COMPUTING AND COMMUNICATIONS

VOLUME 04  ISSUE 01 & 02  MARCH & JUNE 2020

A QUARTERLY PUBLICATION OF THE ADVANCED COMPUTING AND COMMUNICATIONS SOCIETY

32 COMPUTER VISION WITH DEEP LEARNING FOR PLANT PHENOTYPING IN AGRICULTURE: A SURVEY

51 POST – QUANTUM CRYPTOGRAPHY

70 EXPERIENTIAL LEARNING OF NETWORKING TECHNOLOGIES: Understanding Network Layer & IP Addressing

OMNIBUS ISSUE

A COMPREHENSIVE SURVEY ON RECENT ADVANCEMENTS IN COGNITIVE RADIO-BASED INTERNET OF THINGS

A QUARTERLY PUBLICATION OF THE ADVANCED COMPUTING AND COMMUNICATIONS SOCIETY

₹ 100
Advanced Computing and Communications Society (ACCS) is launching India’s first free OPen Access digital Library (OPAL) for discovering technical content published in India by recognized technical and scientific conferences and journals.

OPAL provides a robust indexing platform open to all ACCS members (both individuals and corporate members). Members wishing to use OPAL services should register with the platform. Registration is free and requires only minimal documentation to authenticate the authors and/or organizations. OPAL does not own or claim copyright to materials indexed by it or hosted by it, unless explicitly specified.

Salient Features

1. Search: Search based on the metadata of the technical content
2. Download papers and articles: Download technical content in PDF format
3. Upload: Upload of technical content by authenticated users in a specified format and metadata
4. Diverse content: Text documents, Documents with scientific/mathematical notations, Datasets etc.
5. Omni-Access: Easily accessible on all popular form factors of screens (Desktop/Mobile/Tablet...)
6. API made available: The key features of the OPAL will be exposed to users via an API

Advance your career
Apply Online Today!

Become an Institutional Member of ACCS and start a local ACCS Chapter in your Institute!
An Institutional membership is truly an investment in the future of our engineers. The membership also increases your institution’s visibility in the scientific community and encourages collaborative research.

For details visit www.accsindia.org/membership

Advanced Computing and Communications Society
Gate No. 2, C. V. Raman Avenue
Indian Institute of Science
Bengaluru - 560012, Karnataka, INDIA

Telephone: +91 80 2360 7142
Mobile: +91 76249 57142
Email: info@accsindia.org
Website: http://www.accsindia.org
08  A TRIBUTE TO SATISH DHAWAN
RAJENDRA K. BERA

COMPUTER VISION WITH DEEP LEARNING FOR PLANT PHENOTYPING IN AGRICULTURE: A SURVEY
VINEETH N BALASUBRAMANIAN, WEI GUO; AKSHAY L CHANDRA, SAI VIKAS DESAI

56  INTEGRATING PHYSICAL, CYBER AND THE BIOLOGICAL WORLDS
AN INTERVIEW WITH PROF. JAN RABAЕY, PROFESSOR, EECS, BERKELEY UNIVERSITY

59  EXPERIENTIAL LEARNING OF NETWORKING TECHNOLOGIES:
EXPERIENTIAL LEARNING OF NETWORKING TECHNOLOGIES:
Evolution of Socket Programming – Part II
RAM P. RUSTAGI, VIRAJ KUMAR

POST – QUANTUM CRYPTOGRAPHY
P.V. ANANDA MOHAN

A COMPREHENSIVE SURVEY ON RECENT ADVANCEMENTS IN COGNITIVE RADIO-BASED INTERNET OF THINGS
RAJALEKSHMI KISHORE AND SANJEEV GURUGOPINATH
In this omnibus issue which has been brought amidst disruption of normal activities in our physical world due to COVID19 pandemic, several articles of significant interest have been included.

In their article titled “A Comprehensive Survey on Recent Advancements in Cognitive Radio-based Internet of Things”, the authors survey two promising technologies, viz. Internet of Things (IoT) and Cognitive Radios (CR) which are though not fully mature have nevertheless potential, abundant, and practical applications that impact our present day lives significantly in not so a distant future. While the concept of Internet of Things (IoT) has been presented and discussed at length in a dedicated article of our earlier issue, the Cognitive Radio (CR) technology is being introduced in ACC through this article.

The authors survey the progress made to date and state of art of these two technologies before discussing how the combination of two technologies referred to as CR based IoT (CR-IoT) can be of use in several application domains. The main motivation for CR-IoT stems from bandwidth allocation to be made for IoT devices. The number of IoT devices is expected to grow in large numbers, and it will be very difficult to allocate spectrum bands to large number of IoT devices.

In another interesting paper, the authors discuss the science of Plant Phenotyping. It is the science of the characterization of the crops which is particularly important for decision support in agriculture. This aids plant breeders when selecting the best genotypes that will become the future cultivars well-adapted to different environments. As such, plant phenotyping helps to better understand the functioning of the crops. The knowledge gained is often used to calibrate crop models.

High throughput plant phenotyping methods have shown great promise in efficiently monitoring crops for plant breeding and agricultural crop management. Research in deep learning has accelerated the progress in plant phenotyping research. With many open problems in plant phenotyping warranting further studies, it is indeed a great time to study plant phenotyping and achieve rapid progress by utilizing the advances in deep learning.

The authors discuss various issues and challenges that are to be dealt with, application of supervised learning, and open problems associated with phenotyping with limited data.

Quantum cryptography is the science of exploiting quantum mechanical properties to perform cryptographic tasks.

Post-quantum cryptography (sometimes referred to as quantum-proof, quantum-safe or quantum-resistant) refers to cryptographic algorithms (usually public-key algorithms) that are thought to be secure against an attack by a quantum computer. As of 2019, this is not true for the most popular public-key algorithms, which can be efficiently broken by a sufficiently strong quantum computer. The problem with currently popular algorithms is that their security relies on one of three hard mathematical problems: the integer factorization problem, the discrete logarithm problem or the elliptic-curve discrete logarithm problem. All of these problems can be easily solved on a sufficiently powerful quantum computer running Shor's algorithm. Even though current, publicly known, experimental quantum computers lack processing power to break any real cryptographic algorithm, many cryptographers are designing new algorithms to prepare for a time when quantum computing
becomes a threat.

Currently post-quantum cryptography research is mostly focused on six different approaches: Lattice-based cryptography, Multivariate cryptography, Hash-based cryptography, Code-based cryptography, Supersingular elliptic curve and Isogeny cryptography.

In cryptography research, it is desirable to prove the equivalence of a cryptographic algorithm and a known hard mathematical problem. These proofs are often called “security reductions”, and are used to demonstrate the difficulty of cracking the encryption algorithm. In other words, the security of a given cryptographic algorithm is reduced to the security of a known hard problem. Researchers are actively looking for security reductions in the prospects for post quantum cryptography.

In his paper titled “A primer on Post Quantum Cryptography”, the author dwells on the current scenario in the research area of Post Quantum Cryptography and focuses on two developments made to date in the area viz. Ring Learning With Errors (R-LEW) and New Hope Key Encapsulation Mechanism (KEM).

In continuation with the series of articles, we bring you two installments of Experiential Learning of networking technologies. In the first one, the authors discuss at length on the topic of Evolution of socket programming. The second one, ‘Understanding Network Layer & IP Addressing’, delves into IP addressing and routing. As usual, the exercises at the end of the articles allows the reader to learn these concepts hands-on.

While wishing happy reading to our readers, we also wish them safe and virus-free good health in these times when the foot-print of COVID19 pandemic is growing at a fast rate.

Dr. N Rama Murthy
Editor
The current pandemic has thrust to focus several things that were in the side lines of mainstream technology and waiting to make a breakthrough. One such concept is Cognitive Radio-based IoT. While IoT has made its mark on the technology stage, there is a new thought emerging that a cognition driven radio system is essential in responding to real time scenarios like the present one. We have a survey paper presented by Prof. Sanjeev Gurugopinath et.al. which dives deep into Cognitive Radio-based IoT systems and discusses the various aspects of it from technical challenges to standardization of protocols.

Our Experiential Learning series is getting into the realm of IP Routing and DNS giving a hands-on insight into how DNS and IP routing works behind the scene. The exercises provided allow the student to work at her or his home network (in these days) and learn the basics. We encourage you to follow Prof. Ram P. Rustagi through his tutorial style writing and enrich your learning.

These days are extremely challenging for all of us and I would like to take this opportunity to commend the scientific society for rising to the occasion and fighting the pandemic in its own ways. If the medical fraternity is in the forefront, the computing and communication engineers and professionals both in the industry and academia have ensured that people are connected despite maintaining “social distancing” which is commendable.

I am sure all of you are utilizing this rare opportunity to get yourself in the forefront of doing things that will enrich the field of science, especially the advanced computing and communications.

We at the Advanced Computing and Communications Society have ensured that our journal is published despite challenges and we thank all our contributing authors, editors and people behind the scenes for getting the issue out.

Stay safe and contribute to the advancement of your field.

Dr. Saragur M. Srinidhi
Editor
The Advanced Computing and Communications Society (ACCS) is a registered scientific society founded to provide a forum for individuals, institutions and industry to encourage the promotion of advanced Computing and Communications technologies. ACCS is a Sister Society of the IEEE Computer Society. ACCS is India’s first technical society dedicated to Computing and Communication engineers. We need you to become part of ACCS’s mission to advance the state of computer and communication engineering and technological leadership.

ACCS membership is truly an investment in the future of our engineers and scientists. The membership reflects your active involvement with the cutting edge research in the areas of advanced computing, communications and the related multi-disciplinary fields in electrical sciences. It provides endless opportunities for academic outreach and helps institutions and individuals work towards elevating the standards of research and teaching.

Your membership helps us accomplish several goals:

- Encourage development and dissemination of domain knowledge through its meetings, publications and committee outreach activities.
- Offer public lectures and talks from distinguished speakers on most current topics of interest.
- Provide a voice in legislative policies affecting the industry.
- Provide travel awards to students and scientists around the world.

As an ACCS member, you instantly have:

- Access to latest technical information through workshops & conferences.
- Opportunity to network with industry leaders.
- Engage with leading academic institutions.
- Connections throughout the global marketplace.
- Recognition for your accomplishments by your peer.
- Member-only savings on ACCS programs and services.
- Locate career opportunities with our member base.
- Get to recognize your peers in their achievements.
- ACCS members will receive a member@acccsindia.org email address with a feature rich service for you to create Special Interest Groups across the member base in your area of interest and specialization.

We encourage you to become Life members of ACCS that will provide you access to the industry’s most essential technical information, networking opportunities, career development tools, and many other exclusive benefits.

Please visit the ACCS portal at www.acccsindia.org for additional information on the society.

✉️ secretary@acccsindia.org
☎️ +91 80 2360 7142

Write to us at: Secretary
Advanced Computing & Communication Society
Gate #2, C. V. Raman Avenue, Indian Institute of Science, Bangalore 560 012, India
A TRIBUTE

SATISH DHAWAN
(1920-2002)

Rajendra K. Bera
Acadinnet Education Services, Bengaluru

““The unique feature of his teaching was that he created a spirit of research and enquiry in me, taught me how to design without giving me the design.”

— APJ Abdul Kalam
(11th President of India)

On 25 September 1920, in pre-independent India, Devi Dayal and his wife were blessed with a son they named Satish, in Srinagar, Kashmir. During his lifetime Satish Dhawan would become an iconic figure in India’s higher education system and raise India’s space program to levels then considered a dream. Along the way, he guided and mentored several talented people who too served India with great distinction and honor; the most prominent among them was the self-effacing Avul Pakir Jainulabdeen Abdul Kalam (1931-2015), the Missile Man of India, a distinguished aerospace engineer, highly decorated with India’s national civilian honours (including the Bharat Ratna in 1997) for his immense contributions in building the base for India’s rocket and missile technology. He was an inspiration to civil society. Abdul Kalam eventually served as the 11th President of India from 2002 to 2007. Kalam had nothing but sheer admiration for Dhawan. He wrote, “The unique feature
In order to promote a rapid development of activities connected with the Space Science, Space Technology and Space Applications, the Government of India consider it necessary to set up an organization, free from all non-essential restrictions or needlessly inelastic rules, which will have responsibility in the entire field of Science and Technology of Outer Space and their Applications.”

of his teaching was that he created a spirit of research and enquiry in me, taught me how to design without giving me the design [1].

Satish Dhawan himself was a gifted student. This is evident from his unusual academic profile: a BA in physics and mathematics (1938), an MA in English literature (1941), and a BE (with honors) in mechanical engineering (1945) from the Punjab University, Lahore. In 1946 he traveled on a government scholarship to the United States. There, at the University of Minnesota, he earned an MS in xx in 1947, and from California Institute of Technology (Caltech) he earned the Engineer's degree in 1949 in aeronautics and a PhD in aeronautics and mathematics in 1951. His PhD advisor was the renowned professor Hans W. Liepmann who along with Anatol Roshko (also at Caltech) wrote a classic textbook, Elements of Gas dynamics, in 1957.

Dhawan’s Caltech years gave him the breadth of scholarship, depth of research and an opportunity to hone his skills in investigating problems independently and efficiently. From this point onwards his professional career would be stamped with these rare qualities from which others like Abdul Kalam would benefit in their early professional years and into the future. Dhawan’s Engineer's and PhD theses, respectively, were


Taken together the scientific output was very significant as they produced some pioneering work on boundary layer flows in fluids. Liepmann and his two students Roshko, and Dhawan worked together and demonstrated the importance of the state of the boundary layer, laminar vs. turbulent, on resulting shock-wave pattern and pressure distribution. This was Dhawan’s first participation in active research. [4] This classic work is elegantly presented in NACA Report No. 1100 published in 1952. [5] The trio worked and got along so well that it “laid the foundation” for their lasting friendship over the next half century. Following this work, Dhawan started his PhD work on the direct measurement of skin friction. [6]

In his PhD thesis, Dhawan showed great ingenuity in developing a device to measure local skin friction on a flat plate by measuring the force exerted upon a minute movable part of the plate by means of a reactance device. The device was then used to make accurate measurements in the low speed range, both for laminar and turbulent boundary layers and then in the high speed subsonic range in turbulent boundary layers.

Finally, some measurements were made in supersonic flow. Those were the days when wave drag and induced drag experienced by a body moving through a fluid were much better understood than skin friction and boundary layer whether in theory or in experiment. A much better understanding was critically needed to design high speed aircraft (e.g., fighters) and missiles. The research done was a great success and Dhawan's stature as an outstanding researcher was established. His PhD thesis (with minor editorial changes) also appeared as a NACA Report in 1953. [7]

In addition, Dhawan and Roshko collaborated on the design and construction of an ingenious flexible nozzle for conducting research in supersonic flows. At Caltech, Dhawan's entire time was spent at the Graduate Aerospace Laboratories of the California Institute of Technology (GALCIT); its first director was the Hungarian aerospace genius Theodore von Kármán (1881-1963) from 1930 to 1949.

In 1951 Dhawan returned to India and joined the Indian Institute of Science (IISc) as a scientific officer. In 1955, he became a professor and head of the Department of Aeronautical Engineering succeeding (O.G. Tietjens, 1948 – 1955) and, in 1963, was appointed director of the institute (succeeding Suri Bhagavantam, 1957-1962), a position he occupied till 1981. Throughout his career, the influence of Caltech and Liepmann is highly visible. In scope and aim, he clearly wanted to emulate GALCIT. [8]

As Liepmann wrote about his association with Dhawan: “Ever since then, we in GALCIT have had close contacts with the Indian Institute of Science, and thus a calibration station for admission, leading to some
excellent Indian graduate students at GALCIT.”

During his tenure as director of IISc, he set about rejuvenating the Institute with foresight and vision and expanded its academic scope by introducing new disciplines that included automation and control; materials science; molecular biology and biophysics; computer science; technology for rural areas; theoretical physics, applied mathematics; solid state chemistry; and atmospheric sciences, [9] while augmenting the faculty with new talent.

During 1971–72 Dhawan was a visiting professor at Caltech on a sabbatical. While there, Vikram Sarabhai, highly regarded as the Father of the India’s space program, unexpectedly passed away on 30 December 1971, creating a sudden leadership vacuum in the nascent program. The then prime minister Mrs. Indira Gandhi promptly and urgently summoned Dhawan to return to India and take charge of the program. Dhawan agreed but only after some pragmatic conditions articulated by him were conceded. They included...

...his views about the space program that India should pursue, its administrative structure, and the need to keep it away from Delhi. If these were accepted he would be honored to lead the program. The government agreed, and a new structure was set up involving a policymaking Space Commission, an administrative arm of the government called the Department of Space. And a science and technology agency called the Indian Space Research Organization (ISRO) – all three headed by one person.” [10]

Thus began a parallel career track for Dhawan who continued to be the director of IISc at Bangalore. He convinced the government that this arrangement would bring synergy to the space program as it would establish the necessary close ties between the IISc and ISRO in developing space technology. This was clearly a variation of the GALCIT model he had seen work remarkably well at Caltech. He was persuasive enough. On a token salary of Rs. 1 per month, he became the chairman of Indian Space Research Organization (ISRO) and the chairman of the Indian Space Commission.

Dhawan's strategy for organizing India's space program carried great foresight. This is reflected in the resolution setting up the Space Commission and the Department of Space. It read:

"In order to promote a rapid development of activities connected with the Space Science, Space Technology and Space Applications, the Government of India consider it necessary to set up an organization, free from all non-essential restrictions or needlessly inelastic rules, which will have responsibility in the entire field of Science and Technology of Outer Space and their Applications." [11]

The Space Commission was established with full executive and financial powers modeled on the lines of India's Atomic Energy Commission. Dhawan well knew the exasperating and dampering effect a scientifically uninformed government bureaucracy has on the development of forward looking science and technology, and the herculean efforts needed to counter it. I got a sardonic expression of it from him in a chance encounter when I made a casual remark about the Light Combat Aircraft (LCA) project India had just initiated, while watching a hang gliding demonstration in Bangalore (now Bengaluru) sometime in the 1980s.

Dhawan wanted the space program to have a sustainable national relevance with close interfaces with end users. One of his primary aim was to make space technology work for the development of Indian society, starting from the grassroots. He constantly worried about the lower strata of society and how they could be uplifted with a humane protective cover provided by space technology—the unemployed, unskilled, uneducated—without being uprooted, displaced, and dispossessed of their pristine cultural heritage. He had a deep sense of moral order and social justice.

Among his scientific and technical contribution to the space program was the final configuration—solid-liquid-solid-liquid—of India's Polar Satellite Launch Vehicle (PSLV), the dependable workhorse of ISRO.

Dhawan based the configuration on a mix of factors that included available technical expertise, infrastructure, launch schedule and funds.

Dhawan’s list of accolades include Distinguished Alumni Award in 1969 from Caltech, and an IISc Distinguished Alumni Award in 1981. He was elected to the Indian Academy of Sciences in 1970 and served as its president during 1977-1979. In 1978 he was elected to both the US National Academy of Engineering, and the Indian National Science Academy.

He became a foreign honorary member of the American Academy of Arts and Sciences in 1972. India honored him with the Padma Bhushan (1971), Padma Vibhushan (1981), the Indira Gandhi Award for National Integration (1999), acknowledging him as “one of our foremost scientists, teachers, and national builders and [who] is deeply concerned with the solution of national problems through the use of science”.

He received several honorary doctorates, including one from Cranfield Institute of Technology, UK. Following his death, the satellite launch centre at Sriharikota, Andhra Pradesh was renamed as the Satish Dhawan Space Centre.

Remarks by Hans Liepmann
Here are some views expressed by his mentor Professor Liepmann. [12]

“Satish was immediately accepted and respected by this highly competent and proud group of young scientists. He showed an unusual maturity in judging both scientific and human problems, a characteristic that today is called “leadership quality.””

“Satish could be tough without having to get mad first—a trait that I envy”

“Many years ago Satish told me that accurate weather prediction could improve India’s economy decisively. With the flock of satellites he helped organize, Satish did indeed do something about the weather. Now future geophysical satellites will be launched from the Satish Dhawan Space Center, named in his honor last September.”

All those who have come in touch with Dhawan have expressed similar views and sentiments. Krishnaswamy Kasturirangan (former chairman of ISRO, 1994-2003) summed up nicely the human and humane persona of Dhawan, “a rare human being of his kind – a lovable teacher, an intense researcher, an innovative technologist, an able institution builder and an excellent academic administrator, all at the same time but distinctly visible.” [13] His MA in English literature strikingly stands out in his academic record; it gave him an unusual edge in communicating with the high and mighty and added wit and luster to his personality that never failed to charm people.

My brief recollections
During my career at the National Aerospace Laboratories (NAL) at Bangalore (1971-1995), Dhawan was Chairman, Research Council of NAL (1984–93), I have exchanged perhaps a-dozen sentences with him on various occasions in informal surroundings but had enough opportunity to listen to him between those sentences. I first heard of his academic background as presented here from him in addition to snippets of his younger days at a dinner party in Bangalore sometime in the 1980s. A few of us had gathered around him and we heard him in fascination as he recounted those days. We gasped when he mentioned studying for his MA in English. It was so unexpected! On another occasion I felt that he had some faint idea about my research work, more specifically a research paper of mine that Sir James Lighthill, FRS and Lucasian Professor of Mathematics (1969-1979, succeeding Paul Dirac), had favourably commented on in a letter to me in 1991 and which I had shared with some colleagues and friends. My greatest impression about Dhawan was formed when I met Professor Liepmann in his office at Caltech in early 1980. I curiously asked him how he chose students from India. His response was remarkably simple. He said he knew very little about the Indian education system and therefore had no particular method. But if “Satish makes a recommendation” we don’t ask questions; we simply accept the person! It was therefore a great pleasure for me to learn that Dhawan was posthumously admitted

Liepmann (2002)
Kasturirangan (2003)

“a rare human being of his kind – a lovable teacher, an intense researcher, an innovative technologist, an able institution builder and an excellent academic administrator, all at the same time but distinctly visible.”

— Krishnaswamy Kasturirangan
(former chairman of ISRO, 1994-2003)
to GALCIT’s Hall of Fame in 2018. I was also very disappointed the day I met Liepmann in not being able to meet Professor Richard Feynman who was indisposed that day on account of his cancer. I consider Feynman as my only teacher from whom I learnt physics from his enduring “Lectures on Physics, Vols. I-III”. I wanted to tell him how grateful I was.

I have always wondered If Feynman’s Lectures and Liepmann & Roshko’s book on gasdynamics could teach me so much in absentia, how much more I would have learnt if I had been in Caltech. It was quite obvious to me that Dhawan did benefit enormously from the environment and culture at Caltech which was further burnished by his MA in English literature. Of the many Indian Caltech alumni I have met and know, none has impressed me as Dhawan has. The difference I believe was his MA in English literature. It made him a person one felt delighted to know and listen.


Professor Rajendra Bera is the author of a forthcoming book, The Amazing World of Quantum Computing to be published by Springer Nature. He is also the sole inventor on 28 US patents, all assigned to IBM.

References


---

**Rajendra K. Bera**

Dr. Rajendra K. Bera is the Chief Mentor at Acadinnet and a former honorary professor at the International Institute of Information Technology, Bangalore where he taught two courses: quantum computing, and intellectual property rights. In November 2005 he retired as the head of the R&D group of IBM Software Labs, India. He was with IBM during 1995-2005. Earlier he was a research scientist with the National Aerospace Laboratories, Bangalore (1971-1995); a visiting assistant professor of aerospace, mechanical, and nuclear engineering, University of Oklahoma, USA (1979-1980); and a visiting faculty of aerospace engineering at the Indian Institute of Technology, Kanpur (1988) where he taught fighter aircraft design.
A COMPREHENSIVE SURVEY ON RECENT ADVANCEMENTS IN COGNITIVE RADIO-BASED INTERNET OF THINGS

Rajalekshmi Kishore
Sanjeev Gurugopinath, Department of Electronics and Communication Engineering, PES University

DOI: https://doi.org/10.34048/2020.12.F1
Abstract

The recent research studies on the Internet of Things (IoT) predict that the conventional IoT system without cognition will just be a burden on existing network infrastructure. In this article, we present a comprehensive survey on the recent research advancements in Cognitive Radio (CR)-based IoT (CR-IoT) framework. First, we provide a detailed discussion on the need to employ CR framework for IoT. Towards this end, we discuss in detail the challenges associated with the conventional IoT paradigm on the communication technology. This also includes several interdisciplinary requirements, such as technical challenges, security, hardware, standard, and business challenges that are the key factors for the deployment of CR-IoT systems. Next, integration of emerging techniques such as Blockchain and Machine Learning with CR-IoT, and the associated advantages are discussed. Finally, we brief on the research challenges and future research directions in a CR-based IoT system.

1. Introduction

The concept of Internet-of-Things (IoT) has captured the attention of research community with a vision of connecting physical objects over the Internet at an unprecedented rate [1]. This network of physical objects is an interconnected web of smart devices, sensors, and other individual nodes, through which data can be collected. The accumulated data in its raw form is transmitted over the Internet to be analyzed and processed to find certain patterns or trends [2]. For instance, considering the present global situation of COVID-19 outbreak, inclusion of an integrated digital disease surveillance system may be crucial to control the growth of this pandemic [3]. Motivated by several such important applications, IoT is gaining global attention and becoming increasingly available for predicting, preventing and monitoring emerging infectious diseases. State-of-the-art IoT-enabled health monitoring systems [4] provide real-time surveillance through the use of wearable health-monitoring devices [5], cloud-based remote health testing [6], and artificial intelligence (AI) [7]. Therefore, IoT based smart disease surveillance systems have the potential to prove to be a major breakthrough [8].

Efficient utilization of data handling techniques based on IoT and other new technologies could help in the prediction and early recognition of outbreaks, which can be used to prevent the spread of infectious diseases [9]. Smart disease surveillance systems based on IoT would provide simultaneous reporting and monitoring, end-to-end connectivity and affordability, data assortment and analysis, tracking and alerts, as well as options for remote medical assistance to be adopted, to detect and control infectious disease outbreaks in affected countries [10]. The current situation acts as one of the recent applications where advanced health care system can be developed by an advanced IoT-based system. Hence, the deployment of IoT systems not only provides continuous data assortment and analysis but also provides significant savings and revenues in many areas, such as monitoring and maintaining solutions [11]. This has forced businesses to adopt different strategies to cope with this exponential growth and the challenges associated with them, such as allocating sufficient spectrum bands in IoT applications [12, 13].

Despite its advantages, an IoT system without a brain is not enough to bring the expected convenient and comfortable life to humans. This observation has motivated to the recent development of a new paradigm called as the cognitive Internet-of-things (CIoT) [14], which integrates the operational process of human cognition into the IoT system design. The CIoT paradigm also facilitates systematic developments in the key enabling techniques for the fundamental cognitive tasks involved in its research and development. The cognitive management framework also empowers the IoT towards a better supported sustainable smart city development, where cognition mainly refers to the autonomic selection of the most relevant objects for the given application – a feature which is not present in the conventional IoT. In [14], CIoT is viewed as the current IoT integrated with cognitive and cooperative mechanisms to promote performance and achieve intelligence.

A related topic is the concept of Cognitive Radio Networks (CRN) – which was first proposed by Mitola in 1999 [15], and revisited by Haykin in 2005 from a signal processing perspective [16]. Since then, research on CRN has been one of the actively studied topics in the field of wireless communications. One common point in the fields of CIoT and CRN is that both of them benefit from the recent advances in cognitive sciences [14]. However, the differences between CIoT and CRN are much more than the commonalities, which are listed next. The fundamental application of CRN is well-known, which is its potential as a promising paradigm to improve the utilization of radio electromagnetic spectrum, by allowing unlicensed radios to opportunistically access the idle spectrum licensed to the primary radios [17]. Therefore, CRN is essentially a radio system with the objective to improve the wireless network throughput. On the other hand, CIoT generally consists of (massive) heterogeneous general objects, not just radios, with various objectives designed for different applications. Recently, the adaptation of CRN in IoT was presented in [18], which showed equivalent or better performance than currently used networks such as Wi-Fi, WiMax and Bluetooth [19]. An architecture of a CR-enabled IoT network, that coexists geographically with several primary user (PU) networks, is shown in Figure 1. As depicted, the CR-IoT network can be home
area networks, body area networks, or vehicular networks, which utilize IoT functionalities. The PUs are licensed to transmit over a set of orthogonal non-overlapping channels, to their dedicated receivers. The CR-IoT devices periodically scan the vacant PU spectrum in order to access them opportunistically. Therefore, CR is considered as key IoT enabling technology, and its integration with future IoT architectures and services is expected to empower the IoT paradigm.

The need for an implementation of a CR-IoT network is also motivated by the need for an optimal usage of scarce spectrum resource by opportunistic approaches for radio frequency (RF) spectrum, which is one of the main challenges at hand [20]. The recent, unprecedented increase of wireless communication systems, applications and their users, limits the spectrum resources and connectivity between the interconnected objects. Therefore, the demand for smart devices that can manage and configure their transmission parameters based on the spectrum availability in spatio-temporal dimensions has remarkably increased. The CR technology is the best candidate to meet this requirement, due to its adaptability and intelligence to automatically detect available channels in a wireless spectrum. After that, it possesses the ability to change its transmission parameters to enable high data rate, concurrent communications, and also facilitates in improving the radio operating behavior [21]. In particular, new spectrum resources can be gained by allowing non-licensed users – also called as CR users or secondary users (SU) – to share a licensee user – also called as the PU – frequency bands, such that PUs should be always protected from the interference caused by SUs. Therefore, CR users must possess an additional feature that enables protection of the PUs to be of higher priority over any other own interest. The main benefits of CR systems for IoT devices would be to [19]

- Relieve the spectrum scarcity by transmitting on the unused spectrum, and at the same time avoiding interference with a PU,
- Avoid radio interference and jamming, based on an appropriately chosen spectrum sensing method,
- Support switching to enable the utility of a suitably designed power saving protocol,
- Improve the quality of communications by using high bandwidth services, and
- Improve quality of service (QoS), thereby enhancing the availability, suitability and reliability.

There are three paradigms or schemes by which a CR/SU can share the spectrum band owned by a PU, namely, overlay [22], underlay [23] and interweave [24]. For overlay and underlay schemes, a CR can coexist with the PU without interference or with a predefined, minimum interference. In the underlay scheme, the transmission power of the CR should be within a maximum limit, while there is no such constraint in the overlay scheme. However, the CR should have a full knowledge about the signal of PU, so that it can generate an orthogonal signal to the PU’s signal to eliminate/control interference. In the interweave scheme, a CR is allowed to transmit if and only if a PU is not utilizing its spectrum band. When and if the PU resumes its activity, the

---

**Figure 1:** A cognitive radio-based internet-of-things scenario.
SU should leave the spectrum band by either switching to another band or by ceasing its activity until the band becomes vacant again [12]. The CR works through a cognition cycle with four functional phases, namely sensing, decision, sharing, and mobility [25]. The cognition cycle begins with the spectrum sensing (SS) phase, through which the available spectrum resources are detected over the selected PU band. Based on the detection outcomes, the decision is made to concurrently share the band, or to cease transmission in that band. Once a CR decides to exploit the band, a proper medium access control (MAC) protocol is employed and power allocation is considered to satisfy the PU protection [26]. Finally, switching from band to another is performed through the mobility phase.

Given the above advantages, utility of CR technology improves the spectrum usage and alleviates the spectrum scarcity. With this observation, the IEEE has created the IEEE 802.22 WRAN working group [27] to develop a standard for a cognitive radio-based network for non-interfering opportunistic secondary access in TV whitespaces (TVWS). The research community has adopted CR for dynamic radio spectrum management to enhance spectrum usage, e.g., in ISM bands and as SUs in unused TV bands. Moreover, several companies, such as Motorola, Philips, and Qualcomm, are now investing in the development of CR technology [28]. As a related sidenote, the European research cluster on the Internet-of-Things (IERC) has provided the roadmap on standardization for IoT technologies [29]. One of the IERC projects, namely RERUM [30] is working on adapting CR on the IoT devices. The focus is on investigating the adaptation of CR technology in smart objects to minimize wireless interference and ensure the always connected concept. The advanced capabilities of a CR include spectrum sensing, awareness of its surroundings, learning and self-adapting allow to maintain efficient communication in an opportunistic manner. Overall, the cognitive and self-organization capabilities, and reconfigurable nature of CR along with the advances brought about by CR technologies will drive and enable an efficient IoT. In view of this, we discuss the challenges faced by IoT technologies and provide an extensive survey on the approaches to overcome these challenges by integrating CR with IoT systems. Later, we elaborate on several IoT applications that use CRs. Finally, we discuss open research challenges, recent advancements, ongoing and future research directions in CR-IoT systems. The outline of the contributions of this paper relative to the recent literature in the field can be summarized as:

1. We discuss the integration of the IoT and other emerging technologies, including blockchain-enabled CR-IoT, artificial intelligence, and machine learning in IoT services.
2. We discuss CR-IoT based applications in detail, and highlight on how the system can be designed by making it application-specific, so that it can deliver the set of desired IoT services.
3. Finally, we discuss future research directions and challenges in CR-IoT, in terms of hardware design, data analysis, standardization, networking and security aspects.

The remainder of this paper is structured as follows. We identified the potentials of CR technology that make it an indispensable part of the future IoT communications infrastructure. More specifically, Section II provides the summary of key challenges related to IoT research work in that direction followed by current and emerging technology that enables IoT in Section III. Section IV, describes application of CR in IoT. Section V discusses the integration of IoT with other emerging techniques. Finally, in Section VI we conclude future research directions and challenges in CR-IoT, in terms of hardware design, data analysis, standardization, networking and security aspects.

2. Challenges and Requirements in IoT Paradigm

The IoT paradigm enables the regular physical objects to be smart by exploiting its underlying technologies such as ubiquitous and pervasive computing, embedded devices, communications and sensor networks, IP and applications. However, IoT by itself lacks technology and standards that integrate the virtual world and the real physical world in a unified framework. In the following, we shall shed light on some of the design issues and challenges in IoT – as shown in Figure 2, which are currently hindering IoT to achieve its fullest potential [31]. Later, we argue that CR can be a promising enabler technology for IoT as it can help in addressing many of these following challenges.

A. Technical Challenges

It’s not an easy mission to successfully establish a smooth functionality of IoT systems as a standalone solution or part of existing systems since, there are many technological challenges, which include security, connectivity, compatibility and longevity, standards and intelligent analysis and actions.

a) Security: The main idea behind the working principle of IoT has already put forth a serious security concern, which has drawn the attention of prominent tech firms and government agencies across the globe [32]. The hacking of baby monitors, smart fridges, thermostats, drug infusion pumps, cameras and even the radio in a
car are signifying a security nightmare being caused by the future of IoT [33]. New nodes being added to networks and the Internet will provide malicious actors with innumerable attack vectors and possibilities to carry out their evil deeds, especially since a considerable number of them suffer from security holes. The more important shift in security will come from the fact that IoT will become more ingrained in our lives. Concerns will no longer be limited to the protection of sensitive information and assets. Our very lives and health can become the target of IoT hack attacks [34].

b) Connectivity: Connecting many devices will be one of the biggest challenges of the future of IoT, and it will defy the very structure of current communication models and the underlying technologies [35]. At present, we rely on a centralized server/client paradigm to authenticate, authorize and connect different nodes in a network. Such a model is sufficient for current IoT ecosystems, where tens, hundreds or even thousands of devices are involved [36]. But when networks grow to join billions and hundreds of billions of devices, centralized systems will turn into a bottleneck. Such systems will require huge investments and spending in maintaining cloud servers that can handle such large amounts of information exchange, and the entire system can go down if the server becomes unavailable. The future of IoT will very much have to depend on decentralization in terms of networks. Part of it can become possible by moving some of the tasks to the edge, such as using fog computing models where smart devices such as IoT hubs take charge of mission-critical operations and cloud servers take on data gathering and analytic responsibilities [37]. Other solutions involve the use of peer-to-peer communications, where devices identify and authenticate each other directly and exchange information without the involvement of a broker. Networks will be created in meshes with no single point of failure. This model will have its own set of challenges, especially from a security perspective, but these challenges can be met with some of the emerging IoT technologies, such as blockchain [38].

c) Compatibility and Longevity: IoT is growing in different directions, with different technologies competing to become the standard. This will cause difficulties and require the deployment of extra hardware and software when connecting devices. Other compatibility issues stem from non-unified cloud services, lack of standardized M2M protocols and diversities in firmware and operating systems among IoT devices. Some of these technologies will eventually become obsolete in the next few years, effectively rendering the devices implementing them useless. This is especially important, since in contrast to generic computing devices which have a lifespan of a few years, IoT appliances (such as smart fridges or TVs) tend to remain in service for much longer, and should be able to function even if their manufacturer goes out of service.

d) Intelligent Analysis and Actions: The last stage in IoT implementation is extracting insights from data for analysis, where analysis is driven by cognitive technologies and the accompanying models that facilitate the use of cognitive technologies. Unnecessary competition and deployment barriers in markets, will block the migration of IoT systems to connect as many Things as possible.

Figure 2: Challenges in IoT
B. Hardware challenges

Smart devices with enhanced inter-device communication will lead to smart systems with high degrees of intelligence [39]. Its autonomy enables a rapid deployment of IoT applications and creation of new services. Therefore, hardware research is focusing on designing wireless identifiable systems with low size, low cost and yet, sufficient functionality. As the bandwidth of IoT terminals could vary from kbps to Mbps from sensing [40] simple value to video stream, requirements on hardware are diverging. However, two requirements have been nevertheless the essentials. First is the extremely low power consumption in sleep mode and the other is ultra low cost. Suppose the sleeping time over active time is one million, the leakage power of an IoT terminal shall at least be one million time less than that of active. It is so far impossible when an IoT terminal is sleeping and receiving RF signals. It will be even difficult when using advanced CMOS silicon with relatively more leakage power. Hardware and protocol design are the first hardware challenges of IoT [41]. The cost, performance and power are few factors which are interdependent. The cost of an IoT terminal must be ultra-low. However, so far, there is no low-cost positioning solution for IoT, especially the positioning precision of a short-range IoT terminal must be high. Low active power is also a challenge for low-cost terminal [42]. Traditionally, low cost equals to lower performance or longer process latency. Longer processing latency ends up in higher energy consumption. As the spectrum resource is very limited at the lower part in L band, IoT uses higher RF such as the frequency bands higher than 5 GHz. To use very narrow band with strong power neighbors, the cost of passive component will not be low and that will definitely be a potential challenge in the future.

C. Architecture Challenges

In IoT, data integration over different environments are challenging and will be supported by modular interoperable components. Infrastructure solutions will require systems to combine volumes of data from various sources and determine relevant features to interpret the data, show their relationships, compare data to historical useful information, and support decision-making. Below are some of the architecture challenges faced by IoT.

e) Lack of Standardization: One of the many significant issues faced by an IoT Architecture is the variety of languages, protocols, and standards. Also, the lack of agreement on which it works. IoT architecture does not have a single platform of standardization. It is changed due to the difference of connected things. The hardware and software components related to the IoT should also be standardized. Having a standardized Application Programming Interface (APIs) and software service enables easy migration across systems.

f) Connectivity: Connectivity is an important component of IoT architecture because it plays a crucial role in transporting data from the sensors [43]. Furthermore, it also transmits instructions to the actuators. In the context of IoT, connectivity can be examined at various levels from the frequency bands at the physical layer to MAC protocols at the link layer, transport protocols at the transport layer, network protocols, and mobility at the network layer and application protocols at the application layer.

g) Interoperability: This is one of the major challenges of an IoT architecture. IoT applications offer value to users by combining data sets from various IoT devices. They do this in order to generate interconnected and complex business rules [44]. These business rules further actuate IoT devices to automate the process. Effective business rules generation requires interoperability, which is not an easy task because of multiple vendor problems and legacy system issues.

h) Mobility: Mobility is a common challenge in IoT architecture, where the devices have to move a lot. Therefore, the devices change their IP addresses and networks frequently based on their locations. Since there are a large number of devices and a massive amount of data transmission which leads to network disturbance and data loss, single reference architecture cannot be a blueprint for all applications. Heterogeneous reference architectures have to coexist in IoT. Architectures should be open, standardized and it should not restrict users to use fixed, end-to-end solutions. IoT architectures should be flexible to cater for cases such as identification (RFID, tags), intelligent devices, and smart objects (hardware and software solutions).

D. Privacy and Security Challenges

Cybersecurity is one of the more pressing challenges when it comes to IoT. With the nature of this system being widely connected, it leaves vulnerabilities for hackers to take advantage. As the basic principle of IoT involves connecting devices, it makes everything addressable and locatable which in turn makes our life easier [45]. However, making everything connected to Internet opens the door for hackers. Without proper confidence about privacy and security, user will not be attracted towards IoT [46]. So, it must have a strong infrastructure dealing with security. Below are some of the issues that IoT might face.

* The primary issue is unauthorized access to RFID, which contains user information that can be easily modified or read by the reader. This opens a whole bunch of threat for the user as the data can be easily accessed by a malicious reader [47].
Acquisition of data is also possible other than transmission. In this scenario, some of the possible attacks include tampering where the data in the node can be extracted or altered.

Footing [48] is another security treat, which is caused when traffic amount is high and exhaustion of memory takes place.

Sybil attack [49] is yet another attack wherein multiple pseudo identities are claimed for a node in order for it to give big influence.

Cloud computing in IoT is a big network that allows sharing of resources and some of the security threats faced by shared resources are as listed below:

- Data loss happens when any offender user having unauthorized access can modify or delete the data.
- Cloud computing [50] can also be used for controlling other devices. Once the hackers get hold of an account, they can upload/change certain specific softwares which will give them control of any devices that come in contact.
- In man-in-the-middle (MITM) [51] attack, the hacker works as a third person and can intercept or alter any message.

E. Standards Challenges

Standards play an important role in forming the IoT ecosystem. A standard is essential to allow all actors to equally access and use [52]. Developments and coordination of standards and proposals will promote efficient development of IoT infrastructures and applications, services, and devices. In general, standards developed by cooperated multi-parties, and the information models and protocols in the standards shall be open. The standard development process shall also be open to all participants, and the resulting standards shall be publicly and freely available. In today’s network world, global standards are typically more relevant than any local agreements.

F. Business Challenges

The growing advancement in sensor technology with respect to range, sensitivity, and resolution in terms of speed and coverage, led to the advancement of IoT implementation. Accenture’s latest Industrial IoT trend study confirms that 60% of the companies are already engaged in IoT projects, more than 30% are at an early stage of deployment, and 69% of these companies’ IoT initiatives are focused on reducing operational cost [53]. This advancement in technological initiatives can help businesses generate insight and provide valued services to their customers, at a reduced operational cost. On the other hand, despite the availability of resources and advancements in key IoT components such as sensors and networks, many companies have not been able to make substantial benefits out of their IoT project. According to a data released by Cisco, 74% of surveyed organizations have failed with their IoT initiatives [54]. This is mainly because there are several human factors involved in IoT implementation, beyond the functional elements of sensors and networks. An effective collaboration and integration among all the components of IoT, along with creating a culture of technology within the organization is required to succeed. But in doing so, many companies face critical challenges related to security network investment and data analysis. Moreover, most of the companies are often prepared to handle sensors and data analytics, whereas, they fail to bring the same level of expertise into network investment, system integration, and security. Besides, on the occasion of IoT World Forum 2017, Cisco released a survey that highlights a perception gap between the IT executives and business executives [55]. About 35% of surveyed IT executives perceived their IoT project as successful, while only 15% of business executives believe that the initiative was a success. This percentage gap highlights a rift between technology and value delivered. While the IT department is sure of bringing the technology required for a successful IoT project, it is the lack of value delivered, which makes business executives consider the IoT project to be unsuccessful. Capturing invaluable data that lead to false insight could also be the reason many business executives consider their IoT projects unsuccessful.

3. Motivation for CR in IoT Applications

In section 2, we discussed some of the design issues and challenges that are currently hindering the realization of IoT, such as technical challenges, heterogeneity, low cost low power design, interoperability, connectivity, compatibility etc. which were categorized as technical, architectural, hardware, security, and business challenges. Current fixed spectrum utilization policies have resulted in inefficient utilization of spectrum [18]. To overcome this situation, CRNs opportunistically search for available spectrum bands by dynamically changing transmitter parameters based on interaction with the environment. Each node has the characteristic of fast switching with simultaneous transmissions in addition to channel probing and channel state learning. Dynamic Spectrum Access (DSA) capability enables a CR user to adapt to varying network conditions. A CR user is allowed to use the spectrum with guaranteed protection to the legitimate user, or the PU. With the ongoing development in CRNs and IoT, it is envisioned that CR-based IoT frameworks may become necessary...
requirements in the future. The IoT objects would be equipped with cognition in order to learn, think, and make decisions through understanding of both social and physical worlds [46]. Additional requirements include intelligent decision making, perception action cycle, massive data analytics, on-demand service provisioning, semantic derivation, and knowledge discovery. Therefore, a CR-based IoT is a foreseeable need in the future, which may be due to the following reasons.

- The main motivation comes from bandwidth allocation for IoT devices. The number of IoT objects is expected to grow in large numbers, and it will be very difficult to allocate spectrum bands to these objects. Additionally, the number of PUs will also increase, thus creating problems for unlicensed users. Fixed spectrum assignment policy requires cost to purchase spectrum; therefore, spectrum assignment for such a large number of objects can create unnecessary expenditures. Towards this end, CRNs can facilitate a smooth functioning of IoT devices in all of these situations. Traditional communication techniques do not support spectrum sharing among multiple users. However, CRNs, with its spectrum sharing advantage, will be a boost in the future as objects will increase, each looking for spectrum access [19]. Hence, CRNs may perceive the spectrum environment and provide on-demand services among users through intelligent decision making.

- Cellular communication incurs costs, while Bluetooth and Zigbee have limited range. IEEE 802.22 for a Wireless Regional Access Network (WRAN), a CRN standard, has a long range support and may also be suitable for short-to-long range applications [56].

- CR-based IoT frameworks can alleviate interference situations by looking for interference-free channels through dynamic spectrum access capability [57].

- Mobility will be an important feature of future IoT-based structures, and it will be exceedingly difficult to provide a single continuous communication facility everywhere. Equipped with cognitive capability, IoT objects can achieve seamless connectivity [58].

- To support big data generated by a huge number of IoT objects in the future and semantic derivation issues, a new paradigm of cloud servers is gaining popularity. It is expected that CR-based IoT objects will autonomously search for available storage places in cloud servers and send data by utilizing spectrum sensing [59].

4. Emerging Applications with CR-IoT

In this section, we present some of the potential applications of IoT that can benefit from its integration with CRNs. Table I summarizes the state-of-art literature on applications of CR-IoT in terms of technical parameters relevant to IoT and CR technologies.

- Smart Grids and Smart Metering: The centralized management of the electric grid has experienced a paradigm shift towards the smart grid (SG). It enables the two way communication, which connects the consumer side and the utility companies [72]. This paradigm provides an efficient management and control of the grid in a better way by monitoring of demand and supply in real time. Hence, SG adds new services to the customers, in which they can adjust their energy consumption according to the demand and supply balance. The SG can benefit from the application of the IoT technology since the IoT enables smart monitoring of the environment, efficient management of resources by smart meters, home gateways, smart plugs and connected appliances which provides efficiency in energy production and consumption [60]. Autonomous monitoring, diagnosis and control, and efficient operation of the power equipment for the power generation, and distribution require monitoring and actuating devices [73]. Hence, the inclusion of the IoT in SG is important. This will increase the utilization of information and communication technologies in the SG and provide effective management of power grid system [61]. One major drawback here is the transfer of large volumes of data from a number of meters/devices in a limited spectrum bandwidth without interference to long distances. Current wireless techniques have issues regarding this. Wired techniques such as DSL and optical fiber, and wireless techniques such as cellular can overcome these problems, but need huge expenditures for cable/fiber installation or spectrum purchase. Therefore, CRNs as a viable solution for this issue [74]. An IoT-enabled SG architecture with cognitive radio can be seen in Figure 3, which shows that the elements of the SG such as energy harvesting cognitive radio nodes which communicate over the Internet. The condition of power grid elements, their energy production, the consumption of the energy are observed and measured.
### TABLE I: Summary of the state-of-art literature on different applications of CR-IoT

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Technology Parameter</th>
<th>IoT Context</th>
<th>CR Use</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>[60]</td>
<td>Energy harvesting (EH) approaches</td>
<td>To increase the lifetime of wireless devices</td>
<td>To overcome harsh channel conditions and spectrum scarcity problems in its network architecture</td>
<td>The proposed network is investigated in terms of general operation node and network architecture aspects</td>
</tr>
<tr>
<td>[61]</td>
<td>NB-IoT is proposed to provide secure and reliable communications among different components in a power system</td>
<td>Four typical communication scenarios such as rural area, bad urban area, typical urban area, and hilly terrain area are examined</td>
<td>Conventional communication technologies such as ZigBee, WiMax, LPWAN, and LoRa are investigated</td>
<td>NB-IoT framework performs well in the presence of LOS paths and the performance can be improved with the sub-carrier interleaving technique</td>
</tr>
<tr>
<td>[62]</td>
<td>Novel concept of introducing the CR in the smart home environment</td>
<td>IoT-based smart home that facilitates the CR network for efficient spectrum sensing and management is investigated</td>
<td>A prototype for implementing CR-based communication for smart homes was developed using NI USRP and LabView</td>
<td>Prototype implementation of smart home using SDRs</td>
</tr>
<tr>
<td>[63]</td>
<td>IoT-enabled home automation and security systems using passive infrared motion sensors, temperature sensors, smoke sensors, and web cameras for security surveillance</td>
<td>A framework to build a low-cost smart home security system using affordable components such as Arduino-compatible, Elegoo Mega 2560 micro-controller board with Raspberry Pi and RF signals</td>
<td>Low-cost, RF-based transceiver communication is utilized to create an IoT-enabled smart home security system</td>
<td>An architecture for a cost-effective smart door sensor that will inform a user through an Android application, of door open events in a house or office environment is proposed</td>
</tr>
<tr>
<td>[64]</td>
<td>A CR system is proposed for eHealth applications in a hospital environment, which protects the medical devices from harmful interference by adapting the transmit power of wireless devices based on electromagnetic interference (EMI) constraints</td>
<td>Two e-Health applications, namely, tele-medicine and a hospital information system are considered</td>
<td>The cognitive capability of the system arises due to its EMI awareness to control the wireless access parameters in order to achieve the desired QoS differentiation among different users/applications</td>
<td>Admission control algorithm, and transmit power control methods are used to limit the number of secondary users in the system. This improves the delay and loss probability of the CR system</td>
</tr>
<tr>
<td>[65]</td>
<td>ECC-based smart-healthcare system</td>
<td>To monitor and analyze the physical health of users using cognitive computing</td>
<td>It adjusts the computing resource allocation of the whole edge computing network according to the health-risk grade of each user</td>
<td>Optimizes the computing resources, improves the survival rates of patients in a sudden emergency</td>
</tr>
<tr>
<td>[66]</td>
<td>Explored the potentials of CR technology in future public safety communication infrastructure</td>
<td>Public safety operation embracing SG and IoT to periodically send the sensing information to that cloud using LTE</td>
<td>Introduces a case study and shows how CR technology can be used to enhance the experience of the first responder by improving communication capabilities</td>
<td>CR technology to enhance the emergency service by improving their communication capabilities</td>
</tr>
<tr>
<td>[67]</td>
<td>The operational and technical requirements for DRN is explored for emergency response organizations and personnel</td>
<td>Disaster Response scenario for an area after communication is destroyed, including UAV-based interoperable systems</td>
<td>Assessed the potential for CR technology in DRN solutions</td>
<td>CR technologies can be utilized in future mission-critical situations</td>
</tr>
<tr>
<td>[68]</td>
<td>A secured and efficient communication scheme for a decentralized CR-based IoV (CloV) network is proposed</td>
<td>The performance of CloV is evaluated in terms of packet delivery, packet loss ratio, end-to-end delay, and throughput, using NS-2</td>
<td>The RSU senses the spectrum using an energy detection method</td>
<td>The CR-based approach of CloV outperforms the existing schemes and significantly enhances the performance of the underlying network</td>
</tr>
</tbody>
</table>
via the IoT devices. The observations and measurements are sent over multiple hops to reach the Internet gateway, to send them to remote areas for necessary actions to be taken in the IoT-enabled SG. Power consumption in industrial, commercial and residential sites can be monitored, and the power generated by renewable energy resources can be observed in real-time via the Internet which increases accessibility.

2) Smart homes: Rapid growth of wireless technology prompted for the development of smart home which consist of a large number of self-organizing wireless smart devices like smart phones, wireless cameras, laptops, HDTV, smart meters and smart domestic appliances (see Figure 4). It is envisioned that with the evolution in technology, the era of IoT will be a necessity in the future. Homes will be incorporated with sensors/devices to perform everyday functions for improved quality of life. Home automation and home energy management are already present in the form of certain examples such as smart fridge and smart lights, respectively. These smart devices are equipped with small radio devices for monitoring and remote controlling. Some of these devices use short range unlicensed bands like ISM band, which is overloaded. Also, a large number of smart home devices tend to create inter-user interference while using these bands. The future smart homes will face spectrum shortage problem. The solution to these problems is incorporating a CRN. Additionally, switching from unlicensed ISM band to TVWS will reduce energy consumption [73].

3) Healthcare and medical applications: We have healthcare applications of IoT already in the practical domain. Smart sensors are deployed on and around a patient to monitor critical data such as temperature, blood pressure, glucose level, and others. With remote monitoring, medical staff continuously observes the parameters. Wireless solutions are already there; however, it is urgent that smooth monitoring is ensured. Towards this end, healthcare information is to be relayed to medical staff without any need for spectrum assignment. CR-based IoT frameworks, as shown in Figure 5, can achieve this to long ranges without any worries about spectrum availability [75].

At the outbreak of global crisis due to coronavirus (COVID-19), infectious disease epidemiology is another emerging field of IoT based health care application which is rapidly gaining global attention and becoming increasingly available for predicting, preventing and monitoring emerging infectious diseases. State-of-the-art IoT-enabled health monitoring systems provide real-time surveillance through the use of wearable health-monitoring devices, cloud-based remote health testing, and artificial intelligence. The ubiquitous availability of smart technologies, as well as increased risks of infectious disease spread through the globalization and inter-connectivity in the world necessitates its use for predicting, preventing and controlling emerging infectious diseases [76]. IoT-enabled health monitoring in a global health care infrastructure would provide targeted information for health officials, and has the potential to improve efforts to locate, contain, and prevent infectious diseases. It may help to quickly diagnose infected patients and accurately predict the possible spread of a disease to other locations utilizing travel data. Ultimately, an IoT based surveillance system might help reconstruct the progression of an outbreak and stabilize the economy of the source country rather than having to lock down major cities, borders and businesses.

4) Emergency Networks: Public safety agencies are increasingly using wireless communication technologies to monitor disaster conditions using video surveillance cameras and sensors. The increasing use of this set-up has led to congestion in radio frequency channels allocated to the agency [19]. In order to address the aforementioned issue of optimum resource allocation during emergency response, CR-IoT technology could...
be exploited to replace the current state-of-the-art channel allocation protocols [67].

When responding to an emergency situation, firefighters, police officers or lifeguards can pull out information about the internal design of the building and have a knowledge about who is inside by identifying the personal computing devices – smart phones in particular – present in the house. Indoor sensors and smart meters can also help track the spreading of flames, identify the safest place to enter the house and identify all running appliances that may cause safety hazards. This example is depicted in Figure 6. The figure shows that indoor sensing information can be used by the 1-1-2 emergency service team to better assess the required resources attending the incident. As part of the SG and/or the IoT, this sensing information is periodically sent to the cloud using a variety of wireless communications networks such as LTE. The next generation IoT, WSNs, SG, and big data analytics stands as a major enabler to realize future intelligent infrastructures for enhanced disaster management. The widespread demand for data and the emergence of new services are inevitably leading to the so-called resource crisis. Hence, the evolution of the current centralized model of networked systems to new paradigms such as low power high data rate cognitive networks present a suitable path to counteract this crisis.

The CR-IoT scheme is also a potential candidate in disaster management for partially or fully destroyed network. It can be utilized in a Disaster Management Network (DMN) which is a communication network that is rapidly deployed in the aftermath of a disaster, to provide necessary services after existing communications infrastructure has been damaged [67].

In Figure 7, a DMN is shown where non-governmental organization and emergency services perform their initial setup in an operation center. Static/mobile cognitive radio base stations (CRBSs) can be used to provide radio connectivity, with backend connectivity to the operation center in a multihop manner, or with unmanned aerial vehicles. The operation center can be connected to the Internet via satellite/backhaul links or to other nearby base stations/disaster-sites through multihop CRN. In a disaster scenario, we consider the DMN as a SU and the existing/partially destroyed network a PU. Thus the DMN avoids harmful interference on the already existing systems. If the SU detects a PU in the course of operation, it reconfigures its operation in order to avoid causing interference. To assess damage and service requirements, emergency service personnel can carry auto-configuring CR devices to affected areas immediately after the disaster. These CRs can carry out spectrum sensing [77] and use ad-hoc, multihop, and mesh networking techniques to relay information using delay-tolerant applications. Following initial assessment, mobile/static CRBSs can be used to extend connectivity and provide services to inaccessible regions, as shown in Figure 7. Thus, self organizing and dynamic spectrum management capabilities of CR-IoT nominate it as a strong candidate in disaster management network.

CR-IoT can also be applied for emergency networks, e.g. ambulance, fire, police, and rescue, to provide a significant amount of bandwidth to natural disasters that can
cause a collapse of the communications infrastructure. This necessitates an operation in a CR environment to handle the expected huge amount of sensitive real-time data transmission [78, 67]. In Figure 7, we illustrate how the CR rapid deployment and interoperability between network units can provide connection to all communication units, when the communication is affected in disaster relief environments [78].

5) CR-IoT Vehicular Networks: Recently, trends are shifting toward less dependence on human beings, which has led us toward the Internet-of-vehicles (IoV) paradigm where vehicle control is achieved through the integration of communications, controls, and embedded systems. The IoV system is expected to be an autonomous decision maker for traveling. Safe navigation may be possible in the future through information exchange from vehicles to vehicles, from sensors attached on vehicles, and through users’ intentions. The challenge in IoVs is the availability of spectrum for mobile vehicles, and CRN can be a good solution technology due to its long range and interference-free spectrum sensing [79].

6) Military Applications: The future of military combat is going high-tech, as scientists are in process to create an IoT for combat gear embedded with biometric wearables to help soldiers identify the enemy, perform better in battle, and access devices and weapons systems using speedy edge computing [71]. Towards this end, IoT serves as a potential candidate to bring a paradigm shift in military application development. By integrating systems of sensors, actuators, and control systems into existing military infrastructures, the military can become more efficient and effective. Few areas for IoT military applications are listed below:

Logistics: In terms of military application, the connected sensors and digital analytics that IoT technology offers can be used to track supplies and equipment from their source to where they are needed on the battlefield [80].

Smart bases: Incorporating IoT devices and sensors into military bases can have several positive effects [81]. Automated security screening, for example, increases safety while decreasing manpower, and a network of security cameras connected to their environment via sensors and to a central network via the Internet will also minimize security risks. Smart management of resources — e.g. electricity and water — will increase the capacity and output of military bases while ensuring that the well-being of all individuals inside the base.

Data warfare: By collecting data from a wide range of military platforms — including aircraft, weapon systems, ground vehicles, and troops themselves — the military can increase the effectiveness of their intelligence, surveillance, and reconnaissance systems. This wealth of information will allow the armed forces to identify key threats faster and with more accuracy [80].

5. Integration with Other Technologies

Integration of CR-IoT with other emerging technologies such as blockchain, artificial intelligence tech-
niques, machine learning approaches, can make the IoT network smarter and more productive. Furthermore, it will be a new step towards creating next-generation wireless communication systems with additional capabilities such as robust security, autonomy, flexibility, and intelligent architecture which are more energy efficient. This section discusses the impact of adding blockchain, machine learning and ambient backscattering communication (ABC) techniques that support sustainable and independent communications to CR-based IoT systems.

7) Blockchain-Enabled CR-IoT System: The blockchain is defined as a tremendous, public, secure and decentralized datastore of ordered events, called blocks. Each block contains a timestamp and is linked to a previous block. The events can be updated by only a majority users [82]. Erasing information is not allowed. The datastore is owned by no one, controlled by users and not ruled by any trusted third party or central regulatory instance. In fact, trust is encoded in the protocol and maintained by the community of users. The work in [83] presented a comprehensive survey on blockchain, its types, features, structures, protocols, and the integration of blockchain and IoT systems. On the other hand, the survey in [84] focused on blockchain applications and challenges for real world applications. Adopting blockchain technology is a promising solution to improve system immunity against hacking activities. The blockchain prevents central point system failure and cyber attack which results in improving the security in IoT and CRNs. The blockchain protocols employ a two-key encryption system, i.e., public and private keys; therefore, robustness is added to the security of CRNs [82]. However, the use of blockchain in IoT security is limited and focused on the following areas: asymmetric and symmetric key management, trading of collected data, incontrovertible log of events, and management of access control to data [85]. The security issues related to IoT include authentication and authorization, ownership and identity relationships, governance of data and privacy [86]. Blockchain-based research has been used to tackle some of these issues. Such work is still in its infancy and worth investigating since introducing blockchain in IoT and CR systems shows a promising future in supporting secure data sharing and protecting privacy. On the other perspective, employing blockchain protocols in spectrum auction offers a decentralized validation which increases the accessibility of the CRNs and reduces the implementation complexity, since no central entity is required. In addition, blockchain-enabled spectrum access in CRN is a secure spectrum sharing approach where it provides an optimal collision-free method to access the spectrum opportunities.

8) Machine Learning-Enabled CR-IoT System: Machine learning (ML) approaches are artificial intelligence (AI) applications that establish mathematical models based on observations, i.e., training data, to predict or make decisions. ML approaches enable systems to automatically learn and improve from experience without being explicitly programmed. In other words, ML approaches enable computers or processors to automatically learn without human intervention and to select actions accordingly. ML approaches have a potential ability to
analyze and classify massive amounts of data which makes ML approaches as efficient and powerful tools for performing other processing tasks, such as data analysis, classification [87], feature detection, feature extraction [88] and identification [89]. Also, ML approaches improve the security and the data privacy for big data systems [90]. In addition to the aforementioned advantages, ML approaches can apply complex techniques in a simple way, and provide accurate results of identifying the targets. Therefore, integrating ML approaches with cognitive capabilities can improve the effectiveness of processing large volumes of information from various distinctive resources.

9) Ambient Backscattering Communication-Enabled CR-IoT System: Ambient backscatter communication (ABC) technology has been re-introduced recently, and is quickly becoming a promising choice for self-sustainable communication systems, as an external power supply or a dedicated carrier emitter is not required. By leveraging existing RF signal resources, ABC technology can support sustainable and independent communications and consequently open up a whole new set of applications that facilitate IoT. Recently, ABC has been introduced in a CR context, in which the secondary transmitter (ST) can communicate with a secondary receiver (SR) by backscattering the primary user (PU) signal, whenever the PU is active. In other words, instead of initiating a CR transmission only when the PU is inactive, the ST can backscatter the PU signal to SR, even when the PU is active. Towards this end, a novel opportunistic ABC framework for RF-powered cognitive radio (CR) networks was recently proposed [91]. The proposed framework considers opportunistic spectrum sensing integrated with ABC and harvest–then–transmit (HTT) operation strategies. It is shown that operating the CR network in a combination of these two modes improves the overall energy efficiency [92]. Thus, ABC can be deployed within either licensed or unlicensed spectrum [93]. A distributed MAC protocol in the ambient backscatter communication system for IoT networks was proposed in [94], which enables each backscatter device (BD) to switch freely among backscatter transmission, receiving and energy harvesting in a distributed way. Thus ambient backscatter communications would be adapted to wearable devices, connected homes, industrial Internet, and miniature embeddable to provide pervasive connectivity [95].

6. Future Research Directions in CR-IoT

A. Hardware Designs for CR-IoT

The effective utilization of CRs requires a good hardware design. CR antennas used in one frequency spectrum (cellular) are not the same in size as compared to the ones used in some other spectrum (ISM 2.4 GHz). Furthermore, transmission power levels also vary according to environment. Selection of single-radio or multi-radio is also required [1]. Furthermore, IoT objects are connected to networks through gateways. Thus a gateway design requires flexibility, scalability, security, and energy efficiency. In the case of CR-based IoT objects, additional requirements arise in the form of efficient spectrum resource utilization, particularly in a multi-user scenario [96]. Usually, CR users search for spectrum access individually, but if IoT objects are energy-constrained, gateways may perform spectrum sensing for them. Geo–location-based spectrum searching with history keeping may be a good option. Flexibility and interoperability may be achieved through software defined radios.

B. Data Analysis

Yet another challenge in CR-based IoT is to manage huge volumes of data. Effective algorithm design is required to analyze and interpret high-dimensional and nonlinearly separable data which is heterogeneous and difficult to process. Once categorized, decision making is also difficult, keeping the diversified nature of applications in mind. In some cases, cognitive applications utilize learning/reasoning algorithms and select processes based on history, which may consume time. Hence, it is necessary to discover advanced knowledge-based algorithms for improved performance.

C. Spectrum-Related Functions

Spectrum sensing is a preliminary and fundamental step in CRs. IoT objects behaving like CRs have to look for spectrum in a dynamic environment in the presence of a number of PUs, and the problem is boosted when PUs are in the same band. At the same time, differentiation between licensed and unlicensed signal is also important. Spectrum sensing takes time and
consumes energy. Therefore, fast and energy-efficient algorithms should be designed. It is observed that CRs do not perform any task while sensing spectrum. We believe that multiple radios should be employed where one radio performs spectrum sensing while another transfers the data. However, it will be effective if it is coupled with advanced geo-location-based and history-keeping algorithms. Additionally, we cannot compromise on misdetection probability, but we can compromise on false detection probability. Mis-detection relates to a situation where a PU is not detected, and false detection is detecting a PU when there is actually none [97]. If spectrum is sensed ideally, making a decision among searched bands about data transfer is an issue. Of multiple bands, there is a possibility that certain bands do not meet application-specific QoS requirements. Critical applications need real-time answers, while non-critical applications may compromise on response time. Moreover, assigning spectrum among multiple CRs is also a challenge, and requires resource-shared and application-specific algorithms.

D. Standardization Activities and Challenges

Standardization efforts in IoT are still in their early stages, and standardization of CR-based IoT is a meager topic to discuss. The first and most important problem is related to developing semantic standards as it can accelerate the process in the direction of interoperable solutions. CR-based IoT frameworks may have to support a large number of diversified devices for successful semantically operable scenarios. The physical presence of a huge number of devices is connected to a variety of application demands. This requires continuous semantic information sharing. Other key areas of standardization involve communication protocols at each layer, objects selection, and transmission power. The overall benefits can be achieved if standardization activities are carried out jointly around the globe. Moreover, these efforts have to be streamlined and quick to save cost, which may become another problem in the future.

E. Networking and Addressing

The existence of several PUs and CRs, and energy levels required for transmission create problems for CRs to sense and decide spectrum efficiently. This creates negative impact on the network. Meanwhile, the multi-user scenario has issues as well. The thing-oriented, semantic-oriented, and Internet-oriented nature of IoT already poses problems. When joined with cognitive-capable objects, the problem will be magnified [98].

F. Security and Privacy

The heterogeneity in CR-based IoT frameworks has security problems as we cannot apply the same security levels to all situations. The adaptive capability of CR can become a security problem as an intruder may pretend to be a CR [99]. The security standards have to be applied at all levels, from technology to consumer/business to legal frameworks. Access to information should be easy for validation and monitoring conditions. The introduction of a smart object also introduces privacy issues. A major source of privacy issues is the misuse of application. Privacy is required at the data collection point as CR-based IoT will be a global framework with a variety of technologies and with diverse data. Data is collected using RFID, WSNs, cellular phones, and others, which may lead to severe privacy-related threats. As data is shared along the way, privacy also has to be maintained at data sharing levels. Thus, data anonymization with strict control, management among objects, and identification is required [47].

7. Conclusion

In this article, we have presented the need for cognitive radio networks for IoT. Both technologies are in early stages, but little work has been done in CR-IoT. Consequently, realizing CR-based IoT is still a new paradigm and needs to be studied in detail. It is envisioned that IoT without cognition will just be a burden on existing network infrastructure. However, to support diversity of applications and numerous heterogeneous connected devices, the IoT paradigm poses new challenges to the communication technology. Cognitive radio is considered as a key IoT enabling technology and its integration with future IoT architectures and services will empower the IoT paradigm. We have presented a detailed survey on several interdisciplinary challenges and requirements, such as technical challenges, security, hardware, standard, and business challenges that need to be addressed for the IoT deployment. Integration of emerging techniques such as blockchain with CR based IoT were discussed. We have assessed the potential for CR technology in IoT, and we have outlined how CR can empower IoT. Finally we briefed on the research challenges and future research directions in CR-based IoT.

References


[41] K. L. Lueth. (2019, September) 40+ emerging iot technologies you should have on your radar. [Online]. Available: https://iot-analytics.com/40-emerging-iot-technologies-you-should-have-on-your-radar


REFERENCES


Dr. Sanjeev Gurugopinath received B. E. degree in Electrical and Electronics Engineering from Dr. Ambedkar Institute of Technology, Visvesvaraya Technological University, Bengaluru, in 2004, the M. Tech. degree in Digital Electronics and Communication Engineering from M. S. Ramaiah Institute of Technology, Visvesvaraya Technological University, Bengaluru, in 2006, and Ph.D. in signal processing for communications from the Indian Institute of Science, Bengaluru, in 2015. He is currently a Professor with the Department of Electronics and Communication Engineering, PES University, Bengaluru. His research interests are in the areas of cognitive radios and heterogeneous networks for 5G communication systems, powerline communications, underwater acoustics, and speech signal processing. He was a co-recipient of the Best Paper Award across all tracks at IEEE INDICON 2016. He is a member of IEEE, and a life member of ACCS.

Dr. Rajalekshmi Kishore received her B.E. degree in Electronics and Communication Engineering from N.I. College of Engineering, Tamil Nadu, India in 2003 and the M.E. degree in Communication Systems from National Engineering College, Anna University, TamilNadu, India, in 2005. She completed her Ph.D. degree in Communication Engineering from the Birla Institute of Technology and Science (BITS) Pilani, India, in 2018. She was a visiting research scholar at Khalifa University, Abu Dhabi, UAE in 2018 and she had earlier worked with Jain University Bangalore, India and Electrical and Electronics Engineering Department, University of Bahrain (UoB), Bahrain. She has served as TPC member in various IEEE conferences and her current research is focused on cooperative communications, Energy efficiency in cognitive radio networks, NOMA and IoT.
COMPUTER VISION WITH DEEP LEARNING FOR PLANT PHENOTYPING IN AGRICULTURE: A SURVEY

VINEETH N BALASUBRAMANIAN, Associate Professor, IIT-H
WEI GUO, Assistant Professor, University of Tokyo
AKSHAY L CHANDRA, Research Assistant, IIT-H
SAI VIKAS DESAI, Pursuing Master’s Degree, IIT-H

Abstract

In light of growing challenges in agriculture with ever growing food demand across the world, efficient crop management techniques are necessary to increase crop yield. Precision agriculture techniques allow the stakeholders to make effective and customized crop management decisions based on data gathered from monitoring crop environments.

Plant phenotyping techniques play a major role in accurate crop monitoring. Advancements in deep learning have made previously difficult phenotyping tasks possible. This survey aims to introduce the reader to the state of the art research in deep plant phenotyping.

1. Introduction

Population growth, increasing incomes, and rapid urbanization in developing countries are expected to cause a drastic hike [1] in food demand. This project-ed rise in food demand poses several challenges to agriculture. Owing to a continuous decline in global cultivable land [2], increasing the productivity of the existing agricultural land is highly necessary. This need has led to the scientific community focusing their efforts [3][5] on developing efficient and sustainable ways to increase crop yield. To this end, precision agriculture techniques have attracted a lot of attention.

Precision agriculture is a set of methods to monitor crops, gather data, and carry out informed crop management tasks such as applying the optimum amount of water, selecting suitable pesticides, and reducing environmental impact. These methods involve the usage of specialized devices such as sensors, UAVs, and static cameras to monitor the crops. Accurate crop monitoring goes a long way in assisting farmers in making the right choices to obtain the maximum yield. Plant phenotyping, a rapidly emerging research area, plays a significant role in understanding crop-related traits. Plant phenotyping is the science of characterizing and quantifying the physical and physiological traits of a plant. It provides a quantitative assessment of the plant’s properties and its behavior in various environmental conditions. Understanding these properties is crucial in performing effective crop management.

Research in plant phenotyping has grown rapidly thanks to the availability of cost-effective and easy to use digital imaging devices such as RGB, multispectral, and hyperspectral cameras, which have facilitated the collection of large amounts of data. This in ux of data coupled with the usage of machine learning algorithms has fueled the development of various high throughput phenotyping tools [refs] for tasks such as weed detection, fruit/organ counting, disease detection and yield estimation. A machine learning pipeline typically consists of feature extraction followed by a classification/regression module for prediction.

While machine learning techniques have helped build sophisticated phenotyping tools, they are known to lack robustness. They rely heavily on handcrafted feature extraction techniques and manual hyperparameter tuning methods. As a result, if feature extraction is not carefully done under a domain expert’s supervision, they tend to perform poorly in uncontrolled environments such as agricultural fields where factors such as lighting, weather, exposure, etc. often cannot be regulated. Hence, feature extraction from data has been one of the major bottlenecks in the development of efficient high throughput plant phenotyping systems.

Advancements in deep learning, a sub-field of machine learning which allows for automatic feature extraction and prediction on large scale data, has led to a surge in the development of visual plant phenotyping methods. Deep learning is particularly well-known for its effectiveness in handling vision-based tasks such as image classification, object detection, semantic segmentation, and scene understanding. Coincidentally, many of these tasks form the backbone for various plant phenotyping tasks such as disease detection, fruit detection, and yield estimation. Figure 1 illustrates the dierence between machine learning based plant phenotyping and deep learning based plant phenotyping. We believe that the expressive power and robustness of deep learning systems can be effectively leveraged by plant researchers to identify complex patterns from raw data and devise efficient precision agriculture methodologies. The purpose of this survey is to enable the readers to get a bird’s eye view of the advancements in the field of deep learning based plant phenotyping, understand the existing issues, and become familiar with some of the open problems which warrant further research.

2. Background

2.1 Plant Phenotyping

Plant phenotyping is the science of quantifying the physical and physiological traits of a plant. Plant phenotyping mainly benefits two communities: farmers and plant breeders. By better understanding the traits of the crop, a farmer can optimize crop yield by making informed crop management decisions. Similarly, understanding the crop’s behavior is crucial for plant breeders to select the best possible crop variety for a given location and environment. In the past, plant phenotyping was a manual endeavor. The process of manually observing a small set of crop samples and reporting observations periodically was slow, labor intensive and inefficient. The low throughput nature of these methods has impeded the progress in plant breeding research. However, the advent of modern data acquisi-
tion methods with various sensors, cameras and UAVs (Unmanned Aerial Vehicles) coupled with advances in machine learning techniques have resulted in the development of high-throughput plant phenotyping methods to be effectively used for precision agriculture.

Depending on the method of data collection, plant phenotyping techniques can be classified into ground based, aerial and satellite based methods. In ground based phenotyping, high precision sensors are embedded in handheld devices or mounted on movable vehicles to measure useful traits such as plant height, plant biomass, crop development stage, crop yield etc. Figure 6 contracts the discussed classifications. Movable phenotyping vehicles like BoniRob [6] have been developed where RGB cameras, hyperspectral cameras, LIDAR sensors, GPS receivers and other sensors can be mounted. Aerial based methods typically involve the usage of Unmanned Aerial Vehicles (UAVs) for crop monitoring. The recent advancements in UAVs and high resolution cameras have allowed the researchers to obtain high quality crop images. Tasks such as weed mapping, crop yield estimation, plant disease detection and pesticide spraying have been effectively carried out by UAVs. Satellite based plant phenotyping involves remote sensing of agricultural plots from satellites such as Landsat-8 and WorldView-3. Satellite based methods have been typically used for crop health monitoring over a large scale area such as a region/country. However, the cost of obtaining satellite images, the eect of clouds and the time gap between capturing and obtaining images inhibits its applicability for high throughput plant phenotyping in precision agriculture. With a variety of data collection tools at our disposal, large amounts of image and sensor data have been made available for plant
phenotyping research. The next section introduces deep learning, a set of methods which can effectively recognize useful patterns in huge datasets.

### 2.2 Deep Learning

Machine Learning (ML) is a subset of Artificial Intelligence (AI), that deals with an algorithmic approach of learning from observational data without being explicitly programmed. ML has unimaginably revolutionized several fields in the last few decades. Neural Networks (NN) [7][9] is a sub-field of ML and it was this sub field that spawned Deep Learning (DL). Among the most prominent factors that contributed to the huge boost of deep learning are the appearance of large, high-quality, publicly available labelled datasets, along with the empowerment of parallel GPU computing, which enabled the transition from CPU-based to GPU-based training thus allowing for significant acceleration in deep models’ training. Since its redemption in 2006 [10], DL community has been creating ever more complex and intelligent algorithms, showing better than human performances in several intelligent tasks. The deep in deep learning comes from the deep architectures of learning or the hierarchical nature of its algorithms. DL algorithms stack several layers of non-linear information processing units between input and output layer, called Artificial Neurons (AN). The stacking of these ANs in a hierarchical fashion allows for exploitation of feature learning and pattern recognition through efficient learning algorithms. It is proven that NNs are universal approximator of any function [9], making DL task agnostic [11]. Figure 2 depicts the taxonomy of AI.

**Deep learning approaches** may be categorized as follows: Supervised, semisupervised or partially supervised, and unsupervised. Supervised learning techniques use labeled data. In supervised DL, the environment includes sets of input and corresponding output pairs (often in large amounts), a criterion that evaluates model performance at all times called cost or loss function, an optimizing algorithm that minimizes the cost function with respect to the given data. Semisupervised learning techniques use only partially labeled datasets (usually small amounts of label data, large amounts of unlabeled data). The popular Generative Adversarial Networks (GAN) [13] are semi-supervised learning techniques. Unsupervised learning systems function without the presence of labeled data. In this case, the system learns the internal representation or important features to discover unknown relationships or structure within the input data. Often clustering, dimensionality reduction, and generative techniques are considered as unsupervised learning approaches.

### 2.3 Deep Learning for Computer Vision

Convolutional Neural Networks (CNN) is a subclass of neural networks that takes advantage of the spatial structure of the inputs. This network structure was rst proposed by Fukushima in 1988 [15]. It was not widely used then, however, due to limits of computation hardware for training the network. In the 1990s, LeCun et al. [16] applied a gradient-based learning algorithm to CNNs and obtained successful results for the handwritten digit classification problem. CNNs have been extremely successful in computer vision applications, such as face recognition, object detection, powering vision in robotics, and self-driving cars. CNN models have a standard structure consisting of alternating convolutional layers and pooling layers (often each pooling layer is placed after a convolutional layer). The last layers are a small number of fully connected layers, and the final layer is a softmax
classifier as shown in Figure 3.

Every layer of a CNN transforms the input volume to an output volume of neuron activation, eventually leading to the nal fully connected layers, resulting in a mapping of the input data to a 1D feature vector. In a nutshell, CNN comprises three main types of neural layers, namely, (i) convolutional layers, (ii) pooling layers, and (iii) fully connected layers. Each type of layer plays a different role.

(i) Convolution Layers. In the convolutional layers, a CNN convolves the whole image as well as the intermediate feature maps with different kernels, generating various feature maps. Exploiting the advantages of the convolution operation, several works have proposed it as a substitute for fully connected layers with a view to attaining faster learning times. Difference between a fully connected layer and a convolutional layer is shown in Figure 4.

(ii) Pooling Layers. Pooling layers handle the reduction of the spatial dimensions of the input volume for the convolutional layers that immediately follow. The pooling layer does not affect the depth dimension of the volume. The operation performed by this layer is also called subsampling or downsampling, as the reduction of size leads to a simultaneous loss of information. However, such a loss is beneficial for the network because the network is forced to learn only meaningful feature representation. On top of that, the decrease in size leads to less computational overhead for the upcoming layers of the network, and also it works against overfitting. Average pooling and max pooling are the most commonly used strategies. In [18] a detailed theoretical analysis of max pooling and average pooling performances is given, whereas in [19] it was shown that max pooling can lead to faster convergence, select superior invariant features, and improve generalization.

(iii) Fully Connected Layers. Following several convolutional and pooling layers, the high-level reasoning in the neural network is performed via fully connected layers. Neurons in a fully connected layer have full connections to all activation in the previous layer, as their name implies. Their activation can hence be computed with a matrix multiplication followed by a bias offset. Fully connected layers eventually convert the 2D feature maps into a 1D feature vector. The learned vector representations either could be fed forward for classification or could be used as feature vectors for further processing.

Object Detection and Segmentation. Object detection and segmentation are two of the most important and challenging branches of computer vision, which have been widely applied in real-world applications, such as monitoring security, autonomous driving and so on, with the purpose of locating instances of semantic objects of a certain class. In a nutshell, object detection is the task of identifying locating objects (with bounding boxes) in images. While the task of segmentation is to classify each pixel of images with objects (dog, cat, airplane, etc.). We refer readers to [20, 21] for more information on these tasks. Figure 5 visually contrasts the difference between these tasks.

3. Application of Deep Learning in Plant Phenotyping

Figure 4: In a fully connected layer (left), each unit is connected to all units of the previous layers. In a convolutional layer (right), each unit is connected to a constant number of units in a local region of the previous layer. Furthermore, in a convolutional layer, the units all share the weights for these connections, as indicated by the shared linetypes. Figure and description are taken from [17].

Figure 5: Visual illustration of difference between tasks - Image Classification, Object Detection and Instance Segmentation. Example taken from MS-COCO Dataset [22].
3.1 Ground-Based Remote Sensing for Plant Phenotyping

Automation in agriculture and robotic precision agriculture activities demand a lot of information about the environment, the field, the condition and the phenotype of individual plants. An increase in availability of data allowed for successful usage of such robotic tools in real-world conditions. Taking advantage of the available data, combined with the availability of robots such as BoniRob [6] that navigate autonomously in fields, computer vision with deep learning has played a prominent role in realizing autonomous farming. Previously laborious jobs of actively tracking certain measurements of interest such as plant growth rate, plant stem position, biomass amount, leaf count, leaf area, intercrop spacing, crop plant count and others can now be done almost seamlessly.

Crop Identification and Classification. A crucial prerequisite for selective and plant-specific treatments is that farming robots need to be equipped with an effective plant identification and classification system providing the robot with the information where and when to trigger its actuators to perform the desired action in real-time. For example, weeds generally have no useful value in terms of food, nutrition or medicine yet they have accelerated growth and parasitically compete with actual crops for nutrients and space. Inefficient processes such as hand weeding has led to significant losses and increasing costs due to manual labour [23], which is why a lot of research is being done on crop vs weed classification and weed identification [24, 25, 26, 27,28,29] and plant seedlings classification [30, 31]. This is extremely useful in improving the efficacy of precision farming techniques on weed control by modulating herbicide spraying appropriately to the level of weeds infestation.

Crop Detection and Segmentation. Crop detection in the wild is arguably the most crucial step in the pipeline of several farm management tasks such as visual crop categorization [33], real-time plant disease and pest recognition [34], picking and harvesting automatic robots [35], healthy and quality monitoring of crop growing [36] and yield estimation [37]. However, existing deep learning networks achieving state-of-the-art performance in other research fields are not suitable for agricultural tasks of crop management such as irrigation [38], picking [39], pesticide spraying [40], and fertilization [41]. The dominating cause is lack of diverse set of public benchmark datasets that are specifically designed for various agricultural missions. Some of the few rich datasets available are CropDeep [42] for detection, multi-modal datasets like Rosette plant or Arabidopsis datasets [43-45], Sorghum-Head [37],

Figure 6: Top row of (a) shows BoniRob [6] a ground-based remote sensing robot, (b) shows an unmanned aerial vehicle [32], (c) shows a satellite scanning large areas of land respectively. Bottom row across (a), (b), and (c) shows corresponding example images acquired. Satellite Image Credits: NASA.
Wheat-Panicle [46], Crop/Weed segmentation [24], and Crop/Tassel segmentation [47]. Figure 7 contains some examples from the CropDeep [42] dataset. Figure 8 depicts multi-modal annotations provided in the Rosette Plant Phenotyping dataset [43, 44] i.e., annotations for detection, segmentation, leaf center along with otherwise rarely found meta data.

Efficient yield estimation from images is also one of the key tasks for farmers and plant breeders to accurately quantify the overall throughput of their ecosystem. Recent efforts in panicle or spike detection [37, 48-50], leaf counting [51], fruit detection [52] as well as pixel-wise segmentation-based tasks such as panicle segmentation [53, 54] show very promising results in this direction.

**Crop Disease and Pest Recognition.** Modern technologies have given human society the ability to produce enough food to meet the demand of more than 7 billion people. However, food security remains threatened by a number of factors including climate change [55], the decline in pollinators [56], plant diseases [57], and others. Plant diseases are not only a threat to food security at the global scale, but can also have disastrous consequences for smallholder farmers whose livelihoods depend on healthy crops. India loses 35% of the annual crop yield due to plant diseases [58]. In the developing world, more than 80 percent of the agricultural production is generated by smallholder farmers [59], and reports of yield loss of more than 50% due to pests and diseases are frequent [60]. Furthermore, the largest fraction of hungry people (50%) live in smallholder farming households [61], making smallholder farmers a group that’s particularly vulnerable to pathogen-derived disruptions in food supply.

Owing to these factors, timely disease and pest rec-
ognition becomes a priority task for farmers. In addition to that, farmers do not have many options other than consulting other fellow farmers or seeking help from government funded helplines [62]. Availability of public datasets such as PlantVillage [63], PlantDoc [58] allowed for progress in the area of disease and pest detection. Recent research works in pest and insect detection [64-68], invasive species detection in marine aquaculture [69] and disease detection in plant leaves [70-74], Rice [75-77], Tomato [34, 78-80], Banana [81], Grape [82], Sugarcane [83], Eggplant [84], Cucumber [85], Olive [86], Tea [87], Coee [89] and other similar works take encouraging steps towards disease-free agriculture. Figure 9 depicts banana diseases and pest detection outputs from [81]. This work [90] reports solutions to extant limitations in plant disease detection.

3.2 Unmanned Aircraft Vehicles for Plant Phenotyping

The past few decades have witnessed the great progress of unmanned aircraft vehicles (UAVs) in civilian fields, especially in photogrammetry and remote sensing. In contrast with the platforms of manned aircraft and satellite, the UAV platform holds many promising characteristics: flexibility, efficiency, high spatial/temporal resolution, low cost, easy operation, etc., which make it an effective complement to other remote-sensing platforms and a cost-effective means for remote sensing. We refer reader to literary works [91, 92] for the detailed reports of techniques and applications of UAVs in precision agriculture, remote sensing, search and rescue, construction and infrastructure inspection and discuss other market opportunities. UAVs can be utilized in precision agriculture (PA) for crop management and monitoring [93, 94], weed detection [95], irrigation scheduling [96], agricultural pattern detection [97], pesticide spraying [93], cattle detection [98], disease detection [99, 100], insect detection [101] and data collection from ground sensors (moisture, soil properties, etc.,) [102]. The deployment of UAVs in PA is a cost-effective and time saving technology which can help for improving crop yields, farms productivity and protability in farming systems. Moreover, UAVs facilitate agricultural management, weed monitoring, and pest damage, thereby they help to meet these challenges quickly [103]. UAVs can also be utilized to monitor and quantify several factors of irrigation such as availability of soil water, crop water need (which represents the amount of
water needed by the various crops to grow optimally, rainfall amount, efficiency of the irrigation system [104]. In this work [105], UAVs are currently being utilized to estimate the spatial distribution of surface soil moisture high resolution multi-spectral imagery in combination with ground sampling. UAVs are also being used for thermal remote sensing to monitor the spatial and temporal patterns of crop diseases during various disease development phases which reduces crop losses for farmers. This work [106] detects early stage development of soil-borne fungus in UAV imagery. Soil texture can be an indicative of soil quality which in turn influences crop productivity. Hence, UAV thermal images are being utilized to quantify soil texture at a regional scale by measuring the differences in land surface temperature under a relatively homogeneous climatic condition [107, 108]. Accurate assessment of crop residue is crucial for proper implementation of conservation tillage practices since crop residues provide a protective layer on agricultural fields that shields soil from wind and water. In [109], the authors demonstrated that aerial thermal images can explain more than 95% of the variability in crop residue cover amount compared to 77% using visible and near IR images.

Farmers must monitor crop maturity to determine the harvesting time of their crops. UAVs can be a practical solution to this problem [110]. Farmers require accurate, early estimation of crop yield for a number of reasons, including crop insurance, planning of harvest and storage requirements, and cash flow budgeting. In [111], UAV images were utilized to estimate yield and total biomass of rice crop in Thailand. In [112], UAV images were also utilized to predict corn grain yields in the early to midseason crop growth stages in Germany.

There have also been successful efforts that seamlessly combine aerial and ground based system for precision agriculture [113]. With relaxed flight regulations and drastic improvement in machine learning techniques, geo-referencing, mosaicing, and other related algorithms, UAVs can provide a great potential for soil and crop monitoring [114]. More precision agricultural researches are encouraged to design and implement special types of cameras and sensors on-board UAVs, which have the ability of remote crop monitoring and detection of soil and other agricultural characteristics in real time scenarios.

### 3.3 Satellites for Plant Phenotyping

The impact of climate change and its unforeseeable nature, has caused majority of the agricultural crops to be affected in terms of their production and maintenance. With more than seven billion mouths to feed greater demands are being put on agriculture than ever before,
at the same time as land is being degraded by factors such as soil erosion, mineral exhaustion and drought. It becomes the utmost priority for governments to support farmers by providing crucial information about changing weather conditions, soil conditions and more. Currently, satellite imagery is making agriculture more efficient by reducing scouting efforts of farmers, by optimizing use of nitrogen based on variable rate of application, by optimizing water schedules, identifying field performance and benchmark fields, etc [115]. India alone has 7 satellites specially designed for benefits of farmers [116].

Satellites and their imagery are being applied to agriculture in several ways, initially as a means of estimating crop yields [117] and crop types [118], soil salinity, soil moisture, soil pH [119][121]. Optical and radar sensors can provide an accurate picture of the acreage being cultivated, while also differentiating between crop types and determining their health and maturity. Optical satellite sensors can detect visible and near-infrared wavelengths of light, reflected from agricultural land below. It is these wavelengths which combined, can be manipulated to help us understand the condition of the crops. This information helps to inform the market, and provide early warning of crop failure or famine.

By extension, satellites are also used as a management tool through the practice of PA, where satellite images are used to characterise a farmer’s fields in detail, often used in combination with geographical information systems (GIS), to allow more intensive and efficient cultivation practices. For instance, different crops might be recommended for different fields while the farmer’s use of fertiliser is optimised in a more economic and environmentally-friendly fashion. Providing access to satellite imagery also becomes very important for building trust among the involved parties (farmers and government and private bodies involved). Web-based platforms such as Google Earth Engine, Planet.com, Earth Data Search by NASA, LandViewer by Earth Observing System, Geocento [122] and others [123] provide access to past and present (even daily) satellite imagery of your interest.

Agricultural monitoring is also increasingly being applied to forestry, both for forest management and as a way of characterising forests as carbon sinks to help minimise climate change { notably as part of the UN’s REDD programme [124].

4. Plant Phenotyping with Limited Labeled Data

While deep learning based plant phenotyping has shown great promise, requirement of large labeled datasets still remains to be the bottleneck. Phenotyping tasks are often specific to the environmental and genetic conditions, finding large datasets with such conditions is not always possible. This results in researchers needing to acquire their own dataset and label it, which is often a arduous and expensive affair. Moreover, small datasets often lead to models that overfit.

Deep learning approaches optimized for working with limited labeled data would immensely help the plant phenotyping community, since this would encourage many more farmers, breeders, and researchers to employ reliable plant phenotyping techniques to optimize crop yield. To this end, we list out some of the recent efforts in the area of deep plant phenotyping with limited labeled data.

Data Augmentation

The computer vision community has long been employing dataset augmentation techniques to grow the amount of data using artificial transformations. Artificially perturbing the original dataset with artificial transformations (e.g., rotation, scale, translation) is considered a common practice now. However, this approach has some constraints: the augmented data only capture the variability of the 14 available training set (e.g., if the dataset doesn’t include a unique colored fruit, the particular unique case will never be learnt). To overcome this, several data augmentation methods proposed take advantage of recent advancements in the image generation space. In this work [125], the authors use Generative Adversarial Network (GAN) [126] to generate Arabidopsis plant images (called ARIGAN) with unique desirable traits (over 7 leaves) that were originally less frequent in the dataset. Figure 10 (a) shows examples of images generated by ARIGAN. Other latest works [127,128] use more advanced variants of GANs to generate realistic plant images with particularly favorable leaf segmentations of interest to boost leaf counting accuracy of the learning models. In [129], the authors proposed an unsupervised image translation technique to improve plant disease recognition performance. LeafGAN [130], an image-to-image translation model, generates leaf images with various plant diseases and boosts diagnostic performance by a great margin. Two sets of example images generated by LeafGAN are shown in Figure 10 (b). Other data enhancement techniques are also being employed by researchers to train plant disease diagnosis models on generated lesions [131].

The effort to provide finely annotated data has enabled great improvement of the state of the art on segmentation performance. Some researchers have started working on effectively transferring the knowledge obtained from RGB images on annotated plants either to other species or other modalities of imaging. In this work [132], the authors successfully transfer the
knowledge gained from annotated leaves of Arabidopsis thaliana in RGB to images of the same plant in chlorophyll fluorescence imaging.

### Weakly Supervised Learning

Fruit/organ counting is a well explored task by the plant phenotyping community. However, many vision-based solutions have currently require highly accurate instance and density labels of fruits and organs in diverse set of environments. The labeling procedures are often very burdensome and error prone and, in many agricultural scenarios, it may be impossible to acquire a sufficient number of labelled samples to achieve consistent performance that are robust to image noise or other forms of covariate shift. This is why using only weak labels can be crucial for cost-effective plant phenotyping.

Recently, a lot of attention has been placed on engineering weakly supervised learning frameworks for plant phenotyping. In [48], the authors created a weakly supervised framework for the sorghum head detection task where annotators label the data only until the model reaches a desired performance level. After that, model outputs are directly passed as data labels leading to an exponential reduction in annotation costs with minimal loss in model accuracy.

In other work [133], the authors proposed a strategy which is able to learn to count fruits without requiring task-specific supervision labels, such as manually labelled object bounding boxes or total instance count. In [134], the authors use a trained CNN on defect classification data and use it’s activate maps to segment infected regions on potatoes. Segmentation task requires really rich labels (each pixel of the image is annotated) so this task effectively bypasses the labeling for segmentation altogether. On another note, rice heading date estimation greatly assists the breeders to understand the adaptability of the crop to various environmental and genetic conditions. Accurate estimation of heading date requires monitoring the increase in number of rice panicles in the crop. Detecting rice panicles from crop images usually requires training an object detection model such as Faster R-CNN or YOLO, which requires costly bounding box annotations. However, a recently proposed method [49] uses a sliding window based detector which requires training an image classifier, for which annotations are much easier to obtain.

### Transfer Learning

Transfer learning is a type of learning that enables using knowledge gained while solving one problem and applying it to a different but related problem i.e., a model trained on one phenotyping task (say potato leaf classification) being able to assist another phenotyping (tomato leaf classification) task. Transfer learning is a very well explored area of machine learning. As part of the first steps of adopting existing transfer learning techniques for plant phenotyping, the authors of [136] use CNNs (AlexNet, GoogleNet and VGGNet) pretrained on ImageNet dataset [137] and fine-tune on the plant dataset used in LifeCLEF [138] 2015 challenge. With the help of transfer learning, they were able to beat the existing state-of-the-art LifeCLEF performance by 15% points. Similarly in [139], the authors report better than human results in segmentation task with the help of transfer learning where they transfer learn a model trained on peanut root dataset for switchgrass root dataset (they also report results using ImageNet pretrained models). Leaf disease detection and treatment recommendation performance is also shown to be boosted with transfer learning [140]. In [141], the authors interestingly combined a State-of-the-Art weakly-supervised fruit counting model with an unsupervised style transfer method for fruit counting. They used Cycle-Generative Adversarial Network (C-GAN) to perform unsupervised domain adaptation from one fruit dataset to another and train it alongside with a Presence-Absence Classifier (PAC) that discriminates images containing fruits or not and ultimately achieved better performance than fully supervised models.

### Active Learning

Active learning [143], an iterative training approach that
curiously selects the best samples to train, has been shown to reduce labeled data requirement when training deep classification networks [144, 145, 146]. Research in the area of active learning for object detection [147, 148, 149] has been, arguably, limited. However, numerous plant phenotyping tasks such as detection and quantification of crop yield and fruit counting are directly dependent on object detection. Keeping this in mind, an active learning method has been proposed [135] for training deep object detection models where the model can selectively query either weak labels (pointing at the object) or strong labels (drawing a box around the object). By introducing a switching module for weak labels and strong labels, the authors were able to save 24% of annotation time while training a wheat head detection [46] model. Figure 11 illustrates the difference between regular active learning cycle and proposed active learning cycle. This method demonstrates the applicability of active learning to plant phenotyping methods where obtaining labeled data is often difficult. Along the same lines, to alleviate the labeled data requirement for training object detection models for cereal crop detection, a weak supervision based active learning method [142] was proposed recently. In this active learning approach, the model constantly interacts with a human annotator by iteratively querying the labels for only the most informative images, as opposed to all images in a dataset. Figure 12 visually illustrates the proposed framework. The active query method is specifically designed for cereal crops which usually tend to have panicles with low variance in appearance. This training method has been shown to reduce over 50% of annotation costs on sorghum head and wheat spike detection datasets. We expect to see more research works using active learning for limited labeled data based plant phenotyping in the near future.

5. Challenges and Open Problems

In this section, we describe some of the challenges present in plant phenotyping methods which warrant further research.

The Training Data Bottleneck

Modern phenotyping methods rely on deep learning which is notorious for requiring large amounts of labeled data. While some progress has been made in developing data efficient models for phenotyping, reducing the labeling efforts for training efficient phenotyping tools is still an open problem. We believe that effectively adapting techniques from deep learning such as unsupervised, self supervised, weakly supervised, active and semi-supervised learning will greatly benefit the phenotyping community in observing plant traits with small datasets.

Explainability

Deep neural networks are generally considered as black boxes which produce predictions without sufficient justification. This makes debugging a neural network difficult i.e., it can be tough to understand what caused a wrong prediction. Crop management decisions based on incorrect phenotyping results can cause financial losses. Hence, developing explainable models for plant phenotyping is one of the open problems in this field. Obtaining the reasons behind a given set of plant traits using explainable models has the potential to achieve breakthroughs in our understanding of plant behavior.
in various genetic and environmental conditions.

**Data collection**

Vision based plant phenotyping suffers from challenges such as occlusion, inaccuracies in 3D reconstruction of crops and bad lighting conditions caused by the changing weather. It is therefore necessary to develop phenotyping tools which are robust to visual variations.

6. Conclusions

High throughput plant phenotyping methods have shown great promise in efficiently monitoring crops for plant breeding and agricultural crop management. Research in deep learning has accelerated the progress in plant phenotyping research which resulted in the development of various image analysis tools to observe plant traits. However, wide applicability of high throughput phenotyping tools is limited by some issues such as 1) dependence of deep networks on large datasets, which are diicult to curate, 2) large variations of field environment which cannot always be captured, and 3) capital and maintenance which can be prohibitively expensive to be widely used in developing countries. With many open problems in plant phenotyping warranting further studies, it is indeed a great time to study plant phenotyping and achieve rapid progress by utilizing the advances in deep learning.

**References**


[63] Hughes, D., Salathe, M.: An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing. (11 2015)


[67] Liu, T., Chen, W., Wu, W., Sun, C., Guo, W., Zhu, X.: Detection of aphids in wheat fields using a computer vision technique. Biosystems Engineering 141 (01 2016) 82-93


[91] Shakhatreh, H., Sawalmeh, A.H., Al-Fuqaha, A., Dou,


cal plant disease diagnosis. (2020)


Akshay L Chandra completed his bachelor's degree in Computer Science and Engineering from Jawaharlal Nehru Technological University in 2017. He also holds a PG Diploma in Applied Statistics from IGNOU. Currently, he is working as a research assistant at Dr. Vineeth N Balasubramanian's research lab at IIT Hyderabad. Prior to joining IIT-H, he spent 16 months working in a software firm providing machine learning solutions to clients in healthcare and e-commerce domains. His research interests broadly lie in the fields of computer vision, machine learning and learning with limited labeled data, mainly deep active learning.

Sai Vikas Desai received the BTech degree in Electronics and Communication Engineering from Jawaharlal Nehru Technological University in 2017. He is working toward the Masters degree in Computer Science at Indian Institute of Technology Hyderabad, under the supervision of Dr. Vineeth N Balasubramanian. His research interests include deep learning, computer vision and plant phenotyping. Specifically, he is interested in developing active learning techniques for training efficient deep neural networks with limited labeled data for agricultural and plant phenotyping applications.

Dr Vineeth N Balasubramanian is an Associate Professor in the Department of Computer Science and Engineering at the Indian Institute of Technology, Hyderabad (IIT-H). His research interests include deep learning, machine learning, and computer vision. His research has been published at premier peer-reviewed venues including ICML, CVPR, ICCV, KDD, ICDM, IEEE TPAMI and ACM MM. His PhD dissertation at Arizona State University on the Conformal Predictions framework was nominated for the Outstanding PhD Dissertation at the Department of Computer Science. He is an active reviewer/contributor at many conferences such as NeurIPS, CVPR, ICCV, AAAI, IJCAI, ACM MM with a recent award as an Outstanding Reviewer at CVPR 2019, as well as journals including IEEE TPAMI, IEEE TNNLS, JMLR and Machine Learning. He currently serves as the Secretary of the AAAI India Chapter. For more details, please see https://iith.ac.in/~vineethnb/.

Dr. Wei Guo is currently an Assistant Professor with International Field Phenomics Research Laboratory, Institute of Sustainable Agro-Ecosystems, Graduate School of Agricultural and Life Sciences, the University of Tokyo, Japan. He received his Ph.D in Agri-Informatics from the University of Tokyo, in 2014. His research focus on field-based high-Throughput phenotyping using advanced imaging techniques (drone remote sensing and ground-based observation), image analysis and machine learning approaches. He also serves as the Associate Editor of a science partner journal, Plant Phenomics; and Executive Board Member, Editorial Committee of Japanese Society of Agricultural Informatics.

POST – QUANTUM CRYPTOGRAPHY

P. V. ANANDA MOHAN
Life Fellow IEEE

1. Introduction

Traditionally, information security needed encryption, authentication, key management, non-repudiation and authorization which were being met using several techniques. Standardization of algorithms by National Institute of Standards and Technology (NIST) has facilitated international communication for banking and information transfer using these standards. Encryption can be carried out using Advanced Encryption Standard (AES) using variable block lengths (128, 192 or 256 bits) and variable key lengths (128, 192 or 256 bits). Solutions for light weight applications such as those for Internet of Things (IoT) are also being standardized. Message integrity is possible using host of hash algorithms such as SHA-1, SHA-2 etc., and more recently using SHA-3 algorithm. Authentication is possible using well known Rivest-Shamir-Adleman (RSA) algorithm needing 2048/4096 bit operations. Elliptic Curve Cryptography (ECC) is also quite popular and used in several practical systems such as WhatsApp, BlackBerry etc. Key exchange is possible using Diffie-Hellman algorithm and its variations. Digital Signatures can be carried out using RSA algorithm or Elliptic Curve Digital Signature Algorithm (ECDSA) or DSA (Digital Signature Algorithm). All these algorithms derive security from difficulty in solving some mathematical problems such as factorization problem or discrete logarithm problem. Though published literature gives evidence of solving factorization problem upto 768 bits only, it is believed that using Quantum computers, these problems could be solved by the end of this decade. This is due to availability of the pioneering work of Shor and Grover [1]. For factoring an integer of N bits, Shor’s algorithm takes $O(\log N(\log \log N)(\log \log \log N))$ quantum gates. As such, there is ever growing interest in being ready for the next decade with algorithms that may resist attacks in the quantum computer era. NIST has foreseen this need and has invited proposals from researchers all over the world. In the first round, about 66 submissions were received which have been scrutinized for completeness of submissions, novelty of the approach and security and 25 of these were promote to second round to improve based on the comments received on the first round submission. These will be analyzed for security and some will be selected for final recommendation for use by industry. These are for encryption/decryption, key agreement, hashing and Digital Signatures for both hardware and software implementations. In this paper, we present a brief survey of the state of the art in post-Quantum Cryptography (PQC) followed by study of one of technique referred to as Learning With Errors (LWE) in some detail.

2. PQ algorithms

There are five areas into which the 26 second round PQ algorithms can be classified. These are as follows:

(a) Code based Cryptography

(b) Ring Learning with Errors

(c) Isogeny on Elliptic curves

(d) Multivariate Quadratic equations

(e) Lattice based cryptography

<table>
<thead>
<tr>
<th>Public Key Encryption (PKE) Key Encapsulation Mechanism (KEM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>code based</td>
</tr>
<tr>
<td>BIKE</td>
</tr>
<tr>
<td>Classic McEliece</td>
</tr>
<tr>
<td>HQC</td>
</tr>
<tr>
<td>LEDAcrypt (merger of LEDAkem/LEDApkc)</td>
</tr>
<tr>
<td>NTS-KEM</td>
</tr>
<tr>
<td>ROLLO (merger of LAKE/LOCKER/Ouroboros-R)</td>
</tr>
<tr>
<td>RQC Code based</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Digital Signatures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lattice based</td>
</tr>
<tr>
<td>CRYSTALS-DILITHIUM</td>
</tr>
<tr>
<td>FALCON</td>
</tr>
<tr>
<td>qTESLA</td>
</tr>
</tbody>
</table>

Table I. Second round submissions of PQC competition of NIST

In this article, we focus on Ring-LWE (Learning with errors) based Encryption, Key exchange and KEM operations using one of the Post Quantum cryptography entries New Hope.

3. Learning with Errors (LWE)

This is based on the difficulty of finding a solution for the shortest vector problem in a lattice. Given a number of independent vectors $b_1, b_2, ..., b_n$, the lattice is defined as $L=a_1b_1+...+a_nb_n$ for all $a_i$ belonging to set $Z$. The set $\{b_1, ..., b_n\}$ is called a basis of the lattice.
We first describe integer mode LWE and then we deal with Ring-LWE. Initially we create a secret key value \( s \) and another value \( e \). Next we select a number of values \( \langle A, 3 \rangle \) and calculate \( B[] = \langle A, 5 \rangle \times s + e \). The values of \( \langle A, 0 \rangle \) and \( B[] \) become our public key. If \( s \) is a single value, \( A \) and \( B \) are one dimensional matrices. Note that \( e \) is error column matrix. The difficult problem to be solved is given \( A \) and \( B \), findings. This called search problem. Given enough samples, can we find the secret? On the other hand, there is another problem—decision—can we distinguish a random sample and \( m \) independent samples of LWE. The decision problem wants to find whether there is a secret first of all in the samples. Note that without the error vector \( e \) (i.e. \( e = 0 \)), we can easily solve the system of linear equations. In the presence of errors with some distribution say discrete Gaussian distribution (mean zero and standard deviation \( \sigma \)), solving the system is believed to be quantum hard. Note that error distribution in fact hides the characteristics of the plain text. Consider an example of small dimensions chosen prime \( p = 13 \):

\[
\begin{array}{c|c|c|c|c}
4 & 1 & 1 & 1 & 0 \\
5 & 5 & 9 & 5 & 9 \\
3 & 9 & 0 & 10 & -1 \\
1 & 3 & 3 & 2 & 11 \\
1 & 2 & 7 & 3 & 4 \\
6 & 5 & 1 & 1 & 4 \\
3 & 3 & 5 & 0 & -1 \\
\end{array}
\]

Finally, we get from the last row, the result polynomial as \( A.s + e = 10 + 5.x + 10.x^2 + 7.x^3 \)

Evidently \( n^2 \) modulo multiplications and addition of all these products together with \( e \mod q \) is required.

5. Key Exchange using R-LWE

We next consider Key exchange using R-LWE which is similar to Diffie–Hellman Key exchange. We use subscripts \( A \) and \( B \) to indicate their respective parameters. We consider Alice and Bob as the two sharing the key and describe the steps involved.

(a) Alice chooses \( A \) and secret key \( s \) and error polynomial \( e \) and computes \( b^A = A.s + e \), and shares \( A \) with Bob and sends \( b^A \) to him.

(b) Bob selects a secret key \( s^B = s \) and error vector \( e' = e \) and computes \( b^B = A.s + e \) and sends to \( A \).

(c) Alice now multiplies what is received with secret key which she knows \( s_A \) to get \( s_A.b^B \).

(d) Bob also multiplies \( b^A \) with \( s_B \) to get \( s_B.b^A \).

(e) Note that the shared key is \( s_A.b^B = s_B.b^A \).

Note that all the computation is done in ring of polynomials. Since Alice and Bob use different error vectors, the shared keys will have different rounding errors. This can be resolved by using a process called ‘reconciliation’. In this method, one user sends a hint to the other and both Alice and Bob can use the hint to arrive at the correct shared key.

6. Public Key Encryption

We next describe how a message can be encrypted. We generate 20 random numbers for our public key \( A \). We also generate an error vector \( e \) of 20 samples and we choose a prime \( q = 97 \). WE aso choose a secret key \( s = 5 \). We will compute \((A \times s)\) first and reduce \( (x^4 + 1) \).
e = [3, 3, 4, 1, 3, 4, 4, 1, 4, 3, 3, 2, 2, 3, 2, 4, 4, 1, 3]

Next we compute B = (A+e) mod q as shown below:

B = [15, 45, 2, 20, 13, 30, 32, 45, 4, 3, 34, 78, 55, 51, 23, 67, 44, 34, 17, 75]

Not that B is also public key. Now we can distribute lists A and B to anyone. Suppose we want to encrypt a message bit m.

We pick from our lists A and B few samples and compute u and v as follows:

\[ u = \sum (A_{samples})(mod q) \] and \[ v = \sum (B_{samples})+(q/2)m(mod q) \]

The encrypted message is \((u,v)\). To decrypt, we calculate:

\[ \text{Dec} = v - su \mod q \]

If Dec is less than \(q/2\), the message is a zero, else it is a 1.

For our example we choose from A and B the entries in the five locations \([18, 5, 8, 13, 11]\) and compute u and v as:

\[ u = 34 \] and \[ v = 83 \]

The encrypted message is \((34, 83)\). We get Dec = \((83-5 \times 34) \mod 97\) = 10. Since Dec is less than \(q/2\), we get \(m = 0\). The reader may note however that just for sending one bit we have to transmit two numbers of size q.

7. R-LWE Implementation

In practice R-LWE encryption has three steps: Key generation, Encryption and Decryption. These are as follows:

**KeyGen (a):** Choose two polynomials \(r_1\) and \(r_2\) and compute \(p = r_1^2 - ar_2\). The public key is \((a, p)\) and private key is \(r_2\), the polynomial \(r_1\) is noise and is no longer required after key generation.

**Encrypt (a,p,m):** The message \(m\) is encoded as a polynomial \(m'\) in the ring. Three polynomials \(e_1, e_2, e_3\) are sampled from a distribution. Compute \(c_1 = ae_1 + e_2\) and \(c_2 = pe_1 + e_3 + m'\). The final cipher text is \((c_1, c_2)\).

**Dec(c1,c2,r2):** Compute \(m' = c_1r_2 + c_2\) and recover the original message.

The message is encoded first by writing 1 as \((q-1)/2\) and retaining ‘0’ as ‘0’. The decoding is done y following the rule:

if \((q-1)/4 \leq m' \leq (3q-1)/4\), return 1, otherwise return 0.

Note that \(e_1, e_2, e_3\) will not affect the decrypted text if \(a, q, e, r_1\) and \(r_2^2\) are properly chosen.

Four primitives will be needed in the implementation:
(a) Discrete Gaussian sampler
(b) Polynomial multiplier
(c) Countermeasures against side channel attacks
(d) True Random number generator.

Different approaches may be used in different algorithms regarding these four aspects. We consider for illustration New Hope next.

8. New Hope

With the above background, we describe briefly New Hope Key Encapsulation Mechanism (KEM) and public key encryption (PKE) which uses lot of steps in the generation of A, s and e for meeting the security requirements. Two versions of New hope one with N = 512 and another with 1024 are available. The prime q chosen for N=1024 is 12259. Evidently this means that all coefficients in the 1024 degree polynomials are of 14 bit length. Now the reader may get an idea of the key length needed. The Table II gives the parameters recommended. CPA and CCA mean Chosen Plain Text attacks and Chosen Cipher Text attacks.

Table II: Two versions of New hope one with N = 512 and another with 1024 are available as shown.

<table>
<thead>
<tr>
<th>Parameter Set</th>
<th>pk</th>
<th>sk</th>
<th>ciphertext</th>
</tr>
</thead>
<tbody>
<tr>
<td>NewHope512-CPA-KEM</td>
<td>928</td>
<td>869</td>
<td>1088</td>
</tr>
<tr>
<td>NewHope1024-CPA-KEM</td>
<td>1824</td>
<td>1792</td>
<td>2176</td>
</tr>
<tr>
<td>NewHope512-CCA-KEM</td>
<td>928</td>
<td>1888</td>
<td>1120</td>
</tr>
<tr>
<td>NewHope1024-CCA-KEM</td>
<td>1824</td>
<td>3680</td>
<td>2208</td>
</tr>
</tbody>
</table>

Table II: Two versions of New hope KEM for CPA and CCA

| Parameters: \(q = 12289 > 2^{16}, n = 1024\) |
| Error distribution: \(\psi_{16}\) |

In the above code, the symbol \(\psi\) stands for distributions from which the coefficients of various polynomials are taken. The HelpRec step helps for reconciliation so that both the users arrive at the same key. The parameter \(\mu\) indicates the final key obtained by hashing v. Since the polynomials are of degree 1024, multiplication in the ring with coefficients reduced mod q is a complex process. This can be solved by using Number theoretic transforms (NTT). Both the polynomials A and B to be multiplied mod \((x^n + 1)\) are transformed using NTT and the resulting 1024 values can be multiplied point-wise and inverse NTT can be applied to get the actual product. This needs initial transformation of the polynomial coefficients of A and B using a primitive root \(\omega\) defined as \(\omega N = 1 \mod p\). As an illustration, for \(N = 1024\), \(p = 12277\), we have \(\omega = 49\). Note that New Hope uses...
SHAKE256 hash function (SHA3 variation) for various functions, which takes a seed of 256 bits and outputs several bytes as desired.

**Conclusion**

Post Quantum Cryptographic (PQC) algorithms are more difficult to understand than conventional cryptographic algorithms like AES, RSA etc. It may be noted that while quantum computers need different environment, PQC algorithms will still be implemented in FPGAs, GPUs and embed processors and ASICs. Attempts also have been made to incorporate suit of PQC algorithms in Web browsers and other applications even in IoT devices with limited resources.

**References**


**P. V. Ananda Mohan**

Dr. P. V. Ananda Mohan is Technology Advisor, Centre for Advanced Computing (CDAC), Bangalore. He has received Ph.D degree in electrical communication engineering from Indian Institute of Science, Bangalore, in 1975. From 1973 till December 2003, he was with I.T.I. Limited R&D Division. Later, till 2014, he was with Electronics Corporation of India Limited, Bangalore. His research interests are in the area of Analog VLSI design, VLSI architectures and Cryptography. He has published in these areas in refereed international journals and conferences. He has published five books Switched Capacitor Filters: Theory, Analysis and Design.
INTEGRATING PHYSICAL, CYBER AND THE BIOLOGICAL WORLDS

An interview with Prof. Jan Rabaey, Professor, EECS, Berkeley University

DOI: https://doi.org/10.34048/2020.12.F4

Prof. Debabrata Das of IIIT- Bangalore engages in a conversation with Prof. Jan Rabaey, Professor, EECS, Berkeley University, in an interview recorded during Prof. Rabaey’s recent visit to India. The two discuss the future of wireless research and emerging opportunities.

Prof. Jan Rabaey, is the founding director of the Berkeley Wireless Research Center (BWRC) and the Berkeley Ubiquitous SwarmLab. He has made significant contributions in advanced wireless networks, low-powered integrated circuits, sensor networks and ubiquitous computing.

Excerpts from the interview.

Prof. Rabaey on his research path

I’ve been on the faculty in Berkeley for more than 30 years and my area of interest has always been on the impact of wireless in general on how we as a society will connect together. So let me go back a little bit in time to the early 90s when it became clear that wireless technology would become very important and ubiquitous. A colleague and I set out to solve the question: Suppose you have ubiquitous wireless connectivity and if wireless would be everywhere, how would you build a computer? Would you still build a computer? And the answer we came up was no you don’t build computers anymore, you build user interface devices. Computing will be done somewhere else and you carry the device with you, a portable device, that allows you to interact with the computing in the background. We called that project InfoPath. It was a pad-like device with pen and voice input. Remember this was 1990, a time you had no cellular phones yet. The best wireless technology available was 802.11 that gave us about 1 megabits per second data rate. We built this device to show that indeed it was plausible and as always technology takes time to pick up and before it comes to the marketplace. But obviously 10 years later we got iPads and we got smartphones and all those kind of things. My quest always has been whether I can make this thing smaller, more effective, more ubiquitous. From there on, I move to the domain of what we now call wireless sensor networks. If I say I can make my radio small, I can build a node that has a computer as a wireless interface, has some sensors and connect them all together in a large network. We have a set of projects that you can indeed build something like that at a cubic centimetre and then later scale it down to smaller and smaller sizes. This led to things like the Internet of Things or industrial 4.0, the idea that we are going to sprinkle the world with sensors that are connected and connect information – connect the cyber world to the physical world around us. That was the next step.
Then around mid-2000s, we said okay what happens if we go from a cubic centimeter to a cubic millimeter? What could I do at that point in time? That brought me into the whole sphere where I can have electronics talk directly to biological cells, could be neurons or it could be electrical fields in your body, variable devices of all styles. So it’s kind of been a progression of going to smaller and smaller and lower and lower power devices but ultimately extending the application scope initially from pure communication or data gathering to cyber-physical systems ultimately to cyber biological systems. That’s kind of the road that I’ve been following over the years.

Then around mid-2000s, we said okay what happens if we go from a cubic centimeter to a cubic millimeter? What could I do at that point in time? That brought me into the whole sphere where I can have electronics talk directly to biological cells, could be neurons or it could be electrical fields in your body, variable devices of all styles. So it’s kind of been a progression of going to smaller and smaller and lower and lower power devices but ultimately extending the application scope initially from pure communication or data gathering to cyber-physical systems ultimately to cyber biological systems. That’s kind of the road that I’ve been following over the years.

**Integrated system of wireless devices. The seamless integration of Physical world, Cyber world and the Biological world**

In the past we were always thinking about individual devices with a particular function and the function was in the device. With wireless technology it is possible to have a broad range of devices that are scattered around in the environment to form a network and the function is not anymore in the devices but is in the global network itself. The network provides the function; it provides capabilities of gathering data of different types and nature but also from different places; combine it together either centralized or in a distributed fashion; reason about it; think about it and act so we are actually starting to make intelligent networks. This is where we’re going, for instance with 5G as the next step. 5G promises two things. On one side it’s going to give us higher data rates — remember the one megabits per second of 1990s, now we’re talking about 10 gigabits per second. The types of information you can present and transmit out is going to be so humongously large that I can start thinking about immersive worlds, real augmented reality and all those type of things are going to be possible. On the other side, the capability of connecting millions to billions of devices together allows you to think about a world that is completely filled up with devices connected together, giving us a direct link between the physical world around us and the cyber world which used to be separate; they become one thing; they’re becoming indistinguishable. People talk about digital twins, they are possible. This gigantic network of devices together and their ubiquitousness are giving us new functionality, new capabilities and you go one step beyond to the biological world. With the sensors on our body or inside our body we should be able to get a precise vision on how our body is operating, helping us to correct things that are wrong. It will give us more functionality and capabilities that we don’t have today, you call it augmentation but this will happen as a result of this gigantic wireless connected network. We’re not there yet, we have to think 10-15 years forward but this to me something that is clearly on the roadmap.

**On brain machine interface**

The first set of sensors that people have been building was about the parametric behavior which measures heart rate and other things and you figure out how things go. If I can make my electronics small enough about the size of a biological cell, I should also have the capability of creating a direct interface between a neuron and an electronic device. I can have neurons in our brain talk to electronic devices; transmit the information out and use that to do certain functions that I can describe in a minute. The idea of brain machine interface is to create a channel between our biological computer that we have in our head, create a direct channel to read out information or to write information back. There’s a whole bunch of medical reasons that make this important such as lost motor capabilities because of accidents, spinal cord injuries, which snaps communication between the limbs and the brain. If I can read the signals in the motor cortex and transmit them out to either stimulate the limbs or drive a prosthetic device, I can give motion back to people who lost it. Same thing for instance with people who have Lou Gehrig’s disease where the ends of nerves die off and basically the brain is now left disconnected from the rest of the body. Prof. Stephen Hawking suffered from it. We could have had an interface that latched on to this speech region, motor region of the brain; capture those signals and drive them into speech synthesizers; we can make people with those type of disease speak again and communicate with the rest of the world. Similar is true for mental diseases, post-traumatic depression, all those type of things could be addressed if I have effective brain machine interfaces.

We have demonstrated on primates [in the lab] as well as humans, this technology is indeed feasible. However, the technology today is too bulky, it doesn’t last very long, we don’t want to have a cable coming out of the head while the person walks around on the street. We have to make sure that it is a technology we can implant; that can last for at least 10 years; is powered not by a battery and it has to have a wireless interface. All those factors together require another form factor in shrinkage, reliability, effectiveness. These things need
to happen before we have effective brain machine interfaces that humans can carry with them for tens of years and again plenty of research still to be done but the roadmap is clear.

On the evolutionary cycle of hardware

Things go in cycles. Hardware is actually in the up-swing again. It’s getting more important and the main reason for that is indeed things like IoT, biomechanical interface, self-driving cars, all those type of things are built on innovative hardware platforms. One can argue that Moore’s law is ending and transitional scale does not matter anymore. Ultimately what we're going to be doing over the next couple of decades is build more sophisticated device, devices that have more functionality, better sensing capability, computing, communication, energy harvesting, storage, translation and all in smaller and smaller form factors and higher efficiency. We might see devices that directly communicate with a biological tissue where you interweave biology and physical elements together.

There’s no question in my mind that hardware design capabilities are going to keep on going very strongly over the next couple of decades. It will be different. It will not be the same thing as building digital circuits by putting gates together. It’s broader where innovation will reign and how we bring things together going into third dimension, say, for example, stack memory on [top of] logic [circuit] and sensor, interweave them together. These are the type of things we need to think about and that’s going to require new methodologies, new ways of production, manufacturing, testing and verification. All those type of things are going to be challenged, so there’s plenty of things to be done. I’m a strong believer that hardware will shine in the next decades.

On low-power devices

One way I’ve described my research over the years is how low can you go which is sometimes not the right thing to think about but there’s still plenty of opportunity. Today we know what the limits are, how deep you can go and we still away 4 to 5 orders of magnitude from where we could be. Nature has found some ways to get closer. If I look at biology, some of our cells in the synapses in the brain work at about two orders of magnitude closer than the absolute limit, which means we should be able to get close to that model. There's plenty of opportunity to rethink computing so that we get actually more efficient, there's plenty of place to go but it's not going to be with our traditional, determinist computing instruction set style, machine learning devices. Its not going to be easy.

Prof. Debabrata Das

Dr. Debabrata Das is Professor at IIIT-Bangalore (IIIT-B). Before joining IIIT-B, he had served at G S Sanyal School of Telecommunication at IIT Kharagpur and later at Kirana Networks in New Jersey, USA. At present, he is Principal Investigator (PI) of projects from Department of Electronics and Information Technology, Government of India on Green Broadband Wireless Network and Nokia Research on IoT. He was PI of sponsored projects from Intel, Hewlett Packard, Microsoft, Motorola Research, Nokia, Govt. of India on areas of IMS and Broadband Wireless MAC/QoS/Energy-saving, TVWS and Security. He has more than 125 peer reviewed papers in different journals and International conferences. He has 1 accepted US patent and 10 more are under review. Dr. Das received his Ph.D. degree from IIT-Kharagpur and MTech from IIT-Varanasi. He is a Board Member of IIIT-Bhubaneswar; Technical and Empower Committee member of e-Governance, Govt. of Karnataka. He is Fellow of Institution of Electronics & Telecommunication Engineers (IETE) and Fellow-IE. Dr. Das is recipient of Global IEEE MGA Achievement Award 2012 and Prof. K Sreenivasan memorial outstanding teaching award in the areas of Electronics and Telecommunication in 2017.
EXPERIENTIAL LEARNING OF NETWORKING TECHNOLOGIES

Evolution of Socket Programming – Part II

RAM P. RUSTAGI, Department Of CSE, KSIT Bengaluru,
VIRAJ KUMAR Divecha Centre For Climate Change, IISc Bengaluru

1. Introduction

In our last article [1], we discussed the concept of a perpetual server handling multiple concurrent clients and explored several design/implementation choices. These included creating one child process (or thread) per client connection, pre-creating a number of children processes (and/or threads) and assigning a new TCP connection to one of the free child process/thread and a single threaded server using the socket API call select()[2][3]. We briefly discussed the latter approach using select(), in which, the process maintains a list (array) of open sockets, and each time whichever socket is active, the process uses that socket. For example, when data is available to read from the socket or the socket has enough buffer for the data to be written into, then the network stack will be transmitted subsequently.

The approach of using select() performs better than other approaches of creating multiple threads/processes. As discussed in [1], one of the limitations of using select() is that it is generally limited to handling 1024 concurrent connections. This number turns out to be quite insufficient today as modern servers need to deal with many more than 1024 concurrent connections [4]. There are other inefficiency issues when using select() especially on account of frequent switching between user space and kernel space execution of a process. In this article we focus on these limitations of select() and alternatives available (such as poll()) [5], and epoll() [6]) to improve upon the performance.

2. Understanding use of socket API select()

Network (Socket) programming was first implemented with BSD Unix and C, which was the primary system development language at the time. Thus, the socket APIs implementation and man pages followed C function specifications. To understand performance issues associated with select, let us first understand its working. The man page for select() specifies its usage as:

```
int select(int nfds,
    fd_set *readfds,
    fd_set *writefds,
    fd_set *exceptfds,
    struct timeval *timeout);
```

The first argument corresponds to maximum value of a file descriptor (fd) among all file descriptors (fds) in which the invoking program is communicating. In Unix, every I/O is implemented via an fd (irrespective of terminal I/O, disk I/O, network I/O etc.). The remaining three parameters correspond to file descriptors represented as bit values in a bit vector. This bit vector is of size 1024 bits (by default), and a change may require recompiling the Linux kernel with higher size. Each file descriptor (fd) is represented by a bit value of 1 at a bit position as per the value of fd. For example, if fd value is 100, then bit number 100 corresponds to this fd value 100. Thus, a process needs to set respective bits for all those sockets on which it is connected via TCP to other processes. For example, consider that a server process has established connections with 10 clients (C_1 to C_{10}) and all are communicating. Thus, initially, in general, fds corresponding to C_1, C_2, ..., C_{10} would be 3, 4, ..., 12. The fd numbers 0, 1, 2 being used for standard input, standard output and standard error. Now consider that clients C_2, C_6, C_7 and C_9 have closed connections and thus remaining 6 connected clients are C_1, C_3, C_4, C_5, C_6, C_10. Thus, active fd numbers for this server process would be 3(C_1), 4(C_2), 6(C_3), 7(C_4), 10(C_5), 12(C_{10}). For these open sockets, the fdset value be 0x1478 (0b1010011101000) would be used in socket call as bits having value 1 corresponds to bit positions 3, 5, 6, 7, 10, and 12. The invoking process will pass address (reference) of fdset having value 0x1478 to select() call, informing the kernel about the fds in which this process is interested i.e. would like to process data on these fds. Consider that only the clients C_5 and C_{10} have sent data and other clients C_1, C_3, C_4, and C_6 are idle. Thus, data from C_5 and C_{10} needs to be read and acted upon by the server process. The kernel needs to convey only these sockets of interests upon return by select() call. The kernel implements this by retaining the corresponding bits in fdset (i.e. bit number 7 and 12) and resetting all other bits. Thus, upon return fdset (fdset is passed by reference in select() ) vector is modified, and select returns the value of fdset as 0x1020 (representing fd number 12 and 7 for clients C_5 and C_{10}). The calling process then acts upon these sockets and reads data (or write) on these two fds, and provides response to clients C_5 and C_{10}.

With this basic understanding of select(), let us analyse it from the performance perspective. For illustration purpose, we will consider the 2nd parameter readfds in select() usage, which corresponds to list of fds on which the process can receive the data, and same performance perspective will be applicable to writefds and exceptfds. Additionally, the fdset readfds also contains one special fd corresponding to the listening socket meant for accepting a new connection from some client. When server process invokes the select() call, for each open socket fd, it sets the respective bits in readfds. While the process code executes in user space, the select call is executed in kernel space and this fdset buffer is copied to kernel buffer, and scanned in kernel space execution so that underlying network implementation can identify the fds which needs to be checked for any activity.
i.e. if data has been received from client process. Then, for each of those fds on which some event has occurred (e.g. data has been received from the client), kernel sets the bits for these fds and resets the remaining bits in fdset readfds. The kernel achieves this by monitoring the state of each of fds. Thus, till the data is read fully on an fd by the server process invoking the select(), the kernel will always set the bit on its return from select() call for an active fd i.e. which has data in kernel buffer that process needs to read. This can be better understood by an example. Consider that a server is connected to N clients (C_1, C_2, ..., C_N) and client C_x sends some data (N_x bytes) at time T_y corresponding to fd value x, and another client C_y sends some data (N_y bytes) at time T_z (>T_y) corresponding to fd value y, and server process invokes select call at time T_w (>T_z) with bit value as 1 for all socket fds corresponding to these N clients including those bit number x and y. Upon execution of the select(), it will return fdset value with bit value as 1 for both fds x and y and all other N_z bits as zero. Consider that server process performs a read operation on socket x and reads all the N_x bytes, and another read operation on fd y but reads only M_y (<N_y) bytes and then invokes select() call again. The kernel knows that there is still (N_y-M_y) bytes of data yet to be read on socket fd y and thus it will return the yth bit as 1 in readfds and other bit values as 0.

When the returned value of readfds contains the bit corresponding to listening socket, it implies a new connection and process should execute accept() to establish a new TCP connection, which results in a new socket fd and same should be added to list of open fds. Similarly, when an existing connection is closed, the corresponding bit in fdset should be set as the value 0.

The 3rd parameters of select() corresponds to writeable fds i.e. it contains fds into which the program would like to send(write) the data. The program would like to write the data when socket is writable i.e. there is memory buffer available in the kernel networking stack for the specified socket. The buffer space may become full when the client at other end of socket is not reading any data (manifested as TCP Zero Window phenomenon). In such a situation, if the program writes the data into the socket, the program would be blocked by the kernel till some buffer becomes available. This blocking of the process would have repercussions on communications with other client applications thereby impacting the performance.

Similarly, the 4th parameter in select() corresponds to exceptional fds, which generally consists of all open fds being used by the program. When the network stack encounters any errors on an fd, for example, the other end has suddenly closed the connection and thus this socket is closed by the network stack, then this is indicated by setting the corresponding bit in the exceptional fdset exceptfds. This fdset provides a mechanism for the underlying network to notify the invoking application about any exception/error events other than read and write that occurs on the socket.

### 3. Performance Aspects of select() API

With the above mentioned description of basic operation of select() call, below we analyze its performance related aspects

#### 1. Number of concurrent connections

Today’s internet server needs to serve a large number of

<table>
<thead>
<tr>
<th>Table 1 : Snippets of socket program using select()</th>
</tr>
</thead>
<tbody>
<tr>
<td>01: fd_set rset; /* master read fdset */</td>
</tr>
<tr>
<td>02: fd_set wset; /* master write fdset */</td>
</tr>
<tr>
<td>03: fd_set wk_rset; /* working read set to be passed to select*/</td>
</tr>
<tr>
<td>04: fd_set wk_wset; /* working write set to be passed to select */</td>
</tr>
<tr>
<td>05: int listenfd,</td>
</tr>
<tr>
<td>06: int connfd,</td>
</tr>
<tr>
<td>07: int maxfd;</td>
</tr>
<tr>
<td>08:</td>
</tr>
<tr>
<td>09: listenfd = socket(AF_INET, SOCK_STREAM, 0);</td>
</tr>
<tr>
<td>10: bind((listenfd, (struct sockaddr *) &amp;servaddr, sizeof(servaddr));</td>
</tr>
<tr>
<td>11: listen(listenfd, LISTEN);</td>
</tr>
<tr>
<td>12: FD_SET(listenfd, &amp;rset);</td>
</tr>
<tr>
<td>13: max_fd = listenfd + 1;</td>
</tr>
<tr>
<td>14: cnt = select(max_fd, &amp;wk_rset, &amp;wk_wset, NULL, &amp;timeout);</td>
</tr>
<tr>
<td>15: if (FD_ISSET(listenfd, &amp;wk_rset)) {</td>
</tr>
<tr>
<td>16: connfd = accept(listenfd, (struct sockaddr *) &amp;cliaddr, &amp;clielen);</td>
</tr>
<tr>
<td>17: FD_SET(connfd, &amp;rset); FD_SET(connfd, &amp;wset);</td>
</tr>
<tr>
<td>18: max_fd &lt;= connfd: connfd + 1 ? max_fd</td>
</tr>
<tr>
<td>19: }</td>
</tr>
<tr>
<td>20: for (int ifd = listenfd+1; ifd &lt; max_fd; ifd++) {</td>
</tr>
<tr>
<td>21: if (FD_ISSET(ifd, &amp;wk_rset)) /* data is available for read */</td>
</tr>
<tr>
<td>22: recv(fd, buf, sizeof(buf), 0); /* read the data received */</td>
</tr>
<tr>
<td>23: /* process the response and send */</td>
</tr>
<tr>
<td>24: }</td>
</tr>
</tbody>
</table>

A typical sample snippets of socket program using C is shown in Table 1, and its full version is available as tcpserver_select.c [7]. Lines 01-02 correspond to master fdsets rset(read) and wset(write), and lines 03-04 for the working copy wk_rset and wk_wset of fdsets which is passed to select and is updated upon return. Each time select() is invoked, rset/ wset is copied into wk_rset/wk_wset (not shown in the table). Line 05 declares fd for listening and accepting new connection from clients. Each time a new connection is accepted, it gets its own fd, which is temporarily stored in connfd (line 06). Lines 09-11 are typical for any server side socket programming to let underlying operating system know that server is ready to accept the connections from clients. Line 14 shows the usage of select(). Line 15 checks for arrival of new connection and if yes, it is accepted at line 16 and new file descriptor connfd is added to master fdset in line 17. Line 18 is used to determine the maximum value of fd which should be used for scanning the fds. Lines 20-22 iterate over each fd value to check if the corresponding bit is set upon return from select() and if so, process the data received. The lines 14-21 are run forever in a loop (not shown here).
concurrent connected clients. The number of such connections easily exceeds thousands. A typical client e.g. mobile phone or even a browser on laptop when connects to a website, keeps the TCP connection on (live) even though it may not be transmitting data intermittently. For example, after a user has accessed a URL, the user may take a while to read the content returned by URL and during this reading time, no data transmission (neither by client nor by the server) takes place. In a typical scenario, out of these large number of live TCP connections, at any point of time only few connections are used for transmitting data and other connections remain idle. However, fdset bit vector has only 1024 bits, and this means that a server can only connect to 1021 (=1024-3 for stdin, stdout and stderr) clients. This puts a severe constraint on the server process to serve its large number of clients and thus an alternative approach to select() is needed.

II: fdset scanning overhead. Each time, select() is invoked, kernel scans the fdset bit vector to check for active sockets associated with invoking process. An active socket is one which has seen some activity and requires processing by the invoking program. For each such active socket, the kernel sets the respective bits and resets all other bits (corresponding to inactive sockets) and return the bit vector fdset. Upon return of the fdset value, the invoking process must scan each bit of fdset to identify the active socket fds and process these. Thus, there is always an overhead of double scanning of this fdset bit vector (once by kernel and once by the invoking process). Since this is a continuous operation being carried out for ever, it does degrade the performance.

To comprehend this scanning overhead, consider a scenario where at some point a server is connected to as a large number of concurrent clients, for example 1000. Now assume that client number 1000 is an active client for long duration interaction i.e. it continues to periodically interact with server process for a long time. Thus, bit number 1000 of fdset readfds will always be set whenever select() is invoked. Further consider that most of the other clients has completed the interaction with the server and only few clients continue to communicate, and when new clients come, these get the socket fd value as a small number e.g. 10. Thus, when server process invokes select(), it will set bit 10 and bit 1000 every time in fdset, and the value of nfds (first parameter) will be 1001. Thus, kernel needs to scan the entire bit vector from 1 to 1000 even though only few fds are of interest to server. Assuming socket fd 1000 (corresponding to long duration interactive client) always has data even though it may be just few bytes. In a typical implementation, the client process scans the fdset again from the beginning (bit 1) to the end (bit 1000) to find that there is data on socket 1000. Thus, in such a skewed distribution of socket fds, the scanning overhead does lot of redundant work, thereby impacting the performance.

Further, since the fdset is modified by select() upon return, it becomes necessary for the invoking process to maintain a master copy of fdset of all open fds e.g. rset, wset (lines 1-2 in Table 1). Using this master copy, invoking process must create a working (or temporary) copy of fdset e.g. wk_rset, wk_wset, which needs to be passed to select(). Thus, this is an extra overhead for invoking process to maintain a master copy of all the sets and involves one more buffer copying operation on each invocation of select().

III: State based approach. Kernel maintains the state of each open socket associated with a process whether it has an event of interest (e.g. data to be read or buffer available to write) irrespective of previous invocation of select(). Whenever select() is invoked by a process, kernel returns fdset value based on the state of each socket. For example, whenever data is received on a socket fd for the process to read, kernel keeps track of whether the process has read the complete data. Kernel maintains this state of the socket at all times. Thus, even if a socket has been notified earlier (via setting the bit corresponding to respective socket fd), but the invoking process has not fully read the data, kernel uses this state information to notify the process again. This maintenance of state for each socket is a significant overhead for kernel.

IV. Context switching between user space and kernel space [8][9]. The last parameter of select() call specifies the timeout after which the select call must return when no socket is ready with an event. However, if any socket is active, the select() call returns immediately. When the select is invoked, its execution happens in kernel space (till timeout occurs), the kernel iterates over the fdset, and for each socket fd, it registers callbacks for an event notification [12]. Thus, whenever an event occurs, such as data received on a socket, it iterates again over the fdset to deregister the callbacks. This results in significant work for kernel making it a heavy weight operation. Typically, the moment an event occurs, select() call returns and code execution enters user space. The process has to iterate fdset all over again to identify the active socket fds. After processing the active sockets, the process again invokes the select(), and this cycle repeats forever. Thus, for all practical purpose, for each event on network socket, there is continuous switch between user space and kernel space, a costly operation.

The program samples described here (Table 1 with full working code given in [7]) are written in C (socket programming was implemented at first in C). Our previous article [1] on socket programming used Python as the programming language for simplicity of understanding [10]. The Python language does support select() API and
uses the same semantics as in C, but has made it bit easier from programming perspective. The Python select() method returns 3 lists one each for read, write and exceptions which are of interests to the invoking program.

A sample Python snippet is given in Table 2 with full working code available at [7]. At first glance its implementation looks efficient since invoking program does not do any scanning on sockets to check if it is ready to be processed. However, it is just a convenience implemented in Python since Python implements select() as a wrapper over the underlying Unix select(). It internally scans over all fds, and returns list of objects for each fdset corresponding to read, write and exception. Thus, though the scanning of fd is hidden from the invoking program, it is performed internally by Python socket library and thus all limitations of original select() are applicable to Python as well.

Table 2: Using select with Python

### Poll(): Alternative to select()

To address the limitations on number of socket fds, and avoid maintaining a separate master copy, and poll() socket API is introduced. The usage of poll() is defined as

```c
struct pollfd {
    int fd;
    short events;
    short revents;
};
int poll(const struct pollfd *filedes[], unsigned int nfds, int timeout);
```

The first parameter to poll() is an array of open fds and it does not impose any limits on the size of array. Thus, a process can have any number of open socket fds that can be concurrently open and acted upon. Thus, the first benefit of poll() over select() is removal on the limit of number of open fds. This becomes useful when a server program needs to concurrently communicate with a large number of clients.

In the structure of pollfd, the 2nd and 3rd fields are events and revents. The field events specifies all the events the invoking program is interested in such as read, write, exception and even other events which is not supported by select(). An example of other events could be high priority data to read (corresponding to TCP urgent data in TCP headers [11]), invalid file descriptors etc. The field revents is the value returned by the kernel specifying the events which have occurred. Thus, upon execution of poll(), it does not modify (or destroy) the original event list or socket fds. The invoking program does not have to maintain a separate master copy of open fds. Hence, every time poll() is invoked, no buffer copying from master copy to temporary copy is required. However, the onus of checking the occurrence of events for each fd still lies with the invoking program. The program must check for each fd in the array to identify which events have occurred. The invoking program needs to process all the fds where events (e.g. data available on the socket to be read) have occurred.

A sample program snippet using poll() is shown in Table 3 with full code available at [7]. Line 01 declares the array of client fds to be passed to poll(). After creating a listen socket in lines 04–06, it is set...
with event POLLIN (lines 08–09) so that whenever new connection comes, it can know about the same. The server program runs forever between lines 12–25. The number of valid entries is maintained in the array is determined by maxfd, and first entry in poll fd points to listen socket. All other array elements will contain entries corresponding to TCP connections with clients. Program invokes poll() at line 13, and it first checks if a new connection has arrived (line 14). If so, it is accepted (line 16) and fd for new socket is added to array (lines 15–18). In the lines 20–23, for each of the open socket, check is made if there is data to be read and if yes, it is read and processed in lines 22-23. The program runs for ever. In this same code we haven’t shown the use of POLLOUT event (which is to know if socket buffer is available for write), and POLLERR event to determine any error condition. In actual code, these events need to be processed as required.

Though poll interface mainly addresses the issue of limited number of open sockets and avoids double copying, the other issues still persist as applicable to select(). The problem of scanning the fd array persists as in the case of select(). The invoking program must iterate over all the fds in the array and check, which is a significant overhead especially if the array size is large (i.e. large number of open socket fds), but the event occurs only on few of them. However, the scanning overhead is reduced a bit. In select, fdset size is fixed (1024 bits) and thus scanning happens till the highest number socket fd, whereas in poll() the size of pollfd array corresponds to number of open fds. Thus, only those fds are scanned which are open. Essentially poll() has the same scaling problem as in select() since scanning the fds requires work proportional to number of open fds instead of active fds (the fds of interest).

Though one does not need to maintain master copy of fds thus copying from master copy to temporary copy is avoided, but copying from user buffer to kernel buffer still happens. The amount of data that is copied may actually exceed that for select(). In poll(), each fd requires 64 bits (size of struct pollfd), compared to only 3 bits (1 bit each for read, write and exception) in select() for each fd. Thus, if number of open sockets are more than 3/64*1024, poll() involves more copying than select(). The other common mistake made by a typical developer is when an fd is closed. In this case, the developer sets the field event to 0 (implying no more interest in any events as fd is closed). However, this will result in EBADFD (error Bad FD) which requires unnecessary processing of this fd. Thus, the best way to handle the closed fd is to remove this fd from the array itself. The simplistic way would be to swap this array entry with the last fd entry of the array and decreasing the array size by 1.

Similar to select() implementation, the implementation of poll() is also state based i.e. kernel must maintain state of socket about the event handling [8]. Thus, if an event for a socket has been notified in the previous invocation of poll() and same has not been fully acted upon by the process, on the next invocation of poll() the kernel will again notify the invoking process about existence of the event as it maintains the state of socket. Similarly, switch of program execution between user space and kernel space continues to impact the system performance.

Therefore, from efficiency and scaling perspective, it is desirable that the invoking program should get the list of fds of interest i.e. those fds which are ready for processing. Both select() and poll() follow state based approach [8] which requires kernel to maintain the state of each fd, and provide response after checking the state of each fd. The kernel I/O subsystem is inherently designed using events based approach, such as, arrival of data packets, closing of a socket at the other end, getting a new connection request etc. In implementation of poll() and select(), kernel needs to transform the information from events based approach to state based approach and vice versa.

Thus, it is desirable to have a socket API that uses an event based approach instead of a state based approach. Linux supports an API epoll() [6] that implements events based approach.

4. epoll(): An events based approach

In this approach, once an event occurs and kernel notifies the process about it, the kernel’s responsibility is over. After a process is notified about the event occurrence, it can process the event immediately or make a note of it to be processed later. Further, the occurrence of events is monotonic in the sense that once an event occurs, it never decreases the amount of the information to be processed e.g. amount of data to be read or amount of buffer space available for writing. The main advantage this approach provides is that work done by kernel between two successive invocation of epoll() is proportional to number of events that have occurred during the invocation interval and independent of number of open socket fds a process is working with to serve respective clients.

The approach of events based notification poses its own challenges. Consider that between time T1 and T2, N events have occurred (e.g. other end has sent data N times during this interval) on a single socket fd. There are two choices for kernel to inform about these events. One choice is to inform the process about occurrence of each event (i.e. each time data is received). The other choice is to inform the process only once i.e. about the first event and treat remaining N-1 events coalesced into the first event. The information about subsequent N-1 event is not required till a process starts acting upon the first event. Using the former approach would cause too much work for the involved pro-
cess and in general latter approach is preferred. Another aspect one needs to consider whether these event notifications be asynchronous or synchronous. Asynchronous notification would mean that kernel informs the process whenever the event occurs and thus interrupts the process from whatever task it is doing. Its implementation would involve signals based mechanism. The synchronous notification would imply that kernel informs the process only when it asks for it and thus, event is stored by the kernel till the process asks for it. From the implementation perspective, programming with the latter approach is simpler. Further, in general, till a program processes the first event, it would not really care about all those subsequent events that occur after the first event but before the time program begins to process it.

The new socket API that supports such an implementation is epoll() mechanism introduced in Linux 2.6 [6]. The benefit is there is no linear scan of fds by user program and kernel. There is no limit on number of socket fds that can be used with this and thus enables a process to serve large number (>1024) concurrent clients. The usage of epoll() mechanism is given in terms of 3 APIs [13] [14][15] as below.

The first API epoll_create() creates an instance of of epoll(), and returns an fd (e.g. epfd) which is to be used in all subsequent invocations of epoll interface. The 2nd API epoll_ctl() tells the epoll mechanism which are the fds of interest to the process. This enables a process to add new fds for monitoring, remove a closed fd, and modify an fd for newer events of interests (specified by struct epoll_event). The information about events of interest is provided as a bit mask e.g. EPOLLIN for read, EPOLLOUT for write, EPOLLERR to detect some error condition, EPOLLET for edge triggered behaviour described below etc. The 3rd API epoll_wait() returns the count of descriptors that are of interest to the invoking process. Thus, the invoking process does not scan each of the fd, but instead gets a list of only those fds which needs to be processed.

Epoll interface introduces a concept of level triggered (the default value) and edge triggered (EPOLLET) event. The former is kept for compatibility and is just a faster version of poll(). Level triggered event mechanism is used whenever a program would like to use the semantics of poll(). The edge triggered interface (EPOLLET) implements event driven mechanism i.e. the kernel notifies the program for an event only once. To understand the same, consider the example of a simple client/server communication on a socket fd. Consider that client sends 1000 bytes of data which generates one read event for server. When server invokes epoll_wait, it returns fd for reading. Consider that server reads only 500 bytes of data and decides to postpone the read of remaining data some time later. However, when server invokes epoll_wait() again, there is no event for this fd. This API call would be blocked since kernel would wait for next event to occur (new data to be received) even though data is available on the socket for program to read. This is because, kernel is not using state based approach. To avoid such a case where server program gets blocked, the server must make this fd as non-blocking. With non-blocking socket, when read is done on the socket (which has no data), it returns error EAGAIN. Thus, using epoll_wait(), a socket fd must be added to the array only after read on it returns EAGAIN. The implementation of epoll does support more complex operations such as only one notification for multiple events (when client has sent multiple chunks of data resulting in multiple events). The oneshot notification can be specified with event type as EPOLLONESHOT, which we will not discuss here and suggest the reader to explore the same after having gained familiarity with the basics of epoll interface.

Table 4: Snippets of socket program using new epoll()
A sample program snippet using epoll interface APIs is shown in Table 4 with full code tcpserver_epoll.c available at [7]. For reasons of brevity, only those variables have been declared which are specific to the use of epoll(). The instance for epoll interface is created at line 09. The listening socket to accept new connections is added to list of fds at lines 10-12. The execution of epoll_ctl() for an fd to be done only once for all the events of interest. This has to be modified only when events of interest change, for example, when using non-blocking sockets and a process defers reading of complete data. In this code snippet, only EPOLLIN event (for read) is added. Other event for write (EPOLLOUT), edge triggered (EPOLLER) needs to be used in a real program. The program runs forever between lines 14-27. Fds of interests (for which some events have occurred) are returned by epoll_wait() at line 15. New TCP socket connection is accepted and added to list of sockets in lines 17-21. For existing TCP connections, whenever a socket fd receives an event (such as when respective client sends the data), the data is read at lines 22-24 and processed as required.

5. Summary

We have discussed the evolution of socket program starting from a simple client/server program [1], then evolved to a server programming model handling multiple concurrent clients using multiple threads/processes, and then finally to a single threaded server process using select(), poll() and epoll(). Use of select() is good enough for a server program when number of concurrent clients are limited to few hundreds. If the server program has to serve large number of concurrent clients [4], then using poll() can be used in place of select() following the same program semantics and it is an easier switch. However, when efficiency and service response time plays an important role for a server program, then using epoll() is the best choice. However, this requires that developer should be familiar with non-blocking sockets, which makes programming a bit more complex. Using epoll interface without using non-blocking sockets would most likely result in unsatisfactory performance as it may be possible that communication on one socket fd may cause process to block resulting in no or delayed response on other sockets (clients).

Thus far we have discussed the transport layer covering TCP and UDP and provided a number of programming scenarios to enable the experiential learning of transport layer. In our subsequent articles, we will explore network layer, primarily Internet Protocol (IP), its addressing scheme, subnetting and routing and continue the same approach of learning with simple real practical examples.

### 6. Experiential Exercises

To gain an experiential understanding of select(), poll() and epoll(), a working version of server programs using each of these API is available at [7], respectively as tcpserver_select.c, tcpserver_poll.c, and tcpserver_epoll.c. These programs are written for Ubuntu Linux machine and can run on Ubuntu 14.04, 16.04 and 18.04 LTS (Long Term Support) versions. The client can be a simple netcat (nc) utility or a Python or C program. A working version C client tcpclient.c is also available at [7], which supports many configuration parameters, such as, configure number of data packets to send, size of each data packet and interval between successive packets. Using the combination of these parameters, this program can be tuned to work as different type of clients, and help us to study and understand the behaviour of select(), poll() and epoll() APIs. Below, we describe some simple experiential exercises to develop a better understanding of these socket APIs.

**Note:** All the C programs need to be compiled before these
can be executed. The default IP address and port number for server program is taken as 0.0.0.0 and 9999 but different values can be specified at the time of running these programs. A simple usage of compiling and running these programs is shown in Table 5.

Exercise 1

**Topic: TCP Server using select() to serve few clients**

a.) Invoke the server program select (tcpserver_select.c)

```
$ ./select -s 10.211.55.11 -p 9999 -t 10
```

b.) Invoke 5 instances of client program as below in 5 separate terminal windows.

```
./tcpclient -s 10.211.55.11 -p 9999 -c 50 -i 5000 -d 1
./tcpclient -s 10.211.55.11 -p 9999 -c 10 -i 1000 -d 2
./tcpclient -s 10.211.55.11 -p 9999 -c 10 -i 1000 -d 3
./tcpclient -s 10.211.55.11 -p 9999 -c 60 -i 3000 -d 4
./tcpclient -s 10.211.55.11 -p 9999 -c 10 -i 1000 -d 5
```

The 1st and 4th client run respectively for 250s, and 180s whereas other 3 clients (2nd, 3rd, and 5th) runs for 10 seconds. The server simply displays the new connection whenever it is accepted.

c.) **Learning:** Working of socket select() to serve multiple clients.

Exercise 2

**Topic: TCP Server using select() to demonstrate the overhead of scanning full fdset bit vector.**

a.) Invoke the server program select (tcpserver_select.c)

```
$ ./select -s 10.211.55.11 -p 9999 -t 10
```

b.) Invoke about 500 clients which run for longer duration (e.g. 10 minutes) and other clients run for shorter duration (e.g. 1 minute). An example invocation of creating clients with this behaviour is given above.

d.) After about 24 seconds, only 3 clients will remain live with their corresponding fd on server side being 4, 14, and 514.

e.) The code if (FD_ISSET(ifd, &wk_rset)) at line 150 in tcpserver_select.c is executed about 500 times for the client #514 indicating the wasteful scanning of fdset. The code can be modified to see how many times this condition is executed before an fd is found to be active.

**Learning:** Understanding overhead of scanning fdset in select() when some fd number has a high value.

Exercise 3

**Topic: TCP Server using select() to reach the limit of concurrent clients**

a.) Invoke the server program ./select (tcpserver_select.c)

```
$ ./select -s 10.211.55.11 -p 9999 -t 10
```

b.) Invoke more than 1024 times. This should exceed size of fdset of server and thus accept() should fail.

c.) **Learning:** Understanding the limit on number of clients that can be connected with a TCP Server using select().

Exercise 4

**Topic: TCP Server using poll() to serve concurrent clients**

a.) By default, Linux system has limit of 1024 on number of open file descriptors. For poll() to work exceed this limit, the soft limit should be increased. Use the following to temporarily increase the soft limit to 2000 from 1024 on number of files a process can open.

```
$ ulimit -Sn 2000
```

b.) Invoke the server program ./poll (tcpserver_poll.c)

```
```
c.) Invoke the client with few concurrent clients as below.

```c
# client #1 to 1025, creates server socket fd from 4 to 14.
i=0;
while [ $i -lt 1025 ]; do
   (/client -s 10.211.55.11 -p 9999 -c 20 -d 2 -i 1000 &);
i=$((i+1));
done
```

Learning: Understanding use of poll() to deal with concurrent clients and exceeding the limits to 1024.

Exercise 5

Topic: TCP Server using epoll() to serve concurrent clients

a.) Repeat the first two steps of Exercise 4, i.e. ensure that number of open file descriptors exceed 1024, and run the epoll server program instead of poll server program

```sh
$ ulimit -Sn 2000
```

b.) Invoke the server program ./epoll (tcpserver_epoll.c)

c.) Invoke the client with 1024+ (e.g. 1025) clients as below

```c
# client #1 to 1025, creates server socket fd from 4 to 14.
i=0;
while [ $i -lt 1025 ]; do
   (/client -s 10.211.55.11 -p 9999 -c 20 -d 2 -i 1000 &);
i=$((i+1));
done
```

Learning: Understand the use of epoll() and how it avoids the overhead of checking all open fds.

References


[7]. Ram P Rustagi, source code for example programs using select, poll and epoll along with a client program to connect to server, https://github.com/rprustagi/EL-Evolution-of-Server-Socket-Programming, last accessed March 2020.


[4]. Marek, Select is fundamentally broken, https://idea.popcount.org/2017-01-06-select-is-fundamentally-broken/


Prof. Ram P. Rustagi

Prof. Ram P. Rustagi is currently working as Professor, CSE dept, KSIT Bangalore, and honed up his academic skills with Ph.D from IIT Delhi, and M.Tech from IISc Bangalore. Prior to KSIT, at Cavisson Systems, he mentored new technology development using Machine Learning techniques in Security and Performance Monitoring. At PES University, he had taught Undergraduates, Post Graduates students, and successfully guided 3 Ph.D scholars. At PESU, he brought innovations in teaching computer network and security courses, and introduced practical experiential learning exercises.

Prof. Viraj Kumar

Prof. Viraj Kumar is a Visiting Professor at the Divecha Centre for Climate Change, IISc Bangalore and the Vice-Chair of ACM India’s Special Interest Group in Computer Science Education (iSIGCSE). He was a consultant to the Committee to draft the National Education Policy (2017-18), and contributed to two education-related task groups of the Karnataka Knowledge Commission (2014-16). He holds a PhD in Computer Science from the University of Illinois at Urbana-Champaign.
EXPERIENTIAL LEARNING OF NETWORKING TECHNOLOGIES

Understanding Network Layer & IP Addressing

RAM P. RUSTAGI
Department Of CSE, KSIT Bengaluru

1. Introduction

So far in our articles[1], we have discussed a) Internet and network in general, such as, its functioning, network delays and performance, diagnostic tools to use etc. b) Application layer with detailed focus on HTTP protocol, its message structure, and use of HTTP headers in designing web applications, and c) Transport layer with focus on end to end delivery i.e. delivery of data from a process on one machine to a process on another machine. At the core, all of these require network communication between two computer systems. In this article, we will explore basics of communication mechanisms at network layer i.e. data exchange between two computers systems.

2. IP Address: Network and Host Part

When internet started, internet designers initially fixed the size of Internet address as 32 bits. This limits the maximum number of possible addresses to 2^32 (about 4 billion). At that time, no one had imagined and expected internet to grow to a size as it has reached today. This current address format is also known as IPv4 address. Initially, network layer was part of transport layer and it was separated from transport layer in early 80s. Till that time TCP[4] had already undergone 3 revisions and thus when IP[3] was separated from TCP, it was christened IPv4. Subsequently, after the invention of web in early 90s, internet started to grow at a fast rate and in few years, thus very soon number of devices connected to internet would have exceeded 4 billion. Thus, a new internet addressing scheme, called IPv6 [8], was designed with the primary objective of connecting large number of devices. In IPv6 address space size is fixed at 128 bits. With these many bits, total number of addresses would be 2^128. Considering the world population to be around 8 billion, the number of addresses per person would come to 295, a very high number. This address space is so large that even if each conceivable entity on this earth is assigned an IPv6 address, we would still have a large number of addresses available. We will explore IPv6 in subsequent articles.

A computer system can communicate with other computers only when it is connected to a network via some network interface. A machine can have many network interfaces and thus can connect to many networks at the same time. Examples of multiple such interfaces are Wired ethernet, Wi-fi, Bluetooth etc. When a machine is connected on multiple interfaces, a common question that arises is: does it have a single network address or multiple addresses. This can be better understood by an example analogy. Consider a 3-side corner building i.e. it is spanned on its sides by 3 roads. Consider that it is situated on 5th main on west side and spanned by 3rd cross on north and 4th cross on south side (we assume that on the east side, it has another neighbouring building). Let us further consider that the municipality when assigning the building numbers on each road has assigned it #100 on 5th main, #200 on 3rd cross and #300 on 4th cross. Thus, building can be reached by using any of these 3 addresses, that is, #100 5th main, or #200 3rd cross, or #300 4th cross. Thus, all three addresses belong to same building, though in reality, these addresses correspond to entry doors on the roads spanning the building. By analogy, the entry door of the building on a road acts like a network interface of a computer, and road can be considered as network link. Thus, network address in true sense is assigned to the network interface and not to the machine. When a machine has only one active network interface, network address is commonly attributed to address of machine as well and two are interchangeably used in general.

The network interface of the machine enables it to connect to a network. As the network address is of fixed size (32 bits for IPv4), one has to identify the network as well as unique address within the network from this network address itself. Thus, network address is divided into two parts: network part and host part. It is similar to door address of the building like door number (Host part), and road name (network part). When two building have same road name in their address, these are considered part of same locality. Persons in these two building on the same road can reach each other without ever going thru a road crossing. Similarly, two computer systems can communicate directly with each other when both belong to same network i.e. have same value in network part of the address. If for two computers, the value in network part differs, then these systems belong to two different networks, and thus, can't communicate directly. Systems connected to different net-

![Figure 1: Initial classification of IP addresses](image-url)

---

[1] Article 1
[2] Article 2
[4] Article 4
works can communicate only when these networks are connected by intermediary devices, known as routers. The size of network portion (number of bits starting from most significant bit) decides how many machines can belong to that network (by analogy, length of the road determines number of buildings that can exist on that road).

During the early days, networks were envisaged to belong to three categories: a) Large networks, b) Medium size networks, and c) Small size networks. These three networks were correspondingly classified as Class A, Class B and Class C networks[2], and respective number of bits in the network part of the address were fixed to distinguish between these classes. First 8 bits were assigned for class A; first 16 bits were assigned for class B, and for class C networks first 24 bits were assigned. The classification is depicted in Figure 1. Two more classes were defined as well, namely class D and class E. Class D is used for multicast addresses and have first 4 bits as 1110. Class E is reserved for future use, even today it continues to remain reserved.

Since these first few bits are overlapping among these 3 classes, these classes are identified by setting specific value for initial few bits in network part. Class A network has its first bit set as 0, and the value 01111111 of first 8 bits in the network part is reserved for local loopback interface. Thus, total possible number of values for class A network would range from 00000000 to 01111111, a total of 126 (=28–2) values. Similarly, class B network is identified with value of first two bits as 10. Thus, total number of possible class B networks are 214=16382 (first 16 bits would range from 10000000 to 10111111). Class C network is identified with first 3 bits as 110, and thus have 221 possible values for network addresses. For any of these 3 network classes, there is no classification in the host part. However, two values corresponding to all bits set as 0 and 1 in host part are meant for specific use and can’t be assigned to any computer system. Thus, number of bits in host part for class A network is 24 (32–8), and thus number of possible hosts in a class A is 224–2=1,677,721. This class A network was meant for a large corporation having very large number of computer systems. Similarly, the number of hosts in class B network is 216–2=65534, and number of hosts in class C network are 28–2=254. This class C network was typically meant for catering to most of small home/office networks.

This classification served the initial purpose, but could not cater to requirements of growing number of networks needed by many entities. For example, consider a small office having just 10 computers. At best, it will be allocated a class C address and this means 244 (254–10) addresses cannot be assigned to anyone else and would be wasted. Similarly, if a medium size company has 1000 computers, it will be allotted a class B address which again will result in non-usage of 64534 (65534–1000) addresses. Thus, this fixed classification resulted in under-utilization or wastage of network addresses. Considering this wastage, this classification with fixed size of network was done away with and instead varying size of network part was introduced. This is described in details in RFC 4632[10]. A network needs to have a minimum of two computers for them to communicate, and hence, host part should provide for at least 2 addresses. Since for any network, 2 more values need to be used (all 0s and all 1s), thus we need a total of 4 values in host part i.e. 2 bits.

Thus, network portion can have maximum of 30 (=32–2) bit. Thus, for all practical purposes, netmask can be maximum /30. Similarly, the network portion can have minimum of 1 bit, though most unlikely to be used as it would leaving 31 bits for host part implying 231–2 systems in a single network (a hypothetical case).

3. Representation of IP Address and Netmask

Representing IPv4 address it binary form with 32 bits is cumbersome to read and understand. Representing it as an integer value would be a shorter form but difficult to differentiate between the network and host part. For an easy understanding and interpretation, 32 bits are divided into 4 octets (8 bits) and these are written using decimal value separated by ‘.’ (Dot). Though there is no formal specification for this representation, these were first time described in RFC 790 [2]. The IP address is written as a.b.c.d/n, where a, b, c, d correspond to decimal value of each octet and n represents the number of bits in the network part.

Typically, any computer system or a smartphone connected to a Wi-Fi hotspot will have its IP address as 192.168.x.y/24 (x values would range between 0 and 255, and y value would range between 1 and 254), implying that network portion is 24 bits. Similarly, another IP Address in the same network would be 192.168.x.z/24. Please note that /24 is used to identify the network portion of the IP Address. Thus, these two IP addresses 192.168.x.y and 192.168.x.z belongs to same network with /24 as netmask (24 bits of network portion).

As each system connected to a network need to have an IP address (more precise would be the IP address of connected interface), as a first step towards our experiential learning, we would like to find the IP address of the computer one uses in day to day life. IP address of a Linux system can be identified by using the command ‘ip address show’ on the command terminal of the system as shown in Table 1. In older versions of Linux, the command ‘ifconfig’ was generally used. This old command is still supported in current versions but has been deprecated and it is recommended that user should avoid using this ifconfig command. Similarly, to find IP address of Android phone (when
connected to a hotspot), typical set of steps would be Settings->About->Status, and it will display IP address and other information related to the Android phone.

For a windows system, one needs to explore its settings via

\[ \text{Network} \rightarrow \text{interface} \rightarrow \text{properties} \rightarrow \text{TCP/IP address} \]

The set of steps for experiential exercise to know the IP address of a system is given in Exercise 1.

Table 1 shows two network interfaces, namely as lo and enp0s5 for a Linux system along with their IP addresses. The first interface lo (or lo0 on few other machines) corresponds to loopback interface. The second interface enp0s5 corresponds to actual network interface which connects this system to the ethernet network with its address as 10.211.55.11/24. The interface name may vary depending upon the variant and version of the Linux e.g. it could be eth0 or wlan0 (for Wi-Fi interface) etc. The representation /n (e.g./24) for netmask represents number of bits in network portion. Role of netmask is to determine the network number (corresponding to the network portion) of the network address. If a system has multiple physical network interfaces, then each would have its own IP address but these would belong to a different network number.

However, for most practical purposes, any end user system running a network application would have only one active network interface. An end user device either generates data packets and transmits these on to the network or receives packets on network interface and consumes the same. The devices that have two or more active network interfaces generally work as routers (or gateways) and their main function is receive data packet on one interface and transmit it on to other interface depending upon the destination of the data packet. A computer system that has two network interface does not forward the packets from one interface to another, and only works as either producer or consumer of packets is called dual homed host.

Each computer system is configured with an inbuilt loopback interface. This interface is always pre-configured (by default) with IP address as 127.0.0.1. This interface is not connected to any external network and enables 2 or more processes on the same machine to communicate with each other. The primary purpose of loopback interface is to enable network communication between two or more processes within the same system.

Thus, a user can develop, test and verify network functionality of an application on a standalone computer without the need of connecting it to any actual network. If two processes can successfully communicate on loopback interface using IP Address 127.0.0.1, then from network functionality perspective, these applications will work equally well when communicating on an actual network interface e.g. ethernet, Wi-Fi etc.

Table 1: Finding IP address of a Linux machine

<table>
<thead>
<tr>
<th>Interface</th>
<th>IP Address</th>
<th>Netmask</th>
<th>State</th>
<th>Scope</th>
<th>Gateway</th>
</tr>
</thead>
<tbody>
<tr>
<td>lo</td>
<td>127.0.0.1/8</td>
<td></td>
<td>unknown</td>
<td>host</td>
<td></td>
</tr>
<tr>
<td>enp0s5</td>
<td>10.211.55.11/24</td>
<td></td>
<td>up</td>
<td>dynamic</td>
<td>10.211.55.255</td>
</tr>
</tbody>
</table>

HANDS ON

```
$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 00:1c:42:5a:62:92 brd ff:ff:ff:ff:ff:ff
   inet 10.211.55.11/24 brd 10.211.55.255 scope global dynamic noprefixroute
      valid_lft 1065sec preferred_lft 1065sec
      inet6 fdb2:2c26:f4e4:0:9091:5823:b5b0:ee69/64 scope global temporary dynamic
      valid_lft 60472sec preferred_lft 85669sec
      inet6 fdb2:2c26:f4e4:0:4ba2:dc6c:c082:b421/64 scope global dynamic mngtmpaddr
      noprefixroute
      valid_lft 2591700sec preferred_lft 604500sec
      inet6 fe80::2d45:435:36ec:a36/64 scope link noprefixroute
      valid_lft forever preferred_lft forever
   inet6 fdb2:2c26:f4e4:0:4ba2:dc6c:c082:b421/64 scope global dynamic mngtmpaddr
      noprefixroute
      valid_lft 2591700sec preferred_lft 604500sec
      inet6 fe80::2d45:435:36ec:a36/64 scope link noprefixroute
      valid_lft forever preferred_lft forever
```
4. Netmask, Network Number and Assignable Addresses

A netmask is used to compute the network number given the network address. This is computed by applying bitwise AND operation between 32 bits representation of IP address and netmask. Thus, first we need to convert /n netmask notation into 32 bits. The 32-bit representation of netmask /n is obtained by setting the value 1 for first n most significant bits and remaining 32-n bits as 0 and writing the entire value in binary (or DDN for better readability). Thus, netmask of /24 in binary form is written as 11111111 11111111 11111111 00000000 which in DDN form would be 255.255.255.0. Similarly, netmask of /8 and /16 in DDN form would be respectively written as 255.0.0.0 and 255.255.0.0. These mask values of /8, /16 or /24 were simple to convert, as the boundary between all 1s and all 0s coincided with octet boundaries. For the netmask other than these values e.g. /21 would correspond to first 21 bits as 1, and remaining 11 (=32-21) bits as 0, that is, 11111111 11111111 11111110 00000000, which in DDN form would correspond to 255.255.248.0.

The computation of network can be better understood with an example. Consider an IP class C address 192.168.43.11/21. The binary representation of IP address is 11000000 10101000 00101011 00010011 and binary representation of netmask is 11111111 11111111 11111100 00000000. Bitwise AND of these two gives 11000000 10101000 00101000 00000000 which in DDN form would be N1=192.168.40.0. Thus, network number of 192.168.43.11/21 is 192.168.40.0. In simplistic terms, we can also obtain the network number of an IP address by just setting all the bits in host part as 0. To help understand it further, consider two more network addresses IP1=192.168.42.24/21 and IP3=192.168.48.31/21, both having same netmask of /21. It has been author’s experience that on cursory look, some users think that all these 3 addresses belong to same network, which is not the case. Computing network number of the latter two IP addresses by applying bit wise AND operation, their respective network number would be N2=192.168.40.0 and N3=192.168.48.0. Since N1 and N2 are equal, this implies that addresses IP1 and IP2 belong to same network. However, N1 is different from N3, and this implies that IP1 and IP2 belong to different networks and systems belonging to these networks can’t communicate directly and would require intermediate router(s).

Having computed network number, next natural question would be to determinenumber of possible addresses that can exist (or assigned to systems) in a network. This is solely determined by number of bits in the host part. We have just noted that in the host part, setting all bits as 0 gives the network number. Similarly, a network address achieved by setting all bits as 1 in host part is known as broadcast address. A broadcast address means all machines in the network. So, when a packet is sent to a broadcast address, then that packet will be delivered to all the machines in the network. It is needless to say that only live systems in the network will receive this packet, and any machine which is down will not receive this broadcast packet (there is no queuing of packets for later delivery).

For example, broadcast address for 192.168.3.2/24 would be 192.168.3.255 (obtained by setting all the 8 bits of host part as 1). Similarly, broadcast address for network 192.168.40.0/21 would be 192.168.47.255. Thus, if a lab has network number as 192.168.3.0 and broadcast address as 192.168.3.255, then ping to the address 192.168.3.255 from a system would send the ping request to all the systems in the lab and receive response from all those systems which are up and running (Windows systems in the network would not respond as windows have firewall enabled by default and reject any unsolicited incoming packet). Thus, a single ping request to a broadcast address would result in receiving multiple responses. Exercise 3 provides the steps to send a broadcast ping request and receive multiple responses.

As netmask defines number of bits in network part, and thus number of bits in host part becomes 32-n. Thus, number of possible IP addresses that can be assigned to hosts in a given network with netmask /n will be equal to 232-n-2. The total possible values with 32-n bits are 232-n, out of which two values (corresponding to all 0s identifies the network number, and other corresponding to all 1s identifies broadcast address) are reserved, hence number of assignable IP address becomes 232-n-2. This computation serves as a basis to determine the number of bits required in netmask for an office network. Consider that an office has 20 computers and would like to get a network number so as to assign IP addresses to these computers in the office. For any network, we need to increase the count of IP addresses by 2 to account for network number and broadcast address, thus increasing the total address count in this office to 22 (=20+2). The minimum number of bits required to represent 22 values are 5, Thus, the number of bits in host part would be 5, and hence number of bits in netmask would be 32-5=27. Netmask of 27 bits enables 232-27-2=25-2=30 computers in the network. Even though the office needs only 20 addresses, the network will have 10 extra addresses which can be used in future (or may remain unused). This is better than assigning a class C address with netmask of /24 where it will only use 20 addresses and leave 254-20=234 address unused, and even in future offices unlikely to use them fully and hence assignment of class C address leads to wastage of network address space. Assuming the network number assigned to this network is 192.168.1.0/27, then first assignable address is 192.168.1.1.
in the network would be 192.68.11, the last assignable address would be 192.168.1.30, and broadcast address would be 192.168.1.31 (corresponding to all 5 bits in host part set to 1, i.e. 11000000 10101000 00000001 00011111). Please note that last octet of last assignable address is 30 and not 254. Quite often, author has seen that students err in computing the value of last assignable address, and mistakenly assume that the last octet of last assignable address is 254.

5. Network and Host Reachability

Typically, when a computer is connected to a network e.g. office LAN, or home Wi-Fi network, it gets an IP Address, and same is used by the computer for any communication with any other system. One question that naturally arises is: how does a computer gets the IP Address? Typically, each computer is generally configured as a DHCP (Dynamic Host Configuration Protocol) [7] client, and whenever a computer starts, it broadcasts a DHCP Discover message to check if there is a DHCP server present in the connected network. Typically, a Wi-Fi hotspot (or internet/LAN router) acts as DHCP server in the network and responds to DHCP Discover request message. Similarly, in an office network where computers are connected via ethernet cables, and switches, there is a DHCP server configured by the network admin. The DHCP client initiates communication with DHCP server and obtains the IP address which it can use for the allotted time duration (known as lease time in DHCP parlance). On expiry of lease time, DHCP client renew this allotted IP address. On renewal, DHCP server is expected to allot same IP address as assigned before, but it is not guaranteed and DHCP server and allot a new IP address which is different from the previous one. Further, It is the responsibility of DHCP sever to ensure that each system in the network gets a unique IP address, otherwise there would be a conflict, and machines with duplicate IP Addresses would not be able to communicate with any other system. As all systems have same network number, these systems in an office (or lab) can reach each other directly i.e. without requiring any router.

For a computer system, even though one IP Address is good enough to communicate with other systems, network layer specifications donot impose any restrictions on assigning multiple IP addresses to an interface. Assigning of multiple IP addresses to an interface becomes the necessity under certain situations. For example, during the time of current Covid pandemic, a good number of people are working from home and connect to their office/corporate network using VPN (Virtual Private Network)[9]. Now, network interface works with two IP address, one corresponding to IP Address as belonging to local network (e.g. Wi-Fi hotspot, or local home network) as assigned by DHCP server, and other corresponding to network that belongs to office facilitated by VPN server. We will use this concept of assigning multiple IP addresses to an interface, and assign an IP address of a different network to differentiate between reachability within same network and non-reachability to a different network.

Consider the case of network setup (e.g. in home or office) where two computers are connected to a network. For simplicity, let us assume that two systems CA and CB are connected to a Mo-Fi (mobile phone based Wi-Fi hotspot) and their respective IP addresses are 192.168.43.201 and 192.168.43.202 with a netmask of /24 (it is a typical example of IP address assignment when using Android phone as hotspot). The network number of these system would be 102.68.43.0/24. Ping to CB(IP address 192.168.43.202) from CA(having IP address 192.168.43.201) will be successful assuming no firewall in the network (or hotspot) blocking the packets in the network. Pinging an unassigned IP address in the same network e.g. 192.168.43.99 will result in no response and it will fail with timeout error. However, when we ping to an IP address in an unknown network e.g. 192.168.44.99 (with number as 192.168.44/24), generally, it may givean error indicating network is not reachable or may still show timeout error. However, if one pings www.google.com (assuming hotspot is connected to internet), you get successful response even though www.google.com is in a different network. This involving routing of IP packets which we will discuss in our subsequent articles.

Assigning of multiple IP addresses and checking reachability can also be experienced even if a user has just only computer and no hotspot. On the native system (windows or Linux), install VirtualBox [11] (readers may install any other VM enabling software e.g. VMWare etc. as well), which provides an environment to run virtual machines on Windows, Linux, Mac etc., call host OS. For example, author of this article has installed Parallels software (a commercial software enabling virtual machine environment) on Macbook, and thereby accesses both Linux and Windows under the native (or host) MacOS. Using this VirtualBox (or any other VM enabling software), launch Ubuntu Linux as the guest OS. This VM software, by default, provides DHCP server functionality and guest OS gets the IP address from this built-in DHCP server. Using the guest Linux OS (on a VM), one can assign multiple IP address to same interface as shown in Table 2. By default, Linux network interface has been assigned IP address 10.211.55.11, as shown by command ‘ip -4 addr show dev enp0s5’ (line 01). The network interface enp0s5 has been assigned two additional IP addresses 10.211.55.101 and 10.211.56.101 (line 02 and 03) using the command ‘sudo ip addr add 10.211.55.101/24 dev enp0s5’ and ‘sudo ip addr add 10.211.56.101/24 dev enp0s5’ Assigning of an IP address requires super user privilege and hence ‘sudo’ needs to be used. The host system MacOS has been assigned the IP address 10.211.55.2/24 by the DHCP server of Parallels VM
software (output of line 01 Table 3). When from native OS (MacOS), manually assigned additional IP Address 10.211.55.101 of Ubuntu Linux is pinged, it is successful (line 02 Table 3), whereas the IP address 10.211.56.101 is not reachable (line 03). This is because this IP address has network number 10.211.56.0/24 which is different from network number 10.211.55.0/24 of virtual interface vnic0 of host system MacOS. The set of instructions for experiential learning of this concept of assigning multiple IP addresses to an interface and checking (non)-reachability are described in Exercise 2.

6. Publicly Usable Private Addresses

In a typical network setup irrespective of whether it is in a corporate network, home office, educational institute, or using Wi-Fi hotspots, IP address assigned to any system is generally observed to be either 192.168.x.x/24, or 172.16.x.x/16-172.31.x.x/16, or 10.x.x.x/8. For a small network such as Wi-Fi hotspots or home/office network connected using broadband-DSL (Digital Subscriber Line) or fiber, the most common IP address observed are 192.168.1.x or 192.168.3.x etc. This implies that a lot of computer systems when checked for their IP address would have same value of IP address e.g. 192.168.1.2/24. We have discussed above that a system should have unique IP address and if two system have same IP address, then IP address conflict occurs and these machines will not be able to communicate. For example, machine A at home and machine B in office may have the same IP Address 192.168.3.2/24 assigned to them and both machines work fine and able to access internet e.g. www.google.com. So, a natural question that arises is how do these systems with same IP addresses continue to work fine without any issues.

In the initial days, when internet spread was minimal and mostly confined to academic institutes/research organization, each was assigned a unique network number by internet authorities, such as IANA (Internet Assigned Numbers Authority)[12]. The IP address allocation authority for India is IRINN (Indian Registry for Internet Names and Numbers)[13] and for Asia-Pacific, it is APNIC (Asia Pacific Network Information Centre) [14]. Subsequently, when internet spread became far and wide, depletion of IP addresses was realized, and to circumvent the same, NAT (Network Address Translation) technology was developed[5]. Using NAT, any network that needs to access internet, the globally unique IP address is required only for the internet gateway that connects the network to internet (we will discuss NAT in detail in subsequent articles). All other computer systems in the local network (e.g. office or home network) need not have a globally unique IP address. However, these computer systems do need an IP address which needs to be locally unique within the local network and can be hidden from the internet by the internet gateway. Thus, to cater to this need of assigning unique IP addresses within a local network, few IP network ranges are reserved by internet authorities for private use by any one. These address ranges are a) 10.0.0.0–10.255.255.255 or 10.0.0.0/8, b) 172.16.0.0–172.31.255.255 or 172.16.0.0/12, and c) 192.168.0.0–192.168.255.255 or 192.168.0.0/16. The details of these IP address ranges and networks is given in [6]. These IP addresses can be used by anyone without requiring any permission from any network authorities, and therefore, these IP addresses are also called publicly usable private IP addresses.

7. Summary

We have discussed IP address assignment, and its representation using Dotted Decimal Notation (DDN). Using this DDN notation, 32 bits of IP address are written

```
01: $ ip -4 addr show dev enp0s5
2: enp0s5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    inet 10.211.55.11/24 brd 10.211.55.255 scope global dynamic noprefixroute enp0s5
        valid_lft 1591sec preferred_lft 1591sec
02: $ sudo ip addr add 10.211.55.101/24 dev enp0s5
03: $ sudo ip addr add 10.211.56.101/24 dev enp0s5
04: $ ip -4 addr show dev enp0s5
2: enp0s5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    inet 10.211.55.11/24 brd 10.211.55.255 scope global dynamic noprefixroute enp0s5
        valid_lft 1508sec preferred_lft 1508sec
    inet 10.211.56.101/24 scope global enp0s5
        valid_lft forever preferred_lft forever
    inet 10.211.55.101/24 scope global secondary enp0s5
        valid_lft forever preferred_lft forever
```

Table 2: Assigning Multiple IP Addresses to an interface
01:$ ifconfig vnic0 inet
vnic0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=3<RXCSUM,TXCSUM>
inet 10.211.55.2 netmask 0xffffff00 broadcast 10.211.55.255

02:$ ping -c2 10.211.55.101
PING 10.211.55.101 (10.211.55.101): 56 data bytes
64 bytes from 10.211.55.101: icmp_seq=0 ttl=64 time=2.751 ms
64 bytes from 10.211.55.101: icmp_seq=1 ttl=64 time=0.515 ms

--- 10.211.55.101 ping statistics ---
2 packets transmitted, 2 packets received, 0.0% packet loss

03:$ ping -c2 10.211.56.101
PING 10.211.56.101 (10.211.56.101): 56 data bytes
Request timeout for icmp_seq 0

--- 10.211.56.101 ping statistics ---
2 packets transmitted, 0 packets received, 100.0% packet loss

### Table 3: IP Address of VM virtual interface

<table>
<thead>
<tr>
<th>VM IP Address</th>
<th>Host IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.0.200</td>
<td>10.0.0.200</td>
</tr>
<tr>
<td>192.168.0.201</td>
<td>10.0.0.201</td>
</tr>
<tr>
<td>192.168.0.202</td>
<td>10.0.0.202</td>
</tr>
</tbody>
</table>

as a.b.c.d where each of a, b, c, and d is a decimal number between 0 and 255. We also discussed the role of netmask and network number and understood that two machines in same network can communicate with each other directly i.e. without needing any intermediate router device. We also explored assigning of multiple IP addresses to a single network interface. Lastly, we discussed Network Address Translation (NAT) mechanism to enable users to access internet in view of depletion of IPv4 address space and use of publicly usable private IP addresses i.e. 192.168.0.0/16, 172.16.0.0/12, and 10.0.0/8.

Continuing in this series, in the next article, we will explore basics of IP routing, Variable Length Subnet masking (VLSM) and use of ICMP to diagnose network.

### Experiential Exercises

#### Exercise 1

**Topic:** Identify IP addresses and netmask of your connected devices

- a. Connect your laptop/deskop to internet e.g. via your home DSL broadband router or via your phone hotspot.
- b. Open terminal window and identify network interface, IP Address and netmask.
  - i. On linux, `ip addr show`
  - ii. On Macbook `ifconfig`
  - iii. On windows `ipconfig/all`

#### Exercise 2

**Topic:** Assigning Multiple IP Addresses to an interface

- a. If you don't have VM software, then install VirtualBox[II] and install Ubuntu Linux as the guest operating system under VirtualBox or your existing VM platform. To install VirtualBox and guest OS Ubuntu, follow the instructions as given in [II].
- b. Login into Ubuntu systems and identify the IP address and netmask of the network interface.
- c. Identify the IP address of virtual interface on host system e.g. Windows/Mac etc.
d. Open a terminal window on host machine and ping the IP address of guest OS (Ubuntu). This ping should be successful.

e. On the guest OS (Ubuntu), assign a new additional IP address within the same subnets as that of existing IP address. For example, if existing IP address is 192.168.43.11/24 and corresponding network interface is eth0, then assign 192.168.43.21 to this IP address (replace 192.168.43.21 and eth0 with appropriate values as needed).

   i. sudo ip addr add 192.168.43.21/24 dev eth0

f. From the host machine (Windows or Mac), ping this new assigned IP address. Ping should be successful.

g. Assign a 2nd new IP address but belonging to a different network e.g. 192.168.44.11/24. This network number would be 192.168.44.0/24 which is different existing network number 192.168.43.0/24. Ping to this new IP address (e.g. 192.168.44.11 or whatever you have configured) from host OS terminal windows. The ping should fail.

Learning: Assigning of multiple IP addresses to a network interface reachability within same network and non-reachability in different network.

Exercise 3

Topic: ping to a broadcast address.

a. If you have access to an office or home network which have multiple devices connected to a network e.g. multiple smartphone to a hotspot use that setup.

b. Identify IP address of your laptop as discussed in Exercise 1.

c. Compute the broadcast address for your discovered IP Address. For example, if IP Address of your laptop is 192.168.43.201/24, then broadcast address would be 192.168.43.255.

d. Windows based system by default have their firewall enabled and thus to receive response from these systems, firewall should be disabled for ping request.

e. From the terminal window of your laptop, ping to this broadcast IP Address.

f. You should see multiple responses. The same can be verified by using the command ‘arp -a’ which will output all the IP Addresses discovered.

Learning: Understanding usage of broadcast address e.g. using a single ping command to identify all live machines within the network.

References


MEDIA KIT

Make accessible to engineering students, faculty and engineering practitioners, the advanced computing and communication topics

READER PROFILE

Engineering college students and Professors
Masters and Doctoral students
Engineering practitioners

READERSHIP

PRINT: 20,000 - 25,000
ONLINE: 75,000-100,000

<table>
<thead>
<tr>
<th>SL NO</th>
<th>Page</th>
<th>Cost in ₹</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Full Page</td>
<td>70,000</td>
</tr>
<tr>
<td>2</td>
<td>Half Page</td>
<td>40,000</td>
</tr>
<tr>
<td>3</td>
<td>Double Spread</td>
<td>140,000</td>
</tr>
<tr>
<td>4</td>
<td>Inside Front Cover</td>
<td>125,000</td>
</tr>
<tr>
<td>5</td>
<td>Inside Back Cover</td>
<td>100,000</td>
</tr>
<tr>
<td>6</td>
<td>Back Cover</td>
<td>1,50,000</td>
</tr>
</tbody>
</table>

SUBSCRIPTION

Category A: Institutional for 5 years ₹5,000/- plus 3 copies of ACC
Category B: Non-institutional for 5 years ₹1,200/-
Category C: Non-institutional for 3 years ₹1,000/-

To place an advertisement please contact
+91 76249 57142    +91 80 2360 7142  acc@accsindia.org

Note: Taxes are applicable