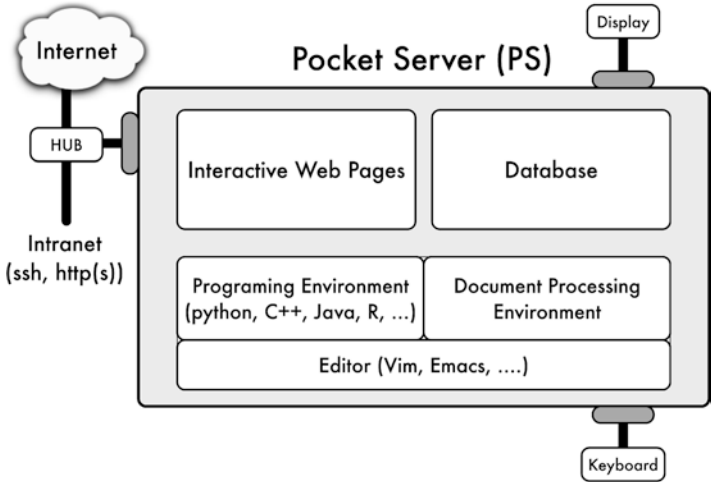

Fig. 2 A schematic picture of a sample structure of the "Pocket Server" (PS)

accepts connection requests from outside and delivers the contents of the course, and the other acts as a client computer for each student in which all necessary tools for studying network science are included in a box.

3 Two Operation Modes of the Pocket Server

As mentioned above, the Pocket Server has two operation modes, the server mode and the client mode. The server mode is the basic mode of the Pocket Server, in which a single Pocket Server works as an ordinary http server. This mode is suitable for giving a browser-oriented course in a lecture room that already provides a set of personal computers for students with a working local area network (LAN). These days, many primary education facilities as well as those for secondary or higher education are equipped with such lecture rooms. In this case, the instructor needs to only bring the Pocket Server and connect it to the screen and LAN of the lecture room. The students access and retrieve the contents of the lectures in the Pocket Server through http connection over LAN. If the server accepts wireless connections, personal computers are not necessary. The attendees can access the contents with their own laptops, tablets, or smartphones. In Fig. 3, we show a schematic picture of the use of the server mode. In case the number of the computers that connect to a single Pocket Server is large and exceeds the capability for handling all requests from the client machines in the lecture room, increasing the number of Pocket Servers for distributed request processing could be an effective treatment.

If the organizer of the lectures can afford to provide a sufficient number of Pocket Servers, each Pocket Server can be operated in the client mode as a stand-alone PC

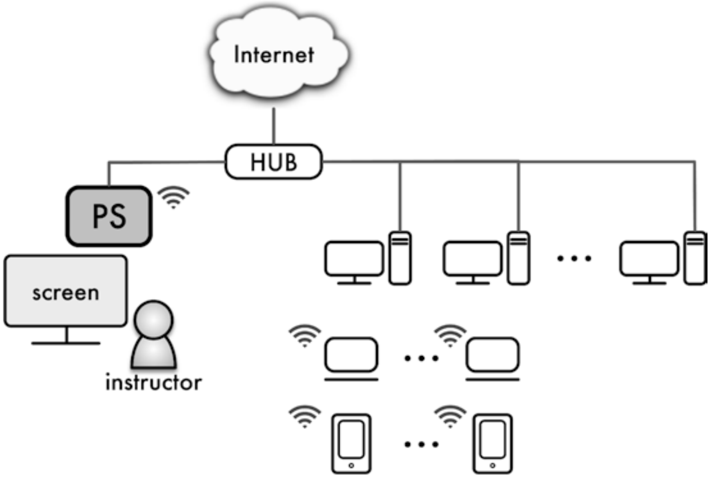

Fig. 3 The use of the server mode

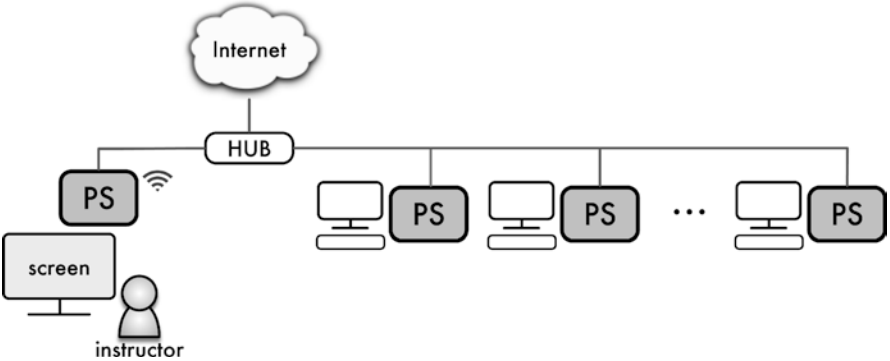

Fig. 4 The use of the client mode

by connecting it to a monitor and a keyboard. (See Fig. 4.) In this case, each student has the same environment as that of the instructor at the start of the session. The students are first encouraged to try the examples presented by the instructor, typing the inputs with their own hands, and see the results. By modifying these examples, the students can start to explore the various topics in network science by themselves. The client mode is thus suitable for command-line oriented lectures. It should be noted, however, that even when most of the contents of the course takes the command-line oriented style, it is always possible for each client to connect to the Pocket Server of the instructor separately. The lectures can thus shift back and forth freely between the server mode and the client mode.

4 Implementation of the Pocket Server

As stated above, the requirements of the Pocket Server are wide-ranging. However, present technology enables us to fulfill all of these requirements in a tiny single-board computer. As a concrete example, we present a sample implementation built on Raspberry Pi 3 Model B.[2] Figure 5 is a real image of this device, and Table 1 lists the specifications of Raspberry Pi 3 Model B (RP3). On this RP3, all requirements of the Pocket Server are realized with various open source software suites. The following are brief construction steps of the Pocket Server on RP3:

1. Format a new microSDHC and put it into the slot of the RP3.

Fig. 5 Raspberry Pi 3 Model B

Table 1 Specifications of Raspberry Pi 3 Model B. ARMv8-A (64/32-bit)

Architecture	ARMv8-A (64/32-bit)
SoC (system on chip)	Broadcom BCM2837
CPU	1.2 GHz 64-bit quad-core ARM cortex-A53
GPU	Broadcom VideoCore IV
RAM	1GB LPDDR2 (900 MHz)
On-board network	10/100Mbit/s Ethernet, 802.11n wireless, Bluetooth 4.1
On-board storage	microSDHC 16GB
Size	85.6 mm × 56.5 mm × 17 mm
Weight	45 g
Ports	HDMI, 4 × USB 2.0, Ethernet, and others

[2] Raspberry Pi 3 Model B is a product developed by the Raspberry Pi Foundation. https://www.raspberrypi.org/products/raspberry-pi-3-model-b/.

2. Install an operating system with NOOBS, which is an official operating system installer.[3] Here we choose the operation system, Raspbian, a Debian-based Linux operating system.[4]
3. Configure network interfaces appropriately.[5]
4. Install necessary software bundle, such as software development environments (C/C++, Java, Python, R, etc.), editors (Vim, Emacs, etc.), libraries, database applications (MySQL, postgreSQL, etc.), graphics-handling applications (ImageMagick, GIMP, etc.), applications for graph plotting (gnuplot, etc.), and text-processing applications (LaTeX, etc.), with the default package manager of Raspbian.[6]
5. Install other network handling applications (Cytoscape,[7] Gephi,[8] etc.).

The sample implementation in this section is mainly built by Python. For scientific computation required for network science, a lot of modules, such as NumPy, SciPy, matplotlib, NetworkX, pandas, and so on, are necessary. Though we can install these modules one by one with any of standard package managers of Python (e.g., by pip), here we use Berryconda,[9] which is an integrated Python distribution for scientific computing adapted for Raspbian based on the Conda package managing system, which includes all necessary modules.

The web application that plays a central role in presenting the contents of the course in the browser-oriented style is constructed on top of the Python module, Flask.[10] This module contains a tiny web server application of its own, which is sufficiently handy for present educational purposes. We do not need a gigantic server application such as Apache for this kind of job. Figure 6 is a screenshot of the entry page of the sample web application for this project. The web application can be made interactive on the server side implemented by Python, as well as on the client side using JavaScript. By using an appropriate database application, documents, graphics, and data necessary for the course are conveniently stored and retrieved from the web application.

Since the Pocket Server built on RP3 is a tiny but self-contained complete Linux server, the students can login to the server from their client machines and use the resources on the Pocket Server even in the server mode operation. The client mode operation is simply to use each Pocket Server as an independent Linux client PC.

---

[3] For NOOBS, see the web page at https://www.raspberrypi.org/downloads/noobs/.

[4] For Raspbian OS, see the web page at https://www.raspberrypi.org/downloads/raspbian/.

[5] Basic guides to configuring RP3 can be found in the web page at https://www.raspberrypi.org/documentation/configuration/.

[6] The Advanced Packaging Tool (APT) for Debian-based OS is also available in Raspbian. See the web page at https://www.raspberrypi.org/documentation/linux/software/apt.md.

[7] For Cytoscape, see the web page at http://www.cytoscape.org/.

[8] For Gephi, see the web page at https://gephi.org/.

[9] Berryconda is developed on GitHub at https://github.com/jjhelmus/berryconda.

[10] The main web page of Flask is at http://flask.pocoo.org/.

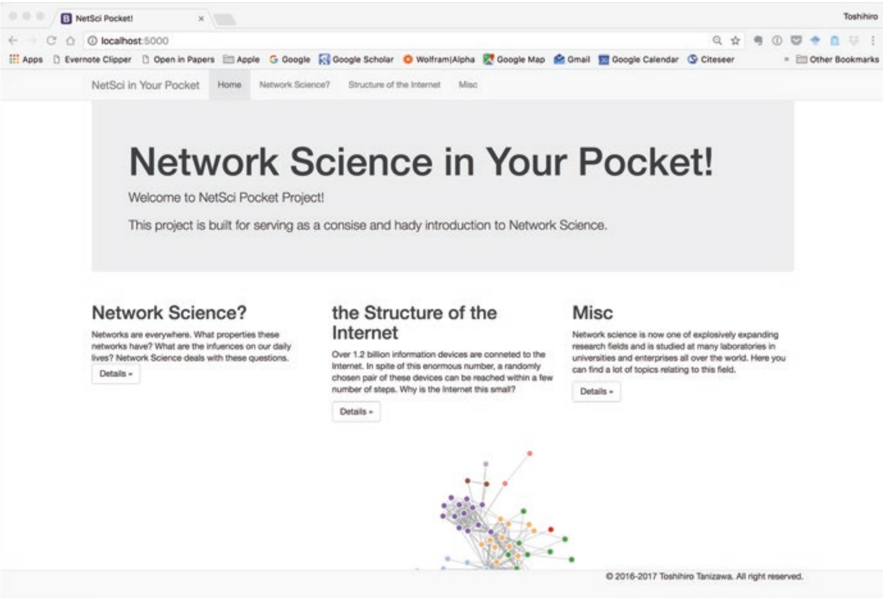

Fig. 6 Screenshot of the entry page of the web application "Network Science in Your Pocket!"

In a session of lectures, Pocket Servers are distributed to the students, and the students use those Pocket Servers as their client machines, in which a complete environment necessary for scientific computing in network science is provided.

5 Case Study of a Network Science Lecture with the Pocket Server

In this section, we briefly report on a network science lecture using the Pocket Server described in this chapter. The lecture was one of several open-house outreach activities at a technical college in Kochi, Japan, held in August 2016 for students from 10 to 15 years old, to promote their interest in science. The room for the lecture is equipped with 40 client personal computers with a projector and a large screen for the instructor. A private local area network connected to the Internet connection over a secure gateway is also available. Since we could not expect the participants to have prior knowledge of network science and sufficient skills for scientific computing, the lecture was planned in the browser-oriented style with three Pocket Servers of identical contents, operated in the server mode for dispersion of access requests from 40 client machines. Three fixed and different private IP addresses were assigned to these servers. Though the number of actual students who attended the course was about ten, the students were divided into three groups according to the number of the Pocket Servers and guided to connect to the server assigned to each group at the start of the lecture.

The contents of the lecture were presented basically by clicking the buttons of the web pages.[11] See Fig. 6 again as the entry web page for this lecture, though it should be noted that the actual web pages used in the lecture were localized versions in Japanese.

The aim of the lecture was to show the small-world property of scale-free networks due to the existence of hubs, by comparing the path length between two randomly selected nodes with that of two-dimensional regular lattices. The key question in this lecture was why the Internet, which consists of several billion of nodes, is so "small", in the sense that we feel almost no frustration in moving page to page by following the links on pages.

Next the instructor showed networks in the real world by showing several images such as the route map of the subway system in Tokyo, a road map of the city of London, a network of coauthorship of scientific papers, a network of the linkage connections between web pages on the Internet, the neuronal network of the human brain, Zachary's Karate Club network, a network formed by related English words, and so on.

After grasping the overall concept of networks, the students were led to lecture pages to learn the concepts of nodes, links, degrees, degree distribution, and path length between nodes. Interactive web pages written in JavaScript were also provided when they were considered to be effective. Figure 7 is an example of this kind

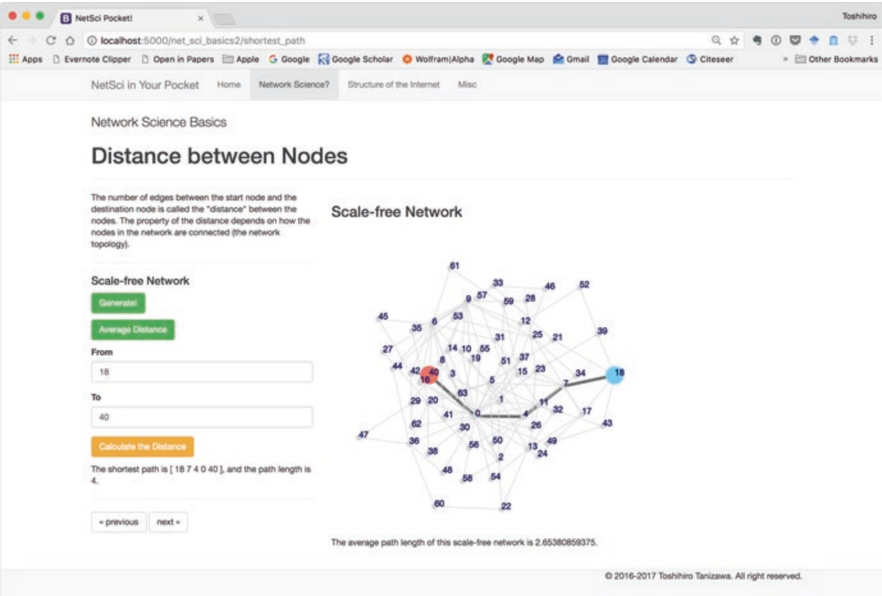

Fig. 7 An example of an interactive web page for the calculation of node distance

[11] The whole web application implemented by python-flask module can be downloaded from the author's GitHub page at https://github.com/toshitanizawa/NetSci-Pocket.

of interactive web page. According to the inputs from the students, the path length between the nodes are calculated and displayed dynamically. Finally, the students learned that the path length in scale-free networks does not increase proportionally to some power of the system size but logarithmically, which is the main reason for the smallness of the Internet.

Though all the students attending this lecture had no prior knowledge of network science, the survey after the lecture told us that they enjoyed learning various characteristics of networks, such as the universality of power-law behavior in the degree distribution or the small-world property. Above all, they were really surprised at the astonishingly small values of the path length in scale-free networks, knowing that they could not find two nodes whose path length was larger than 10, even in a huge network such as a Barabasi-Albert network of about 65,000 nodes.

6 Summary and Discussion

In this chapter, we describe building a small portable server that provides sufficient facilities for delivering lecture courses on network science, even if they are held off-campus. Thanks to current technology, it is possible to build such a server on a single-board computer so small that you can carry it in your pocket. A sample structure, which consists of two operating modes, the server mode and the client mode, was proposed, and an example of implementation of the server built on a single-board computer available on the market was also described. The effectiveness and capability of this server was actually tested at an open-house outreach activity at a technical college in Japan.

In this chapter the author reported a case study for using the Pocket Server in the browser mode. The browser mode is suitable for presenting ready-made contents that are well planned prior to the lectures. In contrast, the client mode is suitable for exploring or developing the style of the contents. Since the Pocket Server is a tiny but complete Linux machine provided with the necessary tools for scientific calculations and numerical simulations, students with sufficient skills can readily start a small research project on network science. The client mode operation of Pocket Servers could, therefore, be effective in an advanced or intensive program for network science targeted to highly motivated students in secondary schools. After instructions from the lecturer about the basic concepts of network science, the students would be introduced to selected topics of interest about network science and could conduct research on their own. You do not have to limit the use of the Pocket Server only to short lectures for outreach activities; an entire course on scientific computing including network science could be taught with Pocket Servers distributed to students.

The biggest challenge at present for the Pocket project is to provide sufficient lecture materials that are available to the teachers at any educational levels. Building a system for compiling the contents of outreach activities performed by members of NetSci community is crucial.

References

1. Erdös, P. and Rényi, A. (1959) On Random Graphs. I. Publicationes Mathematicae, 6. 290–297
2. Albert, R. and Barabási, A-L. (2002) Statistical mechanics of complex networks. Reviews of Modern Physics, 74:47–97.
3. Newman, M. (2003) The structure and function of complex networks. SIAM Review, 45:167–256, 2003.
4. Sayama, H., Cramer, C., Porter, M., Sheetz, L., and Uzzo, S. (2016) What are essential concepts about networks? Journal of Complex Networks, 4(3). 457–474.