IEEE
potentials
THE MAGAZINE FOR HIGH-TECH INNOVATORS
September/October 2018, Vol. 37 No. 5

Insides Outed

In this issue

- Authentication patterns for cloud services
- Assisting future cellular communication
- Software defined networks
- Graphical trust models for agent-based systems

IEEE

IEEE Collabratec™

Bright Minds. Bright Ideas.

Introducing IEEE Collabratec™

The premier networking and collaboration site for technology professionals around the world.

IEEE Collabratec is a new, integrated online community where IEEE members, researchers, authors, and technology professionals with similar fields of interest can **network** and **collaborate**, as well as **create** and manage content.

Featuring a suite of powerful online networking and collaboration tools, IEEE Collabratec allows you to connect according to geographic location, technical interests, or career pursuits.

You can also create and share a professional identity that showcases key accomplishments and participate in groups focused around mutual interests, actively learning from and contributing to knowledgeable communities. All in one place!

Network.
Collaborate.
Create.

Learn about IEEE Collabratec at
ieee-collabratec.ieee.org

IEEE

IEEE potentials
THE MAGAZINE FOR HIGH-TECH INNOVATORS

September/October 2018
Vol. 37 No. 5

ON THE COVER:
Looking inward for engineering answers.

COVER IMAGE: ©ISTOCKPHOTO.COM/DEVRIMB

FEATURES

DEPARTMENTS & COLUMNS

MISSION STATEMENT: *IEEE Potentials* is the magazine dedicated to undergraduate and graduate students and young professionals. *IEEE Potentials* explores career strategies, the latest in research, and important technical developments. Through its articles, it also relates theories to practical applications, highlights technology's global impact, and generates international forums that foster the sharing of diverse ideas about the profession.

IEEE

Digital Object Identifier 10.1109/MPOT.2017.2783535

IEEE Potentials (ISSN 0278-6648) (IEPTDF) is published bimonthly by The Institute of Electrical and Electronics Engineers, Inc. **Headquarters address:** 3 Park Avenue, 17th Floor, New York, NY 10016-5997. Phone: +1 212 705 7900. **Change of address** must be received by the first of a month to be effective for the following issue. Please send to IEEE Operations Center, 445 Hoes Lane, Piscataway, NJ 08854. **Annual Subscription**, for IEEE Student members, first subscription US$5 included in dues for U.S. and Canadian Student members (optional for other Student members). Prices for members, nonmembers, and additional member subscriptions are available upon request. **Editorial correspondence** should be addressed to IEEE Potentials, 445 Hoes Lane, Piscataway, NJ 08854. Responsibility for contents of papers published rests upon authors, and not the IEEE or its members. Unless otherwise specified, the IEEE neither endorses or sanctions any positions or actions espoused in *IEEE Potentials*. All republication rights including translations are reserved by the IEEE. **Copyright and Reprint Permissions:** Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limits of U.S. copyright law, for private use of patrons, articles that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923. For other copying, reprint, or republication permission, write to *IEEE Potentials* at Piscataway, NJ. All rights reserved. Copyright © 2018 by the Institute of Electrical and Electronics Engineers, Inc. Printed in U.S.A. **Subscriptions, orders, address changes:** IEEE Operations Center, 445 Hoes Lane, Piscataway, NJ 08854, Phone: +1 732 981 0060. Other publications: IEEE also publishes more than 30 specialized publications. **Advertising Representative:** IEEE Potentials, 445 Hoes Lane, Piscataway, NJ 08854, Phone: +1 732 562 3946. **IEEE Departments:** IEEE Operations Center (for orders, subscriptions, address changes, and Educational/Technical/ Standards/ Publishing/Regional/Section/Branch Services) 445 Hoes Lane, Piscataway, NJ 08854, USA. Operations Center +1 732 981 0060; Washington Office/Professional Services +1 202 785 0017. Headquarters: Telecopier +1 212 752 4929, Telex 236-411.

Periodicals postage paid at New York, NY, and at additional mailing offices. **Postmaster:** Send address changes to IEEE Potentials, IEEE, 445 Hoes Lane, Piscataway, NJ 08854, USA. Canadian Publications Agreement Number 40030962. Return Undeliverable Canadian Addresses to: Fort Erie, ON L2A 6C7 Canada. Canadian GTS #125634188.

PRINTED IN THE U.S.A.

Digital Object Identifier 10.1109/MPOT.2017.2783536

Just in time

by Cristian Quintero

Time is one of the biggest worries for us as students: we need to find a job, start doing what we learned, and earn some money to pay our debts and start a life. But, sometimes, progress doesn't come as fast as we would like because engineering is a hard field of study. An education estimated to last four years may take up to six, and while we keep studying and giving our best effort, time keeps ticking away, and life goes on.

Such is life; we always have to put forth the effort and make sacrifices to get what we want. In my experience, I am running my own company, and it's been a bumpy road. Since I often attend to my business, I have missed several classes at my university. As I lose classes, sadly, I also miss exams, which has put me behind friends, who will now earn their degrees earlier than I will. So, I had to sacrifice the time it will take to finish school to run a business and have financial freedom. In the end, though, it's been worth it.

The most important thing is to enjoy what you do. I love my company and everything about it. Hearing feedback from a happy client is enough for me to say to myself, "We are doing it right," and all the worries just wash away. Just focus on finding your path, go after that thing that makes you happy, and do it! It doesn't matter if it takes you more time to get what you want or if you feel behind—just try to find the happiness in the journey and everything is going to be fine.

It doesn't matter if it takes you more time to get what you want or if you feel behind, just try to find the happiness in the journey and everything is going to be fine.

Everyone travels at his/her own pace. Someone may have graduated at 22 but did not get a permanent job until 27, while another person earned a postgraduate degree at 25 but died at 50. Maybe you are still single while some of your high school friends are already married with children. Barack Obama left the White House at age 55, and Donald Trump began his presidency at 70. Everyone in this world lives according to his/her own time. The people around you may appear to race ahead of you, and others seem to fall behind. But everyone is running his/her own race in his/her own time. Do not envy them; they are on their own path, and you are on yours. So, relax. You have not arrived late or early. You are just in time.

We are always running to get what we want. What moves you?

About the author

Cristian Quintero (cristianquintero@ieee.org) is the student editor of *IEEE Potentials*.

Digital Object Identifier 10.1109/MPOT.2018.2849518
Date of publication: 6 September 2018

IEEE setting its SIGHT on underserved communities

by J. Patrick Donohoe

The IEEE Special Interest Group on Humanitarian Technology (SIGHT) program is a network of volunteers from around the globe that teams with underserved communities and local organizations to leverage technology for sustainable development. SIGHT partnering is characterized by key project leaders who are citizens and/or permanent residents of the geographical area where the development takes place. Opportunities exist for Student Branches to participate in SIGHT projects.

The SIGHT program operates based on several key values, including a focus on sustainable solutions that make a long-term difference in the lives of people while operating through local volunteers and partners working with local communities. Partnerships are required for a successful SIGHT project, starting with the community and extending to government organizations, nongovernment organizations, schools, hospitals, companies, and others. Continuous training and education are essential for SIGHT and its volunteers to be both effective and sustainable.

The SIGHT program operates based on several key values, including a focus on sustainable solutions that make a long-term difference in the lives of people while operating through local volunteers and partners working with local communities.

SIGHT groups consist of at least six IEEE members who come together to learn about sustainable development, build relationships within their local communities, and implement SIGHT group projects that utilize technology to tackle key problems within the community. Non-IEEE members are welcome and encouraged to join their local SIGHT group. Groups can be formed at the professional or university level. The SIGHT Steering Committee invites proposals for projects aligned with its vision and with budgets between US$1,000 and US$19,999. Proposals must demonstrate consultation with, and a clear understanding of, the needs of target communities and demonstrate a needs-driven approach to strengthening local capacity through technology deployment. An IEEE operating unit (normally the local IEEE Section) must act as a fiscal agent for each SIGHT group. SIGHT groups assess their operations and activities for the year through an annual report submitted each year to IEEE staff.

SIGHT groups work in assistive technology, information and communications technology, agriculture, energy, environment, health, sanitation, water, and robotics—as long as technology development, deployment, or policy is part of the solution to the problem. To implement sustainable and impactful projects in local communities, groups need to build various capacities including, but not limited to, forming community partnerships, recruiting and engaging volunteers, and securing funding.

Some examples of funded SIGHT projects include solar power installation for light-emitting diode lighting; water purification systems and other infrastructure at a remotely located school in Honduras; improvements to small-scale aquaculture in Paraguay through the installation of solar water pumps and lighting, along with a mechanized fish harvesting system; neonatal fatality prevention in Uganda through wearable vital sign monitors designed for resource-constrained hospitals; and a solar-powered charging system for wheelchair batteries in Bangladesh.

You can learn more about SIGHT by visiting http://sight.ieee.org/.

About the author

J. Patrick Donohoe (p.donohoe@ieee.org) is the IEEE Member and Geographic Activities—Student Activities Committee chair.

Digital Object Identifier 10.1109/MPOT.2018.2847838
Date of publication: 6 September 2018

IEEE Students has a new look!

by Shareyna Scott

We are bold. We are imaginative. We see the future of IEEE engineering in you. And now, we have a student visual identity that speaks to the same—serving all 100,000 of us at more than 2,000 Student Branches (SBs) throughout 100 countries, globally. Our new identity couples the IEEE core values and diversity of IEEE Student and Graduate Student Members for all IEEE designated fields of interest.

To create a consistent look and demonstrate the connection of SBs to the larger IEEE, we now have our own identity that can be used in a variety of ways. When using the new look, the IEEE core values, brand personality, and style (as well as the diversity, areas of study, and historical legacies of IEEE Students) should remain in tact.

By using this new look, your SB story can now be told and recognized by so many other members just like you. An easily recognizable identity provides our Student Members with a visual connection to not only the student program but to the larger IEEE as well.

IEEE students can now access a new toolkit that includes a variety of downloadable content including banners, flyers, and more.

While it will serve as the official mark, it also will be easily identifiable to help students connect with the valuable resources that the organization provides to students.

How is this accomplished? Students can show their IEEE pride by wearing "IEEE Students" on clothing, swag, and even labels for your accessories.

Tools of the trade

To assist with the use of the new identity, a tool kit has been developed that includes a variety of easily downloadable content including banners, guidelines, table covers, flyers, video screen frames, bumper stickers, and more. More information can be found at https://brand-experience.ieee.org/guidelines/brand-identity/sub-brand-guidelines-and-templates/students/.

We are looking for students to become ambassadors of the IEEE Students identity. If you are interested or have questions, please contact us at student-services@ieee.org.

About the author

Shareyna Scott (s.n.scott@ieee.org) is the IEEE Student Branch development specialist.

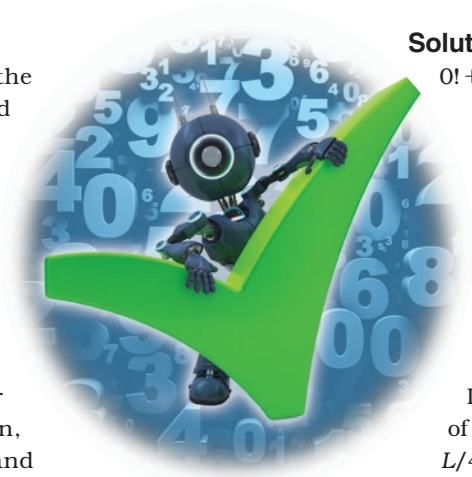
Digital Object Identifier 10.1109/MPOT.2018.2847858
Date of publication: 6 September 2018

Solution #1: Flow Rider

The oarsman's velocity relative to the water is constant. Since he rowed away from the log upstream for 1 h, he rowed downstream for 1 h before he returned to it. During those 2 h, the log moved 2 km, so its velocity relative to the land was 1 km/h.

Solution #2: Der Vampyr

One way to identify the undead sister is to consider all four possibilities for Anna (sane human, insane human, sane vampire, or insane vampire) and write down for each case what her answer requires Betsy to be. Then do the same for Betsy, and find the statements in the two lists that do not contradict each other. There are two pairs, both of which show that Anna is the vampire. The sisters are either both sane or both insane.



NUMBERS—© CAN STOCK PHOTO/123DARTIST, ANDROID—© CAN STOCK PHOTO/KIRSTYPARGETER

Solution #3: It's the Magic Number

$0! + 0! + 0! = 1 + 1 + 1 = 3$.

Solution #4: Belly Up to the Bar

To tell which iron bar is a magnet, make a "T" with them. They stick together when the nonmagnetized bar is the crossbar but not when the magnet is the crossbar.

Solution #5: Tunnel Vision

If the engineer runs to the north end of the tunnel, she travels the distance $L/4$, while the car travels the distance x. If she runs to the south end, she travels $3L/4$, while the car goes $x + L$. Subtracting shows that she could run the distance $L/2$, while the car goes L. Her speed is, therefore, 20 mi/h or 3 min/mi.

Digital Object Identifier 10.1109/MPOT.2018.2846899
Date of publication: 6 September 2018

P

YOU CAN BE THE NEXT STUDENT EDITOR OF IEEE POTENTIALS!

Apply by 24 December 2018

The IEEE is seeking a qualified Student Member or Graduate Student Member to serve as the Student Editor of IEEE Potentials If you'd like to apply, simply complete and mail this application. Be sure to include a copy of your resume and your answers to the essay questions. We look forward to hearing from you.

Don't forget to submit this information and a resume with your application!

REQUIREMENTS

- IEEE Student Member or Graduate Student Member (graduate or undergraduate level) for the time of service (two years).
- Strong interest in the professional, social, and technical concerns of students.
- Desire to see those concerns addressed in IEEE Potentials in an objective, timely fashion.

BENEFITS

- Communicate ideas to students.
- Work with other IEEE members from academia and industry.
- Show prospective employers your professional involvement.

RESPONSIBILITIES

- Provide input and feedback on the editorial content of IEEE Potentials.
- Write an editorial for each issue.
- Read and evaluate articles.
- Serve on the IEEE Student Activities Committee (SAC), which oversees the student programs and IEEE Student Branches established at leading universities and colleges worldwide.
- Attend two IEEE Potentials editorial board meetings and two SAC meetings each year.

ESSAY QUESTIONS

Please answer the following essay questions (in a separate document):

1) In 50 words or less, explain why you would like to be the next Student Editor of IEEE Potentials.
2) As Student Editor, what would your goals and objectives be?
3) List three areas of major concern to student engineers and technology professionals. In approximately 200 words, explore one area in detail.
4) List four editorial topics you would write about as IEEE Potentials Student Editor.

Mail or e-mail to: IEEE Potentials, 445 Hoes Lane, Piscataway, NJ 08854 USA E-mail: potentials@ieee.org

Name _____

Mailing address _____

City _____ State _____ Zip _____

Country _____

Permanent address _____

City _____ State _____ Zip _____

Country _____

Phone number _____

E-mail _____

I am an undergrad _____ or a grad _____ student

Institution _____

Graduation date _____

Digital Object Identifier 10.1109/MPOT.2018.2864427

Analyzing authentication patterns for cloud services

Krishani Liyanaarachchi

©ISTOCKPHOTO/HYWARDS

Cloud services are very popular among enterprises; most businesses use multiple cloud services. At the same time, some enterprises acquire, merge, and partner with other companies. It is at that point where one party will need to access the other party's resources in the cloud as well as its private cloud services. Due to these complex connections, cloud services must deal with different authentication protocols, multiple heterogeneous user stores, and legacy identity systems. As a result, implementing authentication in the cloud has become riskier and more complex.

Currently, the most widely adopted authentication patterns are direct authentication, brokered authentication, federated authentication, and single sign-on. These patterns have their own limitations when it comes to cloud authentication, which will be discussed in detail.

Cloud computing and its benefits

Cloud computing is an evolving paradigm providing software, platform, and infrastructure as services that are readily available, on demand, over the Internet. Users can consume these services on a pay-per-use basis. Cloud computing is also known as *utility-oriented computing*, meaning that software and hardware resources are available for users, much like other utilities such as electricity and telephone.

There are three types of cloud service models available, which are illustrated in Fig. 1. The first type is software as a service (SaaS), which allows users to access software applications over the Internet. Examples of SaaS include Gmail, Salesforce,

Digital Object Identifier 10.1109/MPOT.2015.2451677
Date of publication: 6 September 2018

0278-6648/18©2018IEEE

Twitter, and Facebook. The second type is platform as a service (PaaS), which provides platforms and environments such as operating systems, database management systems, and server software to allow developers to build applications and services over the Internet. Examples of PaaS are Microsoft Azure, Google App engine, and Force.com. The third type is infrastructure as a service (IaaS), which offers access to computing resources such as virtual server space, network connections, bandwidth, and Internet Protocol addresses in a virtualized environment through the Internet. Examples of IaaS are Amazon web service, Google Compute Engine, and Rackspace.

The aformentioned cloud services offer a number of benefits over the traditional approach of running in-house software. These include

- conserving capacity, since capacity is supplied as required by the user
- no in house maintenance, since there is no hardware environment
- fast deployment of new services
- the ability to access data from anywhere
- ease of sharing data.

Perhaps the most important advantage of cloud services is that they reduce the initial costs and maintenance of information technology infrastructure resources and enable businesses to go into production in less time. This leads to cloud services becoming very attractive for start-up and medium-scale business as well as fast-growing, large enterprises.

Security challenges in cloud services

While cloud services provide significant advantages, security in the cloud has become a pressing concern in enterprise software markets worldwide. Trusting cloud providers to keep your valuable data secure can be problematic. The cloud is like a double-edged sword: in the same way it provides as much resources as required by businesses on demand, it is also vulnerable to large-scale attacks on valuable data. Therefore, it is important to pay attention to security, trust, and privacy challenges from the design level of the cloud services themselves. According to Khalid, there are three types of high level security challenges in cloud services: data protection, isolation, and availability; identity management (user authentication, authorization, auditing), and physical security (disaster and data breach). In this article, we focus on user authentication as it is one of the crucial challenges; cloud services must make sure that the legitimate user from the enterprise is accessing authorized enterprise data in the cloud.

Every cloud service has implemented an authentication mechanism for accessing its services in a secured manner. But in most cases, cloud service providers request enterprises to store their account information or their employee's details in the cloud, which contains credentials to access the cloud services as a legitimate user. Cloud service providers could access this information, and this presents a privacy issue to the customer's privacy information. Many software license agreements have specified the privacy of the sensitive information. However, it is difficult for customers to confirm that the proper rules are enforced. There is a lack of transparency in the cloud that allows customers to monitor their own privacy information. If the cloud service provider is compromised, the account information of customers could be handed to attackers.

When enterprises subscribe to more cloud services, multiple copies of users' information spread throughout the Internet. This is a security issue for enterprises as well as cloud service providers. If one service provider is compromised, other providers could be easily attacked. For every cloud service to which one subscribes, users may need to keep track of multiple authentication credentials and exchange their authentication information via the Internet.

One solution to the aforementioned authentication issues in the cloud is that different authentication patterns have been adopted in cloud services. Of these patterns, brokered authentication with single sign-on (SSO) pattern and federated authentication pattern are the most common and popular patterns over cloud services. These patterns could completely eliminate the storage of user credentials within cloud services and transport them through the Internet. Currently, there is the rapid growth of some enterprises with acquisitions, mergers, and partnerships. Therefore, more private and public cloud services have started to be consumed by enterprise employees. Users from several organizations merge with enterprises to work in a collaborative manner—cloud services are no longer limited to

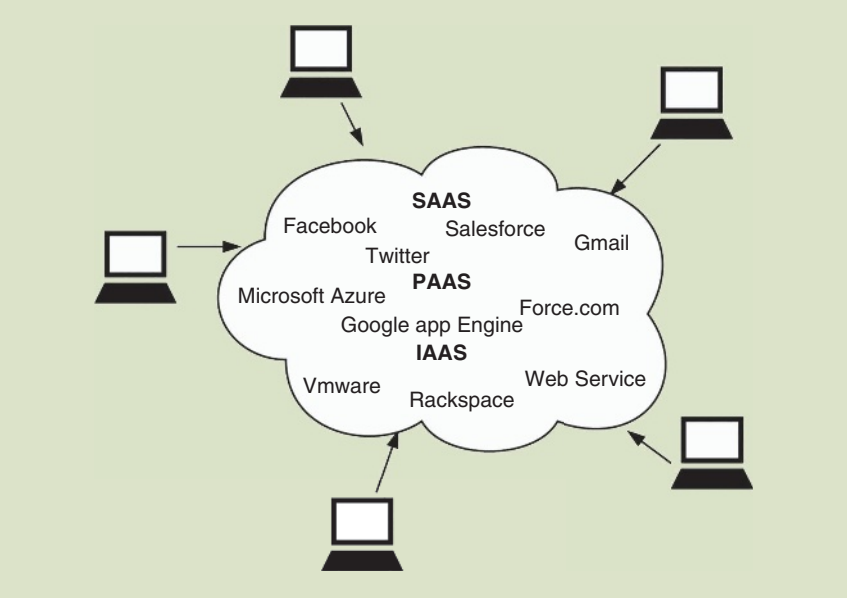


FIG1 The three types of cloud service models: Saas, PaaS, and IaaS.

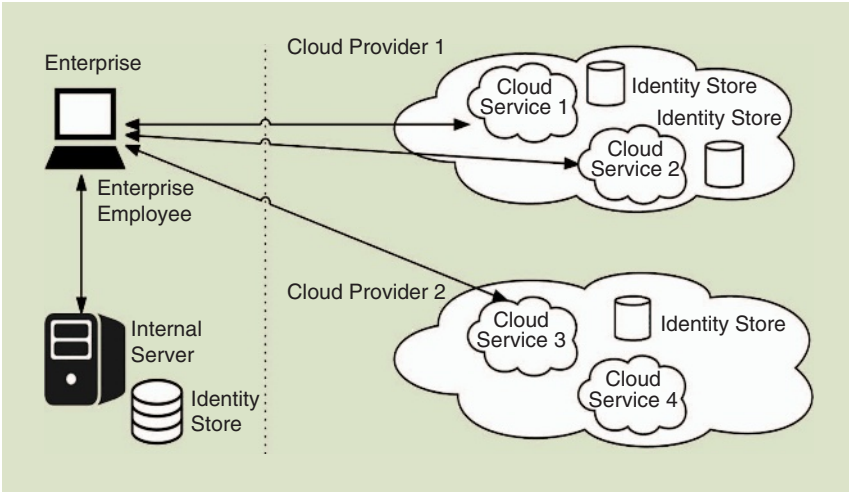


FIG2 A direct authentication pattern.

employees; users from outside a company could also access them. Therefore, enterprises and cloud service providers need to work with multiple heterogeneous user stores, different authentication protocols, and legacy systems, among others, which makes authentication a more complex task.

Direct authentication

Direct authentication is the basic, and most common, way of protecting cloud resources against attackers. Specific groups of users are permitted to access cloud services, and those users are identified by the credentials that are provided to the service. The cloud service is implemented to access an identity store that contains a user's details including his or her credentials. The service can validate the users directly with provided credentials.

Direct authentication involves the following participants:

- Client: the enterprise user who accesses the cloud service through the Internet. The client provides the credentials for authentication during the request to the cloud service.
- Service: the cloud service that validates user credentials with respect to the identity store and only provides access to cloud resources for legitimate users.
- Identity store: the entity that stores a client's credentials and user details for a particular identity domain.

In Fig. 2, we illustrate this mechanism. Assume that the client is an employee of an enterprise. The client can access internal services within the enterprise by providing valid credentials. An internal service validates the employee's credentials using the enterprise identity store. This identity store is not exposed to outside forces and is only maintained by company administrators. Therefore, the internal system cannot be compromised by external attackers outside the enterprise.

Assuming cloud service providers support only direct authentication, each cloud service needs an identity store that is used to maintain the user's credentials. If an employee of an organization needs to access a cloud service, he or she must be registered in the cloud service with his or her credentials. The employee could then access the cloud services by providing his or her registered credentials. This may lead to the following security risks.

1) Credentials of the employees are maintained in more than one place where the enterprise has no control of them. The company needs to trust all cloud providers, which may be impossible due to the large number of cloud services.
2) When accessing cloud services, the credentials of employees are transported through the Internet, which creates exposure to external attackers.
3) Many employees use the same credentials (e.g., a similar password) for all cloud services, as it is easy to remember. If one cloud provider is compromised (e.g., a password leak attack), it can affect all cloud providers and the enterprise data that are in the cloud.
4) Due to the credentials policies (e.g., username/password) of different cloud providers, and as a best practice, employees must use different credentials for each cloud provider, which makes it a headache for employees to maintain their own credentials. This may also lead to a security risk, as employees may use unsecure practices to remember multiple passwords (e.g., keeping a paper record of all passwords).

Most cloud services have moved away from the direct authentication approach due to the aforementioned security risks. However, the direct authentication option is still available for some cloud providers, as it is easy to implement without much cost and complexity.

Brokered authentication

To overcome security risks due to direct authentication, the brokered authentication pattern has been widely adopted by most cloud providers. In direct authentication, there is a direct relationship between the two parties, the client (enterprise employee) and the server (cloud service). In brokered authentication, there is an authentication broker that both parties trust independently. The authentication broker issues a security token to the client, where the client can present them to the cloud services for accessing resources.

We illustrate this mechanism in Fig. 3. Assume that the client is an employee of an enterprise and an authentication broker is installed.

1) The enterprise employee submits an authentication request to the authentication broker.
2) The authentication broker contacts the identity store to validate the employee's credentials with respect to the enterprise identity store.
3) If authentication is successful, the broker issues a security token.

This token can be used to authenticate with the cloud service.

4) A request message is sent to the cloud service; it contains the security token that is issued by the authentication broker.
5) The cloud service authenticates the request by validating the security token.
6) If security token validation is successful, the cloud service returns the response to the employee.

Brokered authentication involves the following participants:

- Client: the enterprise user who accesses the cloud service through the Internet. The client provides the credentials for the authentication broker during the request to the cloud service.
- Service: the cloud service that validates the security token and provides access to cloud resources only for legitimate users.
- Authentication broker: validates user credentials with respect to the underlying identity store and issues security tokens. It maintains the identity information of users and controls security tokens.
- Identity store: stores a client's credentials and user details for a particular identity domain such as an active directory or a Lightweight Directory Access Protocol server.

The main benefit of a brokered authentication pattern is that it provides a centralized architecture for authentication. Therefore, an authentication broker manages trust centrally. This eliminates the need for a client and the cloud service to manage trust independently. As a result, the following advantages can be gained when compared to direct authentication:

1) The credentials of enterprise employees can be kept in an internal identity store where they are managed by the organization's administrators.
2) Credentials are not transported through the Internet and not exposed to external attackers.
3) It avoids multiple credentials for employees.
4) The prior registration of employees' credentials and details is not required. This enables fast access to cloud services.
5) Deprovisioning is not needed, as employees' information and credentials are maintained in one place.
6) If the same security token can be used for multiple cloud services, SSO can be achieved between cloud services.

Some limitations of the brokered authentication include the following.

- Due to the centralized architecture used by brokered authentication, a single point of failure can be created. As a result, if the authentication broker is compromised, all cloud services that trust the authentication broker are in danger.
- A cloud provider must use a mechanism to trust the security token issued by the authentication brokered. Different types of security tokens and protocols can be used by the authentication broker. The cloud provider needs to support communication with any type of authentication broker, which can be complex and not scalable.
- A cloud service needs to trust multiple authentication brokers for each enterprise. If an authentication broker in an enterprise is compromised, it would raise some issues for other organizations that use cloud services.

Due to these security issues and scalability problems, most public cloud providers do not trust security tokens that are issued by every authentication broker; they only trust the security tokens that are issued by specific authentication brokers. If enterprises create some partnership with some other enterprises, then they may need to share the data in their private cloud as well. A private cloud usually trusts only the authentication broker that resides in its enterprise domain.

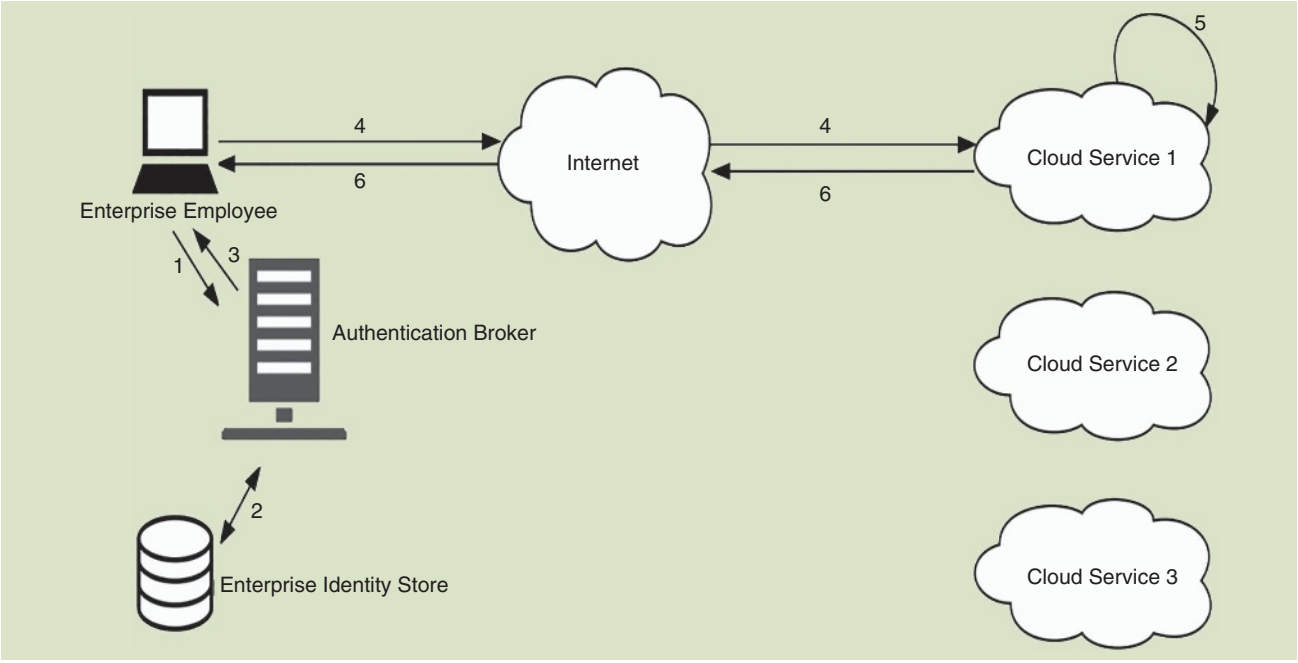


FIG3 The brokered authentication pattern.

Federated authentication

In a brokered authentication pattern, cloud services must trust multiple authentication brokers, which can lead to security risks and scalability issues. As a result, cloud providers are moving toward trusting only some authentication brokers with which a mutual understanding has been agreed upon. Essentially, cloud services do not accept security tokens issued by multiple authentication brokers, and they trust only those tokens issued by their own authentication broker or a trusted broker. Therefore, company employees must exchange the security tokens that are received from an enterprise authentication broker or identity provider (IDP) with the cloud provider's authentication broker or identity provider. We illustrate this in Fig. 4. The following steps are taken:

1) The enterprise employee submits an authentication request to the authentication broker.
2) The authentication broker contacts the identity store to validate the employee's credentials with respect to the enterprise identity store
3) The authentication broker issues a security token.
4) The received security token is posted to the cloud authentication broker.
5) The cloud authentication broker validates the security token.
6) If validation is successful, the cloud authentication broker issues a new security token to the client.
7) A request message is sent to the cloud service; it contains the security token that is issued by the cloud authentication broker. The cloud service authenticates the request by validating the security token.

This pattern provides several advantages.

- Cloud services do not want to trust multiple authentication brokers. It eliminates security risk, and the trust relationship between cloud services and brokers can be clearly defined. This also mitigates the implementation complexity for cloud services.
- Multiple cloud services can trust one cloud authentication broker. Therefore, the registration process for cloud services is much easier. An enterprise can build a trust relationship between the cloud authentication broker and the enterprise authentication broker. Company employees could seamlessly access multiple cloud services that have been registered with the cloud authentication broker.
- The cloud authentication broker can transform the security tokens into a format that service providers can decode and validate. Claim transformations are mostly handled by cloud authentication brokers (e.g., a security token from an organization may present the user's e-mail address in a claim called *email*, while in cloud services, it is presented in the claim *e-mail*). Therefore, claim transformation must be done by cloud authentication brokers.

Single sign-on

As previously discussed regarding brokered authentication patterns, if an authentication broker can issue security tokens that can be used by multiple cloud services, or if an authentication broker can maintain a user session, SSO can be achieved. It provides a seamless access to cloud services as well as internal services. Simply put, SSO allows users, who are authenticated against one cloud service, to gain access to multiple cloud services without repeated authentication by providing credentials. By maintaining the authenticated user's session inside authentication brokers, both brokered and federated authentication patterns can be converted to an SSO pattern.

Many cloud providers support SSO. Examples include the web-browser-based SSO based on SAML2 and OpenID standards: these are popular in SaaS and PaaS providers. Figure 5 describes an SAML2 web browser SSO flow in detail.

Participants

Participants are the same as broker authentication but the referring names are different:

- client: the enterprise user
- service provider: the cloud service that validates security tokens and

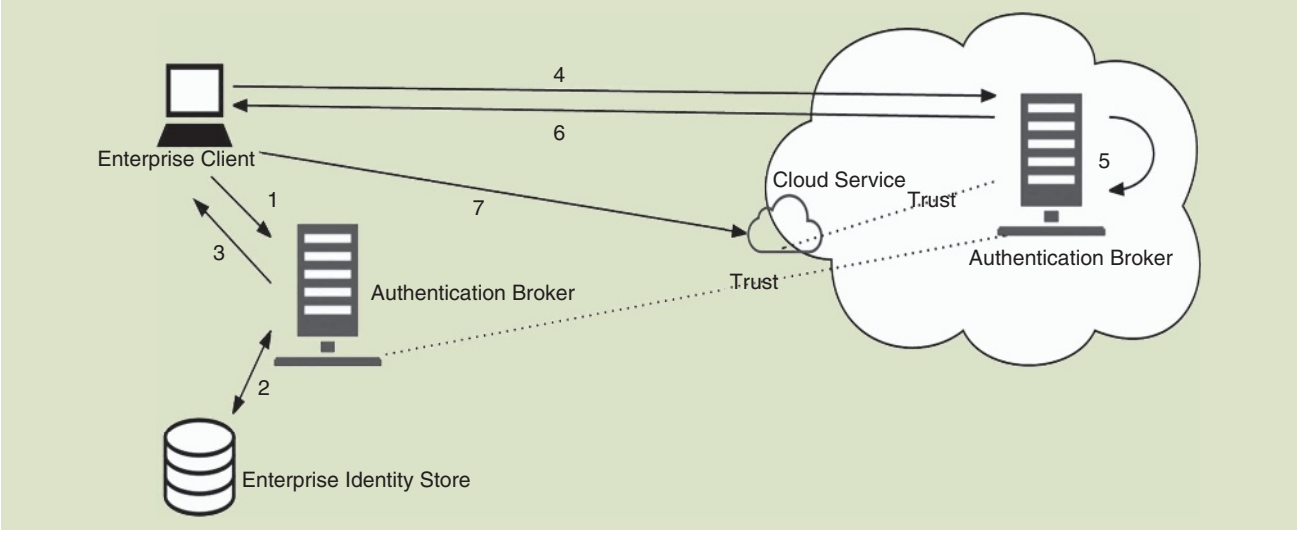


FIG4 A federated authentication pattern.

only provides access to cloud resources for legitimate users
- IDP: the authentication broker that validates user credentials with respect to the underlying identity store and issues security tokens
- identity store: the entity that stores a client's credentials and user details for a particular identity domain.

We illustrate this in Fig. 5, where the following steps are taken:

1) A user visits SP1.com to access the SP1 service.
2) The user is redirected to the IDP.
3) As the user is not authenticated, the IDP is asked to enter the username and password.
4) The user submits his/her username and password.
5) If authenticated, the user will be directed to SP1.com.
6) The user visits SP2.com to access the SP2 service.
7) The user is redirected to the IDP.
8) As the user is already authenticated, he/she is redirected to SP2.com

Cloud identity bus pattern

There are several standards that have been developed for identity federation, such as SAML2, OpenId, OpenId-connect, WS-Trust, and WS-Federation, among others. All of these standards can be used to support the brokered, federated, and SSO authentication patterns. However, it is difficult to expect that all parties in the federation domain (enterprise client, enterprise IDP, cloud IDP, cloud services) support all of these standards. Most of the time, they support one or several standards. As an example, IDP in the enterprise supports the SAML2 web SSO standard only. As a result, cloud IDP needs to support the SAML2 web SSO standard to talk with enterprise IDP. If both do not support the same standard, authentication cannot take place.

It is not practical for enterprises and cloud providers to support all federation standards; supporting more standards means more cost. Moreover, authentication is just one small fraction of the functionalities of organizations and cloud systems when compared with their actual deliverables. For example, Google, Salesforce SaaS supports the SAML2 web SSO profile. Facebook supports OpenID Connect. Twitter supports OAuth. Microsoft SaaS supports WS-Federation. Therefore, we can see federation groups such as SAML2 federation, OpenId-Connect federation, OpenId federation, etc.

Even in a given federation group, there may be many IDPs and cloud services. Each cloud service and enterprise must trust each other. This creates many chains of trust relationships, which are very complicated, and is typically referred to as *spaghetti anti-pattern*.

Thus, even when using a federated authentication pattern, there are two main limitations:

1) complexity in trust management, illustrated in Fig. 6
2) communication issues due to multiple standards, shown in Fig. 7.

As a solution to the aforementioned limitations, a new pattern called the *cloud identity bus pattern* has been proposed. It is an architectural pattern that can be used for implementing communication between enterprises and cloud providers in the cloud environment.

Cloud identity bus is a separate cloud service that provides support for the cloud identity bus pattern and takes care of the overall authentication of the different cloud providers and enterprises. Cloud identity bus pattern provides solutions to the aforementioned limitations, as follows.

1) A given enterprise or cloud provider is only coupled with the cloud identity bus, thus eliminating the need to couple with every cloud provider or enterprise. Therefore,

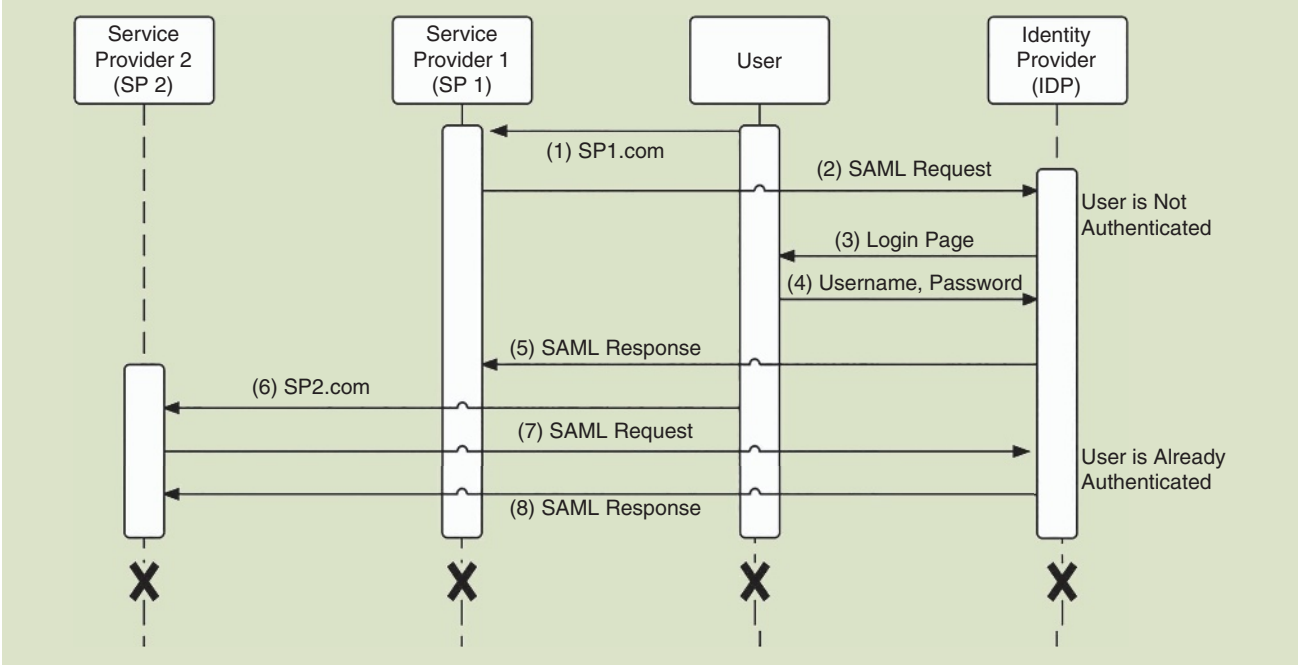


FIG5 An SSO.

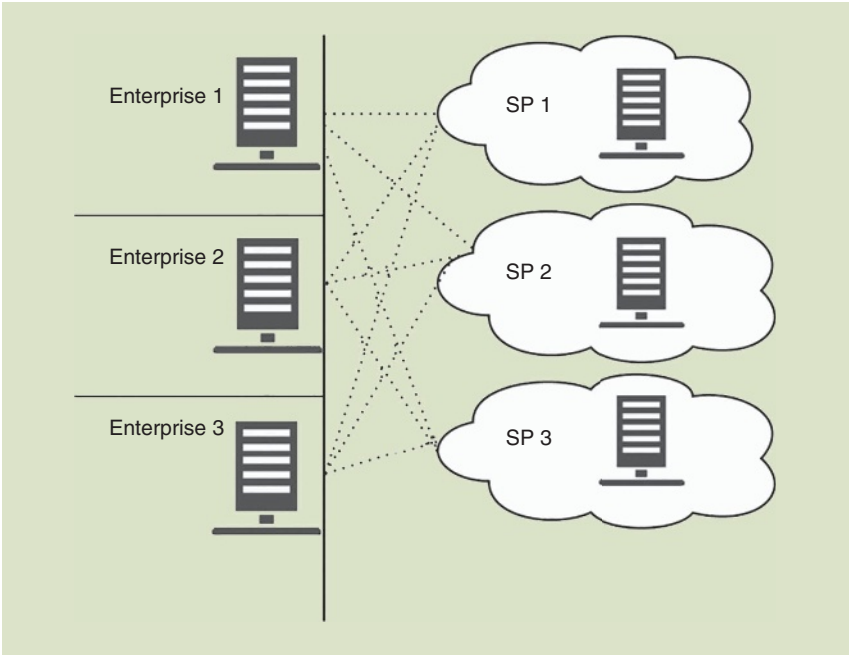


FIG6 The spaghetti anti-pattern.

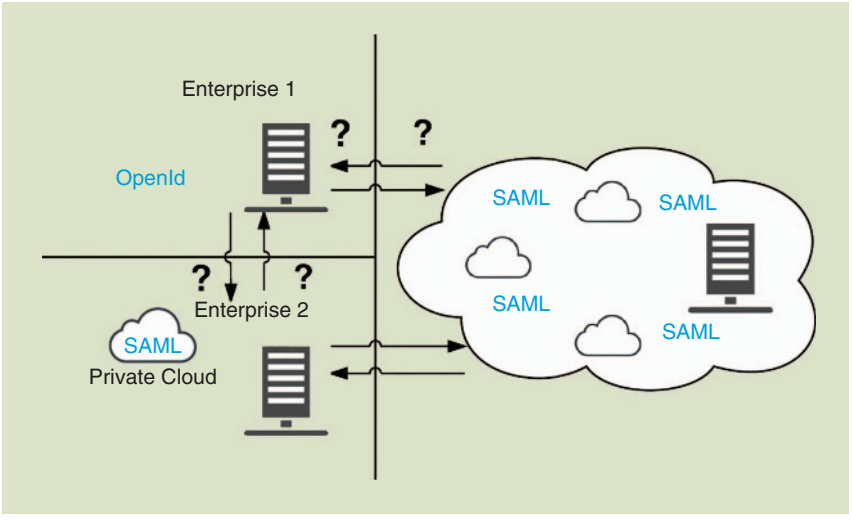


FIG7 Communication issues for different protocols.

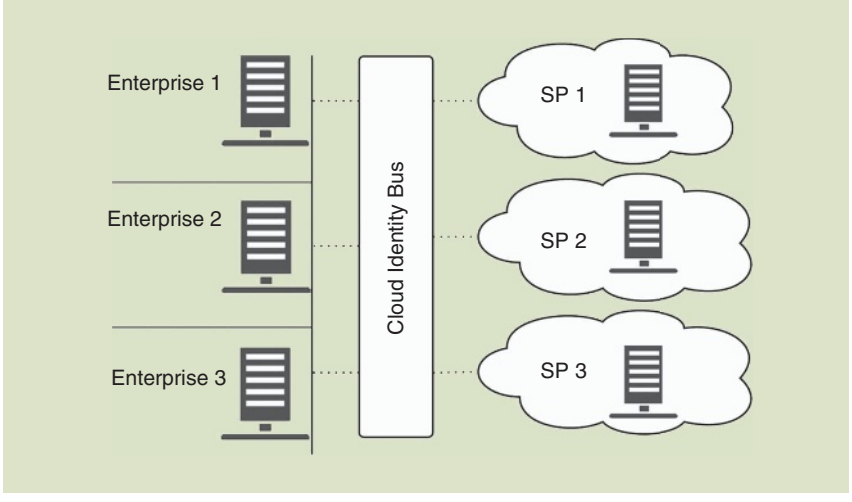


FIG8 The cloud identity bus pattern can solve it.

trust complexity can be eliminated, as shown in Fig. 8.

2) The cloud identity bus supports multiple standards and has the capability to transform protocols from one standard to another. For example, if an enterprise's IDP supports the SAML2 SSO web standard and the cloud provider embraces the OpenId-connect standard, then the cloud identity bus acts as the middleman who mediates and transforms different security tokens with different standards. This is illustrated in Fig. 9.

The high-level components and necessary communication flows are shown in Fig. 10. These include:

- In-bound authentication: comes to the cloud identity bus from cloud providers
- Out-bound authentication: the authentication request going to the enterprise IDP from the cloud identity bus.
- The given cloud provider must be registered in the cloud identity bus, and the in-bound authentication configuration for the cloud provider is defined by the cloud administrator. The cloud provider can publish the cloud services that are supported as a description of the configuration.
- The given enterprise must be registered in the cloud identity bus, and the outbound authentication configuration for the enterprise is defined by the enterprise administrator. If the enterprise supports a private cloud that has been exposed to several other enterprises, the enterprise administrator can establish an in-bound authentication configuration for the private cloud.
- While registering the enterprise, the enterprise can select the public cloud services that it needs to access. But, in some cases (such as for a private cloud), there is mutual agreement between the cloud provider and the enterprise that mentions which cloud services are accessed by the enterprise.
- Based on the agreements or selections of enterprises, the cloud

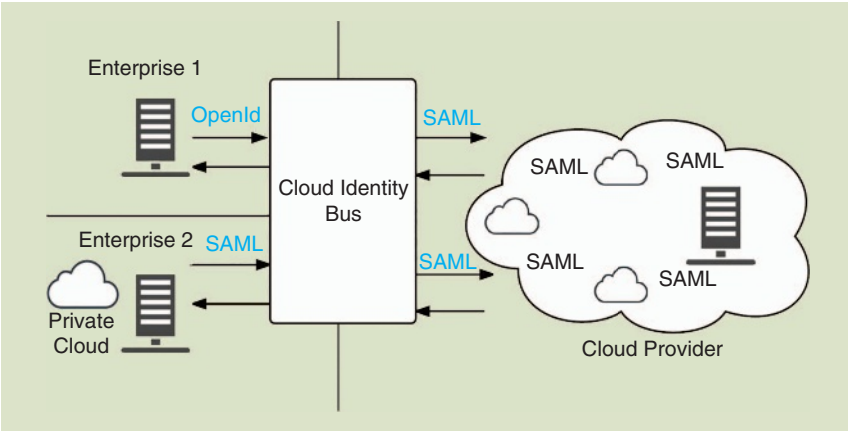


FIG9 The cloud identity bus pattern can perform protocol switching.

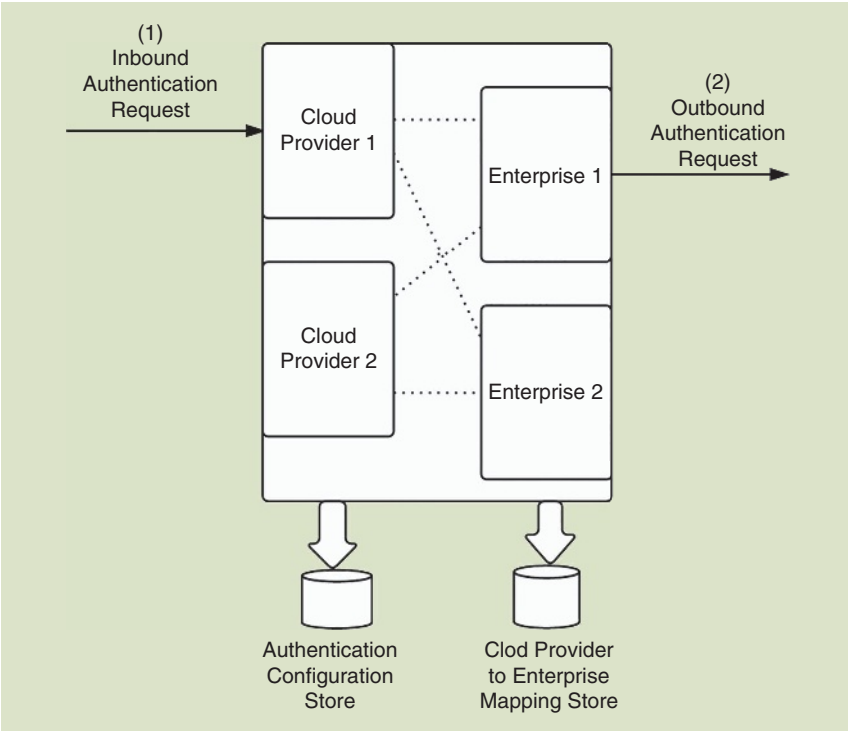


FIG10 A model of the cloud identity bus.

provider picks and configures the enterprises which need to access its cloud services.

- The cloud provider to enterprise mapping would be maintained in the cloud identity bus.
- Based on the cloud provider to enterprise mapping, the cloud identity bus can mediate authentication requests.
- Based on the authentication configurations, the cloud identity bus can execute the protocol transformations.

Conclusion

In this article, we analyzed the authentication patterns commonly found in current cloud applications. We identified four common authentication patterns and discussed their advantages and disadvantages while considering the needs of enterprises. The direct authentication pattern has already depreciated, but it offers a good starting for an authentication pattern for cloud applications with minimum initial cost.

In addition, IaaS providers are still interested in the direct authentication pattern. Brokered, SSO, and federated authentication patterns are currently widely adopted. These patterns are similar to each other, as all patterns are based on a third-party, trusted IDP/broker. However, each pattern has its advantages and disadvantages. Based on the analysis, the federated authentication pattern is the most suitable and secured pattern for cloud-based applications. However, it must be supported so it integrates with different IDPs/brokers in a seamless manner.

Finally, due to multiple authentication protocols and standards and the large number of IDPs/brokers in the cloud, it is not easy to achieve seamless authentication with cloud services. Therefore, the cloud identity bus pattern has been introduced. We discussed how the cloud identity bus pattern can solve the protocol/standard mismatching and spaghetti anti-pattern.

Read more about it

- S. Zhang, H. Yan, and X. Chen, "Research on key technologies of cloud computing," *Phys. Procedia*, vol. 33, pp. 1791–1797, 2012.
- G. Pallis, "Cloud computing: The new frontier of Internet computing," *IEEE Internet Comput.*, vol. 14, pp. 70–73, Sept. 2010. doi: 10.1109/MIC.2010.113.
- U. Khalid, A. Ghafoor, M. Irum, and M.A. Shibli, "Cloud based secure and privacy enhanced authentication & authorization protocol," *Procedia Comput. Sci.*, vol. 22, pp. 680–688, 2013.
- H. Y. Huang, B. Wang, X. X. Liu, and J.M. Xu, "Identity federation broker for service cloud," in *Proc. IEEE Service Sciences Int. Conf.*, 2010.
- D. N. Sriram, "Federated identity management in intercloud," M.S. thesis, Technical University of Munich, 2013.
- E. Maler and D. Reed, "The Venn of identity," *IEEE Security Privacy*, vol. 6, no. 2, pp. 16–23, 2008.
- S. Wang. (2011). An analysis of web single sign-on. [Online]. Available: http://blogs.ubc.ca

About the author

Krishani Liyanaarachchi (krishliyana@gmail.com) is currently a master's degree student who is studying computer science at Åbo Akademi University, Finland.

P

Mobile cells assisting future cellular communication

©ISTOCKPHOTO/METAMORWORKS

Syed Shan Jaffry, Syed Faraz Hasan, and Xiang Gui

Technology has brought miraculous advancements in the way we communicate, shrinking distances and eliminating time-bounds. The wireless version of telecommunication has particularly made data exchange more convenient than ever. The most common technology used for wireless data exchange is cellular technology, as evident by the ever-growing number of mobile phone users.

The first version of cellular technology—the first generation (1G)— came out in the 1980s. Second-generation and third-generation (3G) networks followed soon after and have now been replaced, in most countries, by fourth-generation (4G) technology. It is necessary to periodically update wireless technology because each turn of the decade experiences a massive evolution in end-user demands. For example, the 1G network was designed for voice calling and text messaging only. In more recent years, 3G and 4G technologies are required to sustain a huge amount of network traffic to support applications like online gaming, video calling, and multimedia streaming.

The increase in network traffic is also driven by the fact that smartphones have become a new way to stay connected. In line with other industry giants, a survey conducted by Nokia reported that estimated network traffic will be 10,000 times greater by 2020 as compared to 2010. Apprehension that the existing cellular infrastructure will be overwhelmed by a mammoth amount of traffic has paved the way for the introduction of the fifth generation (5G) of cellular technology.

What will 5G look like?

In existing cellular networks, base stations serve as central entities that liaison communication between users. The base station is referred to as the *evolved Node-B* (*eNB*) in long-term evolution nomenclature. An eNB (or a set of eNBs) wirelessly connects one user with another. The coverage region of an eNB is called a *cell*, which is typically 1 km in radius. A cellular network is, therefore, a network of interconnected cells or eNBs as shown in Fig. 1. This traditional eNB-centric architecture is a potential bottleneck if the amount of data increases with the predicted rate.

Thus, a 5G network calls for a number of changes in this eNB-centric design. From an architectural viewpoint, 5G prefers a distributed mechanism in which

Digital Object Identifier 10.1109/MPOT.2018.2816964
Date of publication: 6 September 2018

0278-6648/18©2018IEEE

mobile phones share data without frequently engaging the eNB. Instead, it is envisaged that the 5G network will comprise groups of small- to medium-sized networks employing multiple wireless technologies, each tailored to serve a specific user group. Such an anticipated network is called the *heterogeneous network* (*HetNet*), which will allow the provision of customized services to diverse user requirements. The cells within a HetNet should have a smaller size and lower transmit power. These so-called small cells (SCs) are meant to serve densely populated regions like malls, stadiums, or a group of people traveling inside a public vehicle.

An SC network

The users within an SC communicate through an SC-eNB, which is then connected to the core-network through the Internet. Neighboring SC-eNBs communicate with each other via sidehaul links. The users inside an SC, like a bus, connect with an SC-eNB via access links. It is important to take into account the use of SCs for vehicular users because more than 76% of the people who use public transport also use smartphones while traveling, according to the 2015 Mobile Consumer Survey of Australia. Similar trends have been observed in other big cities, for example, Shanghai (84%), London (89%), and New York (93%), as shown in Fig. 2. These on-the-go users have become so important that a new area of research called the *mobile cell* (*MC*) has recently emerged that addresses the needs of vehicular users. The future cellular environment with MCs is shown in Fig. 3.

Cells that move!

By definition, MC groups vehicular users and connects them to a dedicated MC-eNB that is placed in-vehicle (a bus or a train). The MC-eNB serves as a gateway between vehicular users and the external network. An MC is a nonstationary version of an SC. One difference between an MC-eNB and an SC-eNB is that the former has a wireless link with the core-network, while the latter is typically connected over wired connections to the network.

The main advocacy for MCs comes from an Ericsson ConsumerLab report that states approximately 55% of the vehicular users on board New York public transport are dissatisfied with the quality of service (QoS) they receive. This user dissatisfaction is also felt across other major cities in the world, which stems from the fact that wireless signals lose power while penetrating through a vehicle's body. Consequently, when the signals are received at a lower strength, they are not correctly interpreted by

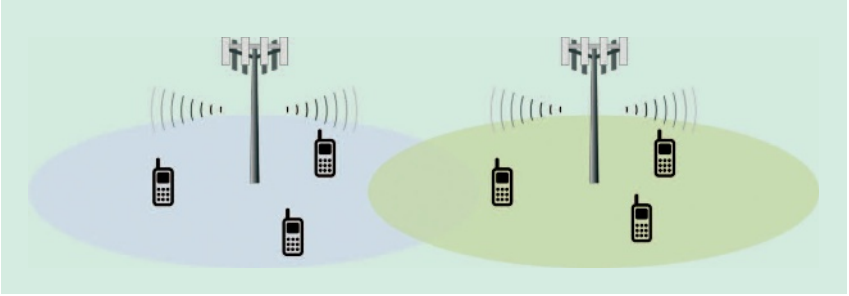


FIG1 Conventional cellular architecture.

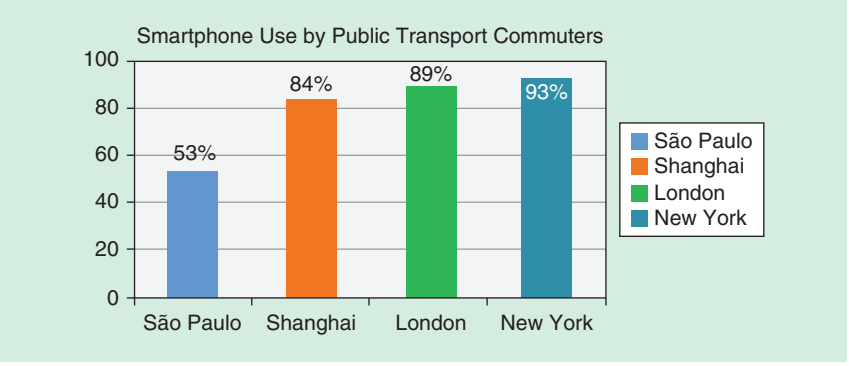


FIG2 Smartphone use by public vehicular users (Ericsson ConsumerLab report, 2014–2015).

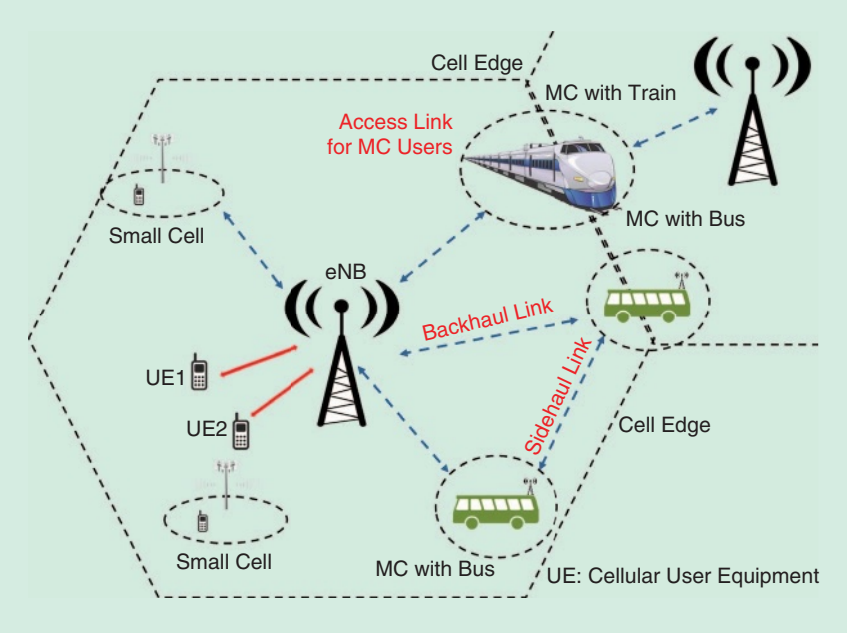


FIG3 The future cellular environment with mobile cells (MCs).

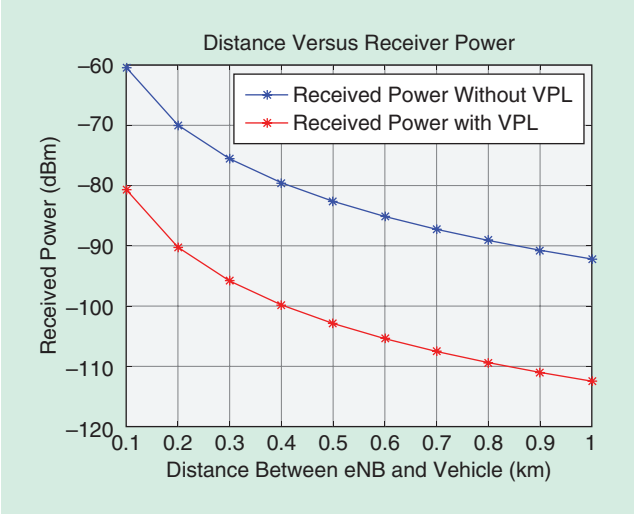


FIG4 Received power with and without VPL compensation.

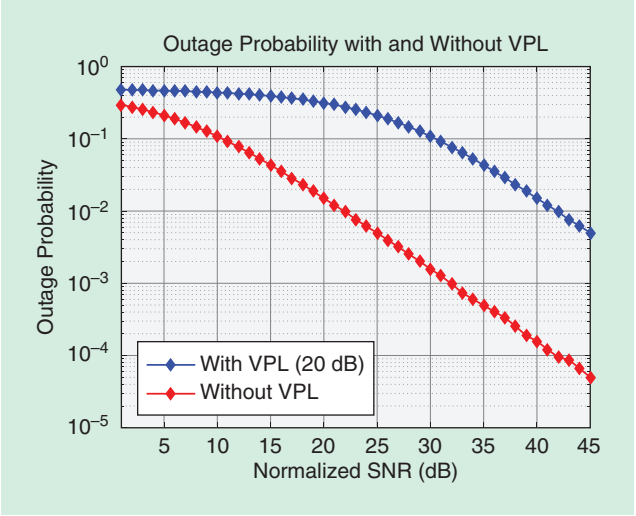


FIG5 Outage probabilities with and without VPL compensation.

the receiver. This so-called vehicular penetration loss (VPL) is typically measured in terms of the signal-to-noise ratio (SNR) and results in poor call quality, frequent dropped calls, and slow network speed.

A typical effect of VPL can be understood by reviewing Fig. 4, which demonstrates that as the vehicle moves away from the eNB, the users traveling inside experience lower signal power. However, if VPL is compensated by using an MC, the received power increases by a factor of 20 on the decibel scale (equal to 100 times on linear scale). This improved signal quality guaranteed by the MC-eNB ensures that the probability for a user to get disconnected from the network, also called *outage probability* (*OP*), is low. As a general rule, the higher the SNR, the lower the outage probability. This is demonstrated in Fig. 5. Consequently, low outage probability results in better service experience for users.

Handover assistance with MC in 5G

Another critical issue for vehicular users is related to managing the inevitable handovers, which occur when users cross the boundary of one SC-eNB to enter a neighboring SC-eNB. The handover rate is an important parameter that exerts undesirable signaling overhead on the network. Handover management in the 5G network will become more complex in the presence of smaller-sized cells. As a rule of thumb, the handover rate is proportional to the number of active connections per vehicle, along with the vehicle speed. In this context, Fig. 6 demonstrates that MCs will considerably reduce the handover rate. Since all vehicular users in an MC will communicate through the MC-eNB, not all of them will require individual handovers when they leave the coverage of one eNB (or SC-eNB). Instead, only an MC associated to an eNB performs the handover which caters for several active several users that are on board. To simplify the discussion, consider N vehicles moving with a constant speed with ten cellular users per vehicle. The use of an MC-eNB will require only a single handover instead of ten, consequently increasing the spectral efficiency.

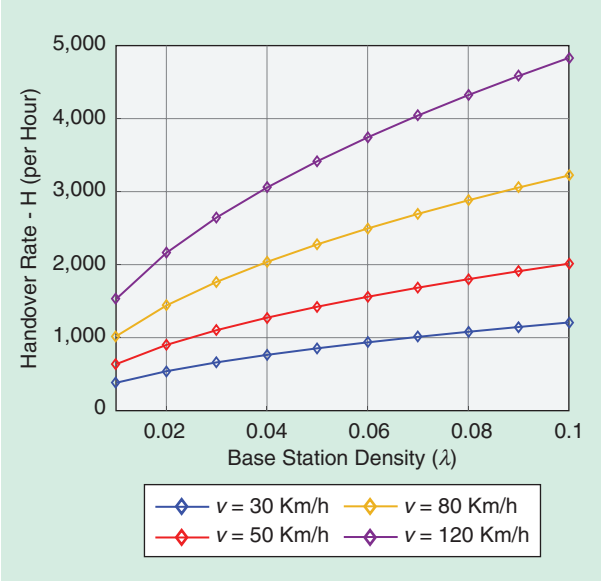


FIG6 The impact of vehicle velocity and base station (in per km^2) on handover rate [ten vehicles with ten active connections (average) per vehicle].

Offloading central network using MC

According to Cisco Visual Networking, the global mobile data traffic per month accumulated to less than 10 GB (giga = 10^9) before 2000. It skyrocketed to nearly 7.2 exabytes (EB) (exa = 10^{18}) by the end of 2016. It is projected that the monthly mobile data traffic will soar to 49 EB by 2021. This is shown in Fig. 7 as well. With the roll-out of several new resource-intensive and interactive applications, present-day eNBs will collapse if they relay all associated data transfer. The MCs can offload some of that burden by caching the most popular contents and sharing it with other MCs in close proximity. This will stop users from approaching the network for accessing popular content redundantly.

As such, an MC-eNB can serve as a dedicated Internet

data-sharing cache for the vehicular users. This idea is effective because the small amount of popular content over the web makes up a large proportion of the overall web traffic.

Research conducted by Jiang et al. using real-world data from the streets of Sweden in 2014–2015 advocated how the use of a dedicated cache on public buses could save more than 20% of data bandwidth. The same research also demonstrated that commuting time and routes affect the requested data from the Internet. It maintained that 85% of the requests made during peak hours originate from some specific routes. Using MCs to access popular content will reduce latency in accessing popular content. This concept of MC-cache is similar to the idea of cloudlets—a technology that brings mobile clients closer to the server. With proper caching algorithms in place, an MC-cache can spare invaluable frequency resources for network operators.

MCs in mobile black spots

In addition to improving QoS, MCs provide basic network services to users that end up in mobile black spots (MBSs). As shown in Fig. 8, a black spot can be a temporary out-of-coverage region, for example, when a natural disaster destroys network infrastructure (such as base stations). The Third Generation Partnership Project (3GPP)—an organization that governs cellular standards—has introduced Proximity Services (ProSe) to provide help for such calamity-hit regions. MCs that are within the coverage of a macro eNB can act as the first responders' communication services in emergency situations. Researchers in South Korea conducted simulations to demonstrate how an MC can be an effective way to connect devastated regions with the core-network using a band dedicated for public safety (the 700-MHz band).

Similarly, a black spot can be permanent if there is no network infrastructure set-up at all. Usually such regions are remote areas that do not have network coverage and are difficult to reach. Such areas are typically inhabited by a small population. Installing network infrastructures in these areas is not cost effective because the return on investment is quite low. However, these far-away regions still require wireless connectivity, especially to call for help when needed.

Several countries are responding to this need nationally. For example, the Australian government is spending AU$170 million (~US$126 million) in its Mobile Black Spot Program. Its neighbor, New Zealand, has introduced the Mobile Black Spot Fund (MBSF), which invests NZ$36 million (~US$25 million) in improving cellular services along the main highways and popular tourist destinations. By acting as repeaters for out of coverage users, MCs can cover the black spots in a more cost effective manner. However, a number of research challenges that must be addressed still exist.

Practical challenges and issues

Spectral issues

An MC has three kinds of links. The first is the direct link between the MC and vehicular users called the *access link*. The link over which neighboring MCs communicate with each other is the *sidehaul link* (MC-to-MC link). Finally, the MC connects to the network infrastructure over *backhaul links* (MC-to-core network). Given that the radio spectrum is an expensive commodity, cellular service providers will seek to integrate MCs within the available spectral resources. This is still an open question for researchers investigating MC technology. With the inclusion of the new frequency channels, as envisioned for 5G, one solution is to use separate bands for MC access-links and integrate sidehaul and backhaul communication in the available spectrum. The 3GPP has mandated to integrate sidehaul links with the existing uplink spectrum when no uplink channels are free. However, this will require efficient algorithms for spectrum sharing between the eNB and SC-eNB.

Traditionally, the spectrum-sharing decisions are made by the eNBs.

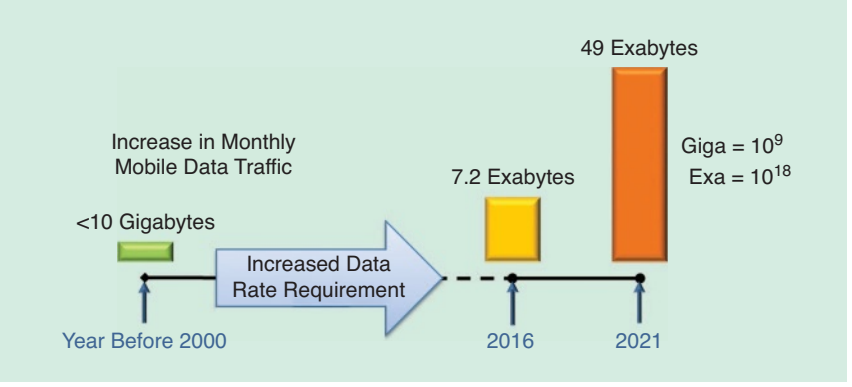


FIG7 The trend in increasing global monthly mobile data traffic (Cisco report). (Figure not drawn to scale.)

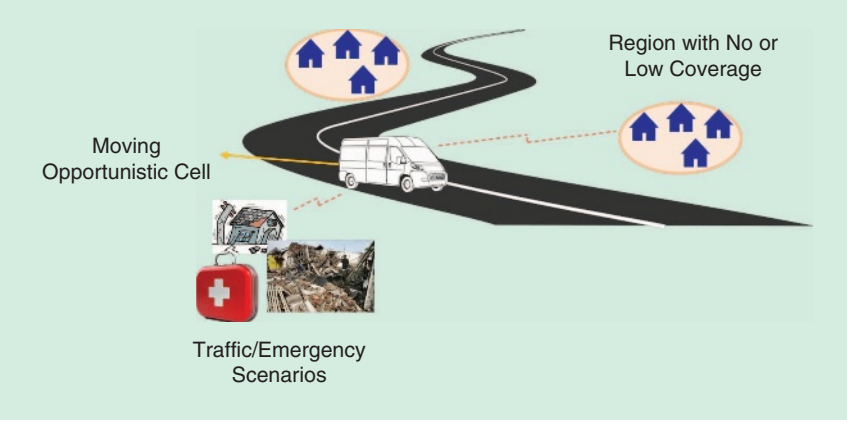


FIG8 First-aid response in disaster management and covering an MBS.

These decisions are based on the available spectral resources. With the rollout of MCs, in addition to the existing cellular links, managing MC communication links requires novel resource-allocation methods.

Synchronization and discovery

While MCs can provide services in MBS regions, synchronization between the MC and the user's devices and/or multiple MCs are critical for establishing communication links. Synchronization is necessary to ensure that the transceivers are well aligned in time for sending and receiving data. After synchronization, the initial step in communication is to discover neighboring MC nodes. For a fast-moving MC node, time-to-synchronize and time-to-discovery are important measures.

Synchronization and discovery methods in areas that have cellular coverage will be controlled by the network itself, according to 3GPP guidelines. However, areas where no cellular services are available, like in MBS regions, pose significant challenges. Without synchronization, seamless MC-to-MC communication is difficult to maintain. The 3GPP has defined initial guidelines for the out-of-coverage synchronization methods but practical synchronization and discovery procedures are still an open challenge for the research community and industry.

Inter-MC communication and connectivity

Adjacent MCs would communicate with each other over sidehaul links. For fast-moving MCs, resource scheduling for the sidehaul link and maintaining seamless connectivity are challenging tasks. The 3GPP mandates the use of unused frequency spectrum in the uplink cellular transmission for sidehaul communication. For example, when the uplink channels are idle, they may be used for MC-to-MC communication. Scheduling the inter-MC communication side by side with the uplink cellular transmissions is a computationally intensive task. However, it would offer seamless connectivity when assisted by the network. On the other hand, in out-of-network scenarios, there would be no uplink/downlink transmissions, so spectrum can be fully utilized for sidehaul communication.

Conclusion

MCs are expected to be an important component of future wireless networks. They will enhance call quality for cellular users traveling inside a vehicle, offload the network by caching popular contents, and reduce the number of handovers to be managed by the network. MCs could also provide effective coverage in black spot regions as well.

While some researchers are concerned about the risks associated with the increased radiations of 5G, vehicular users will be subjected to low-power transmissions. Having said that, the health impacts of MCs are still an open area of research. Like any new variant of technology, some technological challenges exist to date which must be addressed before MCs can be integrated into 5G.

Read more about it

- S. Jaffry, S. F. Hasan, and X. Gui, "Making a case for moving small cells," in *Proc. Int. Telecommunication Networks Application Conf.*, 2016, pp. 1–6.
- Y. Sui, J. Vihriala, A. Papadogiannus, M. Sternad, W. Yang, and T. Svensson, "Moving cells: A promising solution to boost performance for vehicular users," *IEEE Commun. Mag.*, vol. 51, no. 6, pp. 62–68, June 2013.
- F. Jiang, K Thilakarathna, M. A. Kaafar, F. Rosenbaum, and A. Seneviratne, "A spatio-temporal analysis of mobile internet traffic in public transportation systems: A view of web browsing from the bus," in *Proc. 10th ACM MobiCom Workshop Challenged Networks*, 2015, pp. 37–42.
- X. Lin, R. K. Ganti, P. J. Fleming, and J. G. Andrews, "Towards understanding the fundamentals of mobility in cellular networks," *IEEE Trans.*, vol. 12, no. 4, pp. 1686–1698, Apr. 2013.
- A. Merwaday and İ. Güvenç, "Handover count based velocity estimation and mobility state detection in dense HetNets," *IEEE Trans. Wireless Commun.*, vol. 15, no. 7, pp. 4673–4688, July 2016.
- M. Shin, S. T. Shah, M. Y. Chung, S. F. Hasan, B. C. Seet, and P. H. J. Chong, "Moving small cells in public safety networks," in *Proc. IEEE Int. Conf. Information Networking*, 2017, pp. 564–568.
- S. Prasad, S. K. Peddoju, and D. Ghosh, "Energy efficient mobile vision system for plant leaf disease identification," in *Proc. IEEE Wireless Communications and Networking Conf.*, 2014, pp. 3314–3319.
- L. Zhang, D. Fu, J. Liu, E. C. Ngai, and W. Zhu, "On energy-efficient offloading in mobile cloud for real-time video applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 1, pp. 170–181, Jan. 2017.

About the authors

Syed Shan Jaffry (S.Jaffry@massey.ac.nz) is a Ph.D. degree candidate at Massey University, New Zealand. He earned his B.E. degree from NED University, Pakistan, in 2011 and his M.E. degree from the Chonbuk National University, South Korea, in 2014.

Syed Faraz Hasan (F.Hasan@massey.ac.nz) earned his bachelor's degree in electrical engineering in 2008 and his Ph.D. degree in wireless networks in 2011. He is a senior lecturer at Massey University, New Zealand, where he leads the Telecommunication and Network Engineering Research Group.

Xiang Gui (X.Gui@massey.ac.nz) earned his Ph.D. degree from the University of Hong Kong in 1998. He held teaching and research positions at Shanghai Jiao Tong University, China, and Nanyang Technological University, Singapore, before joining Massey University, New Zealand, in 2003, where he is currently a senior lecturer. He is a Senior Member of the IEEE.

P

Software-defined networks

Deepika Vasudevan and Samrudhi Nayak

A software-defined network (SDN) is an emerging network architecture that allows a centralized software program to control the behavior of an entire network. This is done by separating the components that make up a network (such as the data, control, and infrastructure planes) in a completely different way as opposed to its traditional counterpart. It divides the network's control logic from the underlying routers and switches.

It also allows network administrators to dictate the infrastructure of the network through this centralized software (known as the *SDN controller*) instead of manually working with the physical switches or routers that make up the network. It introduces the ability to program the network, which is a relatively new concept in the area of networking.

In traditional network architecture, network devices, such as routers, have the requisite information to forward the packets according to the data in the forwarding table, as shown in Fig. 1(a). This forwarding table or data plane has information about known paths to the destination node. If there is no path, then it builds the forwarding table using information such as the next hop, destination node, and hop limit to compute it from the source to the destination.

When a packet arrives at a network device, the forwarding table is used to make a decision on where to send the packet. This is done by the control plane, which is the component that carries signaling traffic and routing. As this lies in the device,

Digital Object Identifier 10.1109/MPOT.2015.2448733
Date of publication: 6 September 2018

The basic architecture of an SDN comprises an infrastructure layer and a control layer that communicate with one another through a southbound API.

a network administrator must configure each of these network devices to change the way the packet travels. As a result, the control is distributed to disparate devices rather than one entity.

SDNs, due to their centralized perspective, offer a new way to manage networks. As seen in Fig. 1(b), the network devices have a communication line with the control layer or the SDN controller. This communication line is in the form of an application programming interface (API) called *southbound API*. As a result, the control and data planes are now in two different devices in an SDN.

The SDN controller is the entity that provides flow control among the network devices to enable intelligent networking. This allows the network administrator to shape the network traffic from the centralized control console without having to touch the network devices and change any network device rules when necessary. We use the term *intelligent networking*, as this control layer now has the view of the entire network and can provide better network management. Of course, having a view of a network that comprises a vast number of devices might be cumbersome for administrators. So, the architecture also offers a virtualization layer through which only the required part of the network will be viewed.

SDN architecture

The basic architecture of an SDN is given in Fig. 2, and it comprises an infrastructure layer and a control layer that communicate with one another through a southbound API (shown by its standard counterpart, OpenFlow). Above the control layer resides the business application, which exercises its control over the network using a northbound API.

The infrastructure layer comprises the hardware found in the network in terms of routers and switches. The infrastructure found in an SDN is less expensive than those found in a traditional network, and they do not need to be constantly changed to upgrade the network. This is because they are now programmable and their control lies with the SDN controller. They are connected to the controller via southbound APIs.

Southbound APIs form the interface between the SDN controller

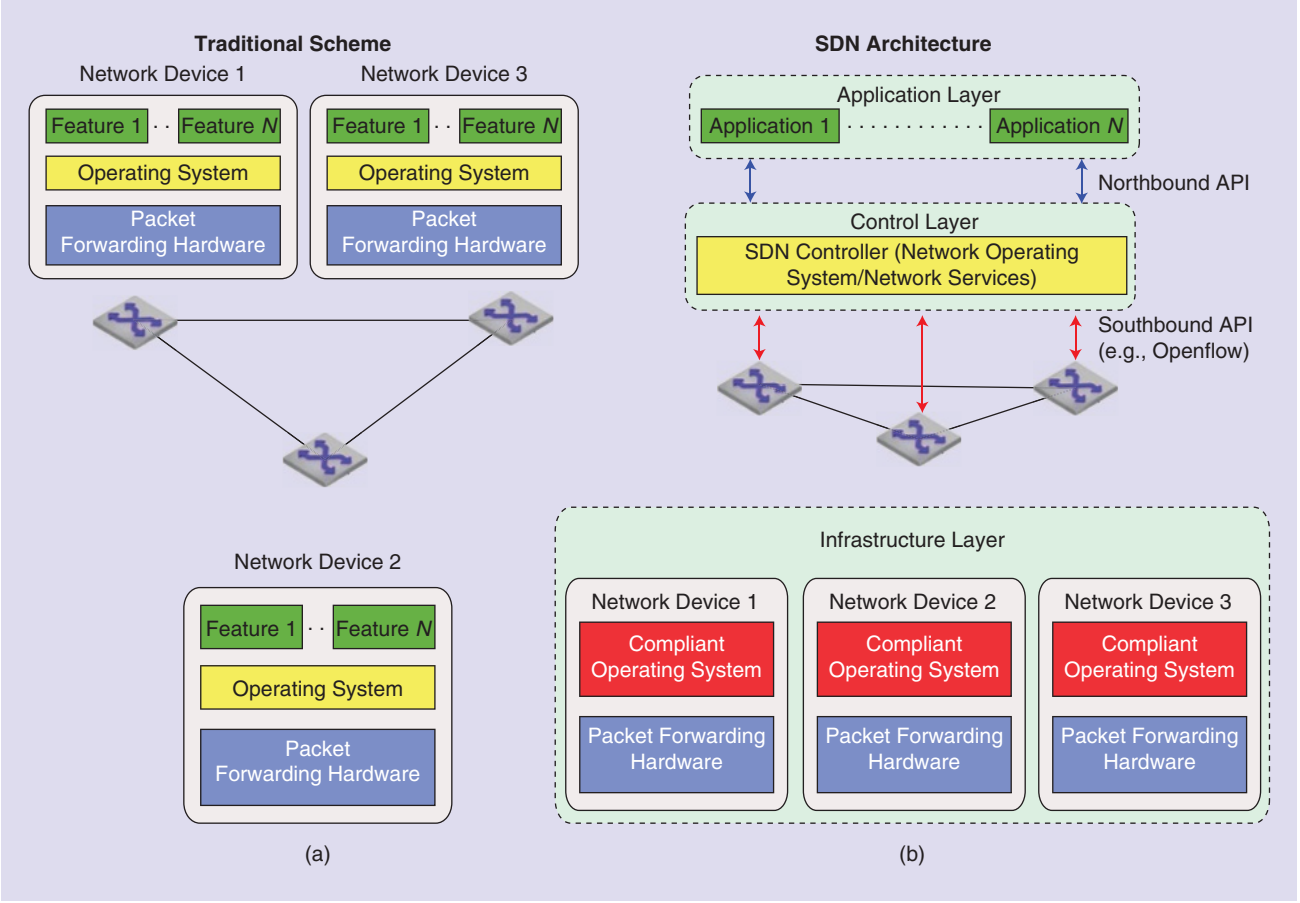


FIG1 (a) Traditional network architecture and (b) SDN architecture. (Figure courtesy of A. Rios.)

and the network switches or routers. These have control over the forwarding operations, event notifications, and statistical reporting and also advertise the capabilities of the network. Essentially, it allows a controller to define the behavior of the hardware in the network. The standardized and most common southbound API is OpenFlow.

The control layer comprises the SDN controller, which is the centralized unit in charge of translating the network requirements from the SDN applications down to the network switches. It also provides the business applications residing in the application layer with an abstract view of the network. This is done through the northbound API.

Northbound APIs present an abstraction of network functions with a programmable interface for applications to consume the network services and configure the network dynamically. They allow the applications to dictate the behavior of the network. The application layer consists of programs that explicitly communicate their network requirements and expected network behaviour to the controller through the northbound APIs.

With reference to the rapid evolution and adaptation of SDNs, the need for standardization emerged. The movement toward this agenda began with the emergence of OpenFlow, promoted by the Open Networking Foundation. It provides a way to dynamically and programmatically control the behavior of switches throughout the network. This is now an important technology that is used to realize SDNs.

OpenFlow

OpenFlow is the first standard communications protocol defined between the control layer and the infrastructure layer of an SDN architecture. It allows direct access to, and manipulation of, the forwarding plane of network devices such as switches and routers.

OpenFlow uses forwarding as opposed to switching or routing. The control plane updates the entries in the forwarding table of the router or switch, which is used by the data plane to dispatch the frames and packets to their respective ports. An OpenFlow controller uses a rule set to control packet forwarding in the data plane of an OpenFlow switch. It centrally manages how packet forwarding in the data plane of the OpenFlow switch will be handled by issuing processing rules for the data plane.

OpenFlow is the first standard communications protocol defined between the control layer and the infrastructure layer of an SDN architecture.

The OpenFlow protocol consists of three types of messages. The first is the controller to switch messages that are sent by the controller to modify, specify, or delete the definitions of the flows; request information on the capabilities of the switches; retrieve information from a switch; and process a packet again by sending it back to the switch when a new flow is created.

The second type of message is the asynchronous message sent by the switch to send the packets that do not match the existing flow to the controller, inform the controller of the packets that have been removed when their time to live parameter has expired, and inform the controller of the change in port status or of any errors that occur.

The third type of message is the symmetric message, which is sent either by the switch or the controller. These are either hello messages exchanged between the switch and controller or echo messages used to determine whether the controller to switch connections is still active. They also include the experimenter messages used for future extensions to OpenFlow technology.

This architecture provides users with high programmability and configuration capabilities; therefore, fine-grained per-flow traffic control can be achieved in the OpenFlow network, where it is difficult to achieve

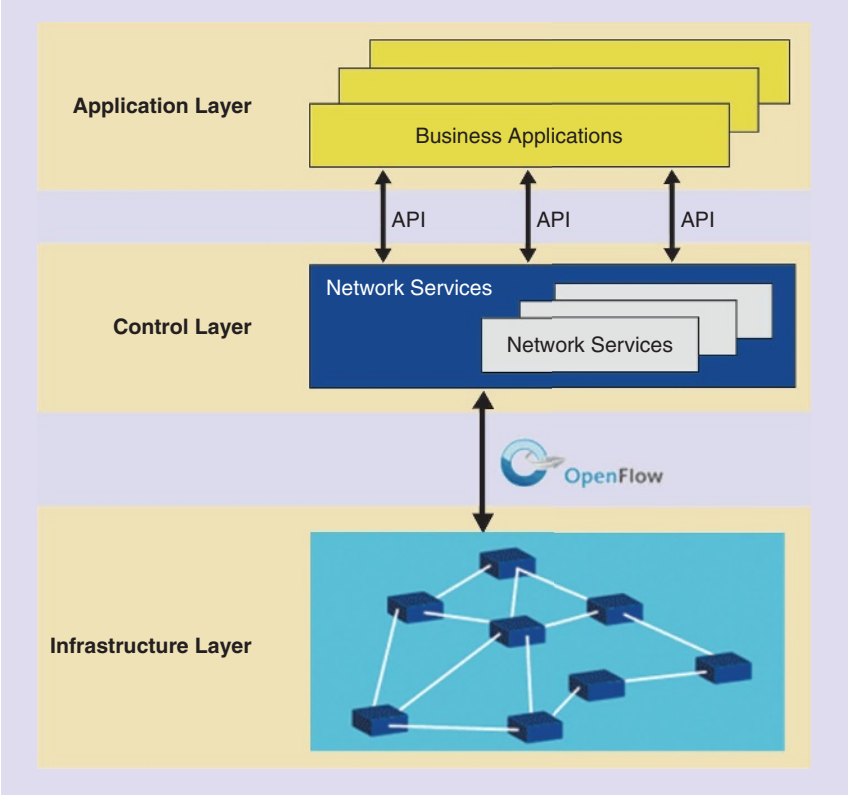


FIG2 SDN architecture (courtesy of The Open Networking Foundation.)

A major concern of SDNs has always been scalability, specifically in terms of decoupling the control and data planes.

with existing network equipment. Since it is the first standard that has been created, the terms *software-defined networking* and *OpenFlow* are used interchangeably. As it is an interface that enables communication between the controller and the infrastructure, it is clearly not the case.

Recent developments

Current research developments in the field of SDNs are divided into the following areas: switch designs, controller platforms, resilient communication, scalability, performance evaluation as well as security and dependability.

The switches that are currently available for SDNs are very diverse. They differ in terms of the feature set, performance, interpretation, as well as architecture. This heterogeneity obviously leads to implementation problems, and many solutions are now being found to solve it.

As the SDN controller is fundamental to an SDN model, many research efforts are being devoted to increase its performance, scalability, and modularity while also making it a highly available, programmer-friendly software. Achieving resilient communication is of high priority when it comes to networking. Although there are examples of SDN networks being resilient at scale (such as Google B4), there is a lack of sufficient research and experience in building and operating fault-tolerant SDNs. As a result, this area is now actively being pursued by researchers.

A major concern of SDNs has always been scalability, specifically in terms of decoupling the control and data planes. There are limitations in terms of quality of service and overheads that are caused, which are currently being addressed.

There are very few performance evaluation studies that have been done concerning OpenFlow and SDN architecture. Simulation studies and experimentation are the widely used techniques used to measure performance in SDNs. However, they take a lot of time and effort to produce consistent outputs. Therefore, current research is focused on analytical models that can help simplify performance evaluation of SDNs.

Due to the danger of cyberthreats and attacks, security and dependability are top priorities in SDNs. While experimentation in the field of security in SDNs is being conducted by a few commercial ventures, it has yet to be commercially adopted. From the dependability perspective, the availability of Internet routers is a major concern. Numerous threats have been identified in SDN architectures and various approaches to mitigate these risks have been found.

The latest development in SDNs is the onset of ProgrammableFlow. This uses the standard OpenFlow such that all the projects share the same physical network but multiple individual virtual networks are established. This ensures that each project works on its own exclusive virtual network, thereby eliminating the need to use different individual hardware components and connections for each project. This allows for the free and flexible operation of each project with the least amount of physical restrictions. It also maintains the same level of security as traditional networks but is much swifter in operation. Current research is being done to determine in-depth information on network resource usage patterns so as to boost effective resource utilization.

Conclusion

SDNs are an emerging technology that is taking over the networking world. It has the potential to revolutionize legacy data centers and large networks by providing a flexible way to control a network. Due to its recent emergence, it is vastly untapped but, over time, its possibilities and benefits can be thoroughly exploited and used.

Read more about it

- A. Rios. Programmable networks: Separating the hype and the reality. [Online]. Available: http://www.barcinno.com/programmable-networks-hype-reality/
- Open Networking Foundation. Software-defined networks. [Online]. Available: https://www.opennetworking.org/sdn-resources/sdn-definition
- *SDN 101: An Introduction to Software Defined Networking*. Citrix, May 2014.
- D. Kreutz, F. M. V. Ramos, P. Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, "Software defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, Oct. 2014, pp. 14–76.
- Y. Nakajima, "Standardization progress in software defined networking/openflow," *NTT Technical Review*, Feb. 2013.
- S. Azodolmolky, *Software Defined Networking with OpenFlow*. Packt Publishing: Birmingham, U.K., 2013.

About the authors

Deepika Vasudevan (deepikav2793@gmail.com) earned her bachelor's degree in computer science and engineering in 2015. She is an IEEE Student Member and a member of the Computer Society of India.

Samrudhi Nayak (samrudhinayak91@gmail.com) earned her bachelor's degree in computer science and engineering in 2014 and is pursuing her M.S. degree in computer science at the University of Illinois at Chicago.

P

Graphical trust models for agent-based systems

Emily Hernandez and Donald Wunsch

Human interactions are founded on trust and reputation. When someone purchases an item, he or she trusts in the transaction without giving it much thought. A buyer rarely questions whether the cashier is reputable, if the transaction is safe, or what previous customers thought about the buying experience. Verifying all of these issues each time would be prohibitively expensive. The burgeoning field of computer-aided transactions needs trust models to achieve sufficient trust levels without burdensome verification costs.

©ISTOCKPHOTO/COMIC SANS

Digital Object Identifier 10.1109/MPOT.2016.2578966
Date of publication: 6 September 2018

The burgeoning field of computer-aided transactions needs trust models to achieve sufficient trust levels without burdensome verification costs.

In mathematical terms, computational trust is an uncertainty estimate that one agent uses to judge how well another agent will perform a task. As systems increase in complexity, an agent's trust and reputation are important factors in decision making. When agents cannot directly oversee interactions and exchanges with others, metrics for rewarding honesty and other "trustworthy" behavior provide incentives for good behavior and penalize agents who cannot complete tasks.

Graph theoretic models of trust for intelligent systems are powerful tools for this problem. Using trust models in complex networks allows agents to use mathematics to help determine an exchange's risk, improve decision-making skills, and obtain extra information to evaluate an agent. By evaluating data about experiences and transaction results, trust models allow for more reliable decisions. Incorporating trust eliminates unnecessary agent communication and provides additional security.

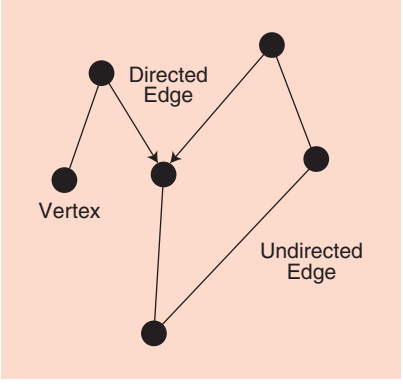


FIG1 A sample graph-theoretical model.

Trust models help simplify and model several real-world situations. With applications from social networking to e-commerce, trust models can have agents that represent individual people or entire firms and businesses. Studying how these kinds of agents interact with one another has obvious benefits. Though much research has been done to optimize and generalize trust models in all areas, the complexity of the problem presents challenges to effectively modelling trust.

Classifying trust: What parameters are considered?

The two most prevalent types of trust models are cognitive and graph-theoretical. Cognitive models are based on belief in a biological sense, while graph-theoretic models use probability and statistics to quantify trust. In computer science and engineering, graph-theoretical models are more applicable, as they allow trust and reputation to appear as subjective probabilities based on utility and past interactions. Individuals have expectations, expressed as probabilities, that others will perform a certain way, and these can be measured against outcomes. Figure 1 shows a sample graph-theoretical model. All graphs contain two elements: vertices and edges. A vertex, drawn as a dot, represents a single node. An edge is drawn as a line that connects two vertices. An edge can connect two different vertices or connect one node to itself (called a *loop*). Two vertices connected by an edge are considered adjacent. A directed edge is one in which order matters. Directed edges are drawn with an arrow to indicate in which direction the vertices are connected. Graphs containing only directed edges are sometimes called *directed graphs* or *digraphs*.

When generating trust models, information gathering is critical. Different models depend on varying information sources, each with respective benefits and shortcomings (Fig. 2). Graph-theoretical models rely on direct experiences and indirect information, because these sources are most reliable, observable, and interpretable. Higher-level

FIG2 A summary of information sources. The two most standard and readily available information sources are direct experiences and indirect information. More complex models have methods to include situational or sociological data.

models are able to incorporate less concrete information such as social status and prejudice.

The most reliable information source in a trust model is direct experience. Direct experiences are an agent's personal interactions with a target agent or personally observed interactions of other agent groups. These interactions form the most basic trust model comprised only of one user's evidence.

The most abundant form of information is indirect information, which is obtained by one agent from the rest of the network. The agent gathering information relies on other agents, called *witnesses*, to report their direct experiences and observations of a target agent (Fig. 3). When factoring this type of information into a model, each witness's honesty can skew the resulting data. Agents may manipulate the information they report to make themselves look better, such as by omitting positive transactions so their own transactions appear superior. In large networks, indirect information is sometimes the only way an agent can gather data.

Beyond these two typical sources, trust models can incorporate a range of sociological factors. Elements such as an agent's role in society (higher power positions signify more trustworthiness), competition with other groups, and collaboration bring new insight to models that can improve reliability. These higher-level models use social relationships to estimate trustworthiness. Because sociological factors are highly situational, trust models that incorporate these concepts are typically limited to specific networks with in-depth agent interactions. Prejudice is another outside factor that impacts how trust levels are assigned. Models that incorporate this feature assign agents properties based not only on data about themselves but also on their group's statistics.

Once information has been gathered, trust values must be defined for each system agent. To do this, trust models use either a global or subjective approach (Fig. 4). Global

FIG3 Indirect information. Outside agents report their experiences and observations to agent 1 so that it can make a decision about agent 2, the target.

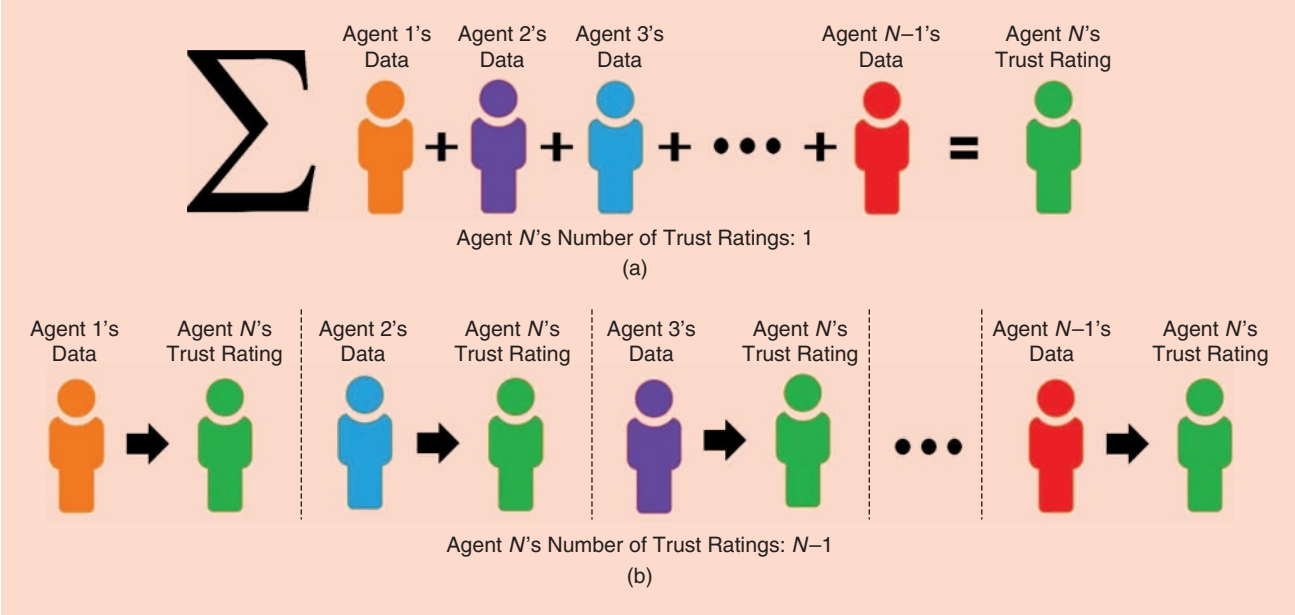


FIG4 A comparison of global and subjective trust models. a) In a global model, all evidence from every agent in the network is summed together to generate agent *N*'s trust rating. In addition, all agents have access to this global rating and can update it with their new inputs. b) In a subjective model, each agent's data are analyzed to generate a unique trust score. Each unique score is determined by one agent's direct and indirect interactions with the target agent. These scores are private and update independently as each agent obtains new data.

In computer science and engineering, graph-theoretical models are more applicable, as they allow trust and reputation to appear as subjective probabilities based on utility and past interactions.

trust models assign each agent one trust value based on the evaluation of all opinions of all agents who have interacted with the agent being evaluated. This global score is made publically available to all agents in the community and is updated each time an agent gains new information about the target agent.

Subjective models, on the other hand, are more personalized. Each agent assigns a unique trust value to another based only on the information it has gathered. Although the evaluator agent may use indirect information and direct experiences to rate other agents, it does so privately. In this type of model, every agent in a network of size N would have $N-1$ trust scores, one from each agent's point of view. Each of these $N-1$ scores updates independently as agents gain new information. With subjective models, speaking about trust must be done with the perspective of agent X instead of globally.

In large networks, every agent cannot participate in multiple interactions with every other agent. For these systems, subjective trust models incur more risk than global models. By using a global model, each agent in the large network has access to a wider set of opinions and experiences, condensed into the agent's trust value, to make his or her decisions. Although the risk of deception is present, scarcity of agent interactions outweighs the downside of accessing the information.

For smaller systems, subjective trust has its benefits. When networks contain small groups of agents that interact frequently with each other, each agent establishes stronger connections and repeated experiences on which to base his or her unique trust assessment. In such a network, having several "point-of-view" trust values still benefits the system. When deciding between a global or subjective trust model, the network's available interactions must be assessed. If interactions between agents are scarce, a global trust model is more beneficial for the system. To determine if a subjective trust model may be accurately used, a certain minimum threshold of interactions between agents has to be established, below which a subjective model cannot correctly assess the network.

Context dependence and reliability are two final classifications of trust models. In a single-context system, an agent has one trust value that holds for all interactions, regardless of their context or category. The agent's trustworthiness is assumed to be consistent in all areas. For example, an agent that completes transactions in medical areas and fine arts is assumed to have the same trust level across all of its transactions. This approach works well for models with limited situation types but not necessarily for broader networks. A multicontext model incorporates different trust values for every included context. In the previously stated case, the agent would have one trust value for its medical transactions and a separate one for its fine arts interactions. While allowing multiple contexts greatly improves a trust model's accuracy, problems arise when information is too scarce to divide the set of experiences into different contexts.

Finally, in some trust models, a trust value's reliability can be accounted for when making decisions. Reliability is a single value linked to the trust value and can be based on an agent's number of interactions, witness reliability scores, or the recency of information used to evaluate trust. Higher reliability scores translate to the trust value's weight in the decision-making process. If an agent's information has a reliability of 0.99, trustworthiness will be heavily considered, while data with a reliability of 0.1 is assumed to be insignificant.

Examples of trust models

A statistical approach to situational trust, cooperation thresholds, and reciprocity

One of the earliest trust models was proposed by Stephen Marsh in 1994. He considered only direct experiences of agents for this model. Marsh defined three types of trust: basic, general, and situational. Basic trust models an agent's disposition to trust, based on the sum of all its prior experiences. Basic trust is not tied to specific agents and instead reveals how likely an agent is to trust others based on how it has previously been treated. General trust is the trust an agent has for another without considering specific situations. Situational trust is the product of general trust, situation importance, and utility gained. Importance varies based on how likely a particular outcome is, while utility is a static measure of how useful an event's outcome could be.

Marsh used three different statistical methods to select data for trust estimates: maximum, minimum, and mean (Fig. 5). An optimistic agent selects the maximum trust value from its previous interactions, a pessimistic agent uses the minimum value, and a realistic agent takes the average of its trust over all prior situations. Once this estimate has been calculated, the agent must decide whether it should cooperate or not.

In addition to situational trust, Marsh implemented a cooperation threshold that depends on the transaction's importance, situational risk, and perceived agent competence. If the situational trust value is above the cooperation threshold, the transaction will take place. The final feature of this model is reciprocity, or the idea that agents will "pay back" favors given to them in the past. For example, if an agent cooperates in a transaction but the target agent defects, the first agent reduces its trust in the target while the target gains trust for the first agent.

A probabilistic trust model for the Prisoner's Dilemma using Boolean choices and player honesty

Schillo, Funk, and Rovatsos's 2000 model incorporated direct experiences and indirect information separately. This model was designed for interactions with Boolean trust outcomes, or outcomes rated as positive or negative without in-betweens. To test this model, the researchers used a Prisoner's Dilemma set of games.

The Prisoner's Dilemma problem is widely used in computer science to understand agent-based systems. To better understand the Prisoner's Dilemma, consider the following example. Two prisoners are arrested and placed in separate cells. A prosecutor offers both a choice: confess or say nothing. If one prisoner confesses and the accomplice stays quiet, only the accomplice goes to jail. If both prisoners confess, both are convicted and guaranteed early parole. If both remain silent, they will receive sentences for a less serious crime. The "dilemma" in this situation is that, regardless of what the other prisoner does, each is better off confessing. However, if both confess, the outcome is worse than if they both had stayed quiet. In this way, the Prisoner's Dilemma models situations in which rational, selfish agents must decide whether to cooperate for the common good or prioritize selfish behavior.

When applied to trust models, cooperation signifies a successful transaction while defection (selfishness) is a failed interaction. Each agent wants to maximize its total payoff score, but as Fig. 6(b) indicates, the reward matrix presents mutual cooperation as the best combined payoff while tempting each player to defect with the highest single-player score. The decision to risk cooperation knowing that the other player could defect, or defect and risk getting a lower score than mutual cooperation, is the dilemma of the game.

To augment the traditional Prisoner's Dilemma game, the researchers included a partner-selection phase, in which each agent claims what it plans to do in the actual game [Fig. 6(a)].

The Prisoner's Dilemma models situations in which rational, selfish agents must decide whether to cooperate for the common good or prioritize selfish behavior.

Both player agents can access the results of all games they have played, the results of neighbor agents, and the claim their opponent makes. The outcome of one game reflects both agent honesty (did the agent perform as it claimed?) and Prisoner's Dilemma behavior (did the agent cooperate or defect?). For each two-stage game, four honesty scenarios are possible. To calculate the trust that agent A gives to agent B, divide the number of times agent B was honest by the total number of observed games with agent B playing.

In addition to direct experiences, each agent can communicate with any agent it has met previously to access additional indirect information. The resulting data structure in Fig. 7 is a TrustNet, a directed graph in which each node represents an agent and edges represent shared information between one witness node and the root node (the owner) about a child node (the target agent).

In a TrustNet, the owner sorts through information from witness nodes to determine how much biased data are present. To simplify this process, this model assumes witnesses will not lie about negative interactions, since hiding this information only makes them look worse if caught. Witnesses may, however, hide positive information about other agents to make themselves appear more trustworthy. This simplification reduces the problem to determining which witnesses have biased information and how much was biased. To solve this, hiding information is modeled as a stochastic process. An agent decides to share positive information with probability p and hides it with probability $(1-p)$. This represents a Bernoulli chain when applied repeatedly and can be recursively analyzed to build an approximation of what the witnesses would have said if they were completely honest. Regarding the Prisoner's Dilemma stage of the game, an agent gleans information about agent reliability based on how often he or she cooperates and defects and updates the TrustNet accordingly. The resulting trust

FIG5 A summary of agent types and how each obtains general trust.

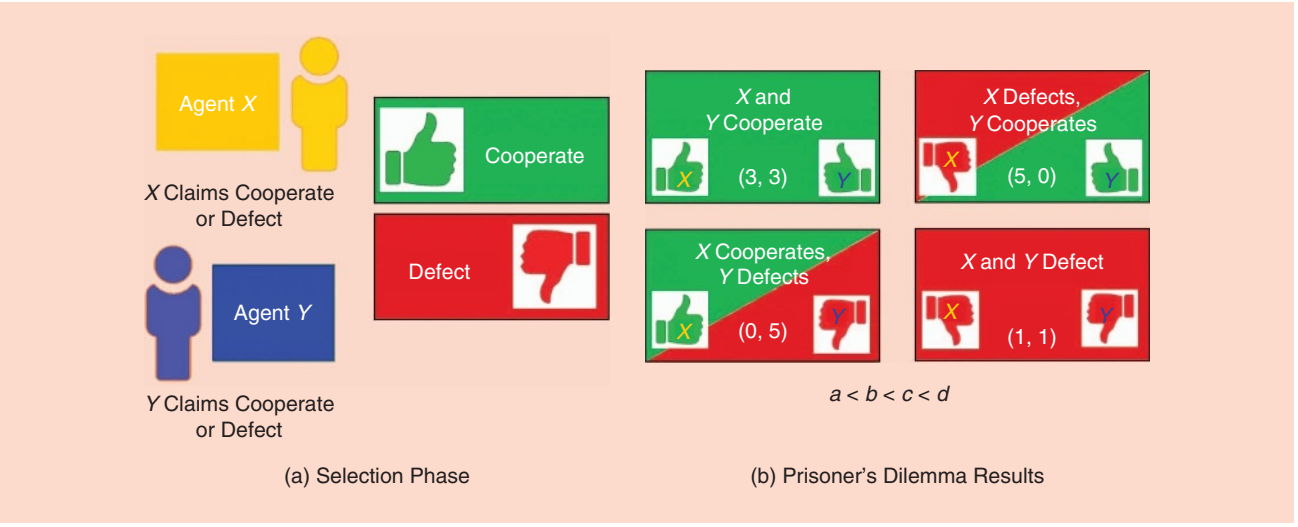


FIG6 A Prisoner's Dilemma game model. (a) Agents *X* and *Y* claim that they will cooperate or defect. (b) Four Prisoner's Dilemma outcomes are possible, shown as (*x, y*), where *x* and *y* are the respective agents' scores. The scores must follow the relationship $a < b < c < d$ to tempt agents to defect and earn a higher score while providing the best mutual outcome with both agents' cooperation.

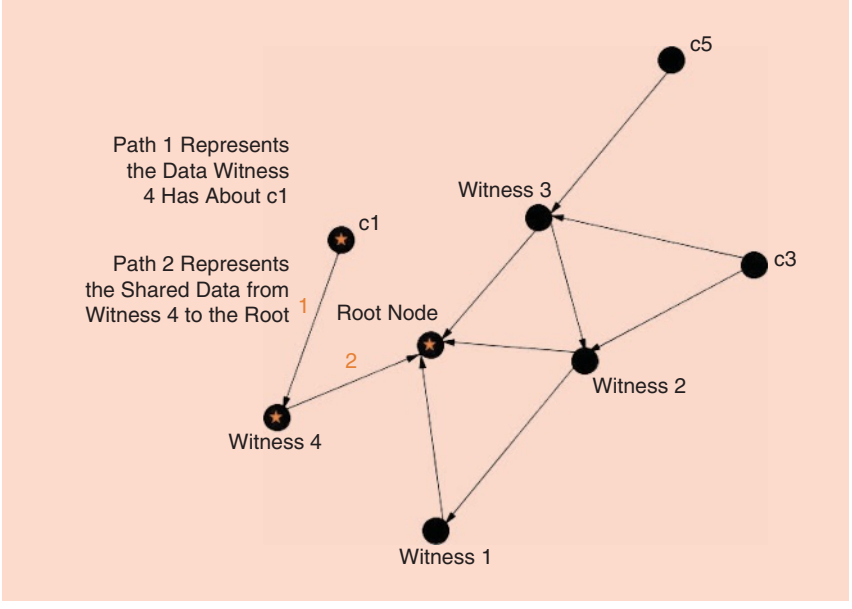


FIG7 A TrustNet example: The root node is the destination of all directed edges, and each edge represents transmitted information. In the labelled path, witness 4 shares some subset of information with the root node.

value is subjective, since each agent has a unique TrustNet.

Neural networks and Markov decision processes for the iterated Prisoner's Dilemma

Seiffertt et. al.'s 2009 paper incorporated Markov models for the iterated Prisoner's Dilemma game. Interest in this type of Prisoner's Dilemma game began with Robert Axelrod's tournament. Axelrod invited colleagues from around the world to develop computer strategies to participate in an iterated Prisoner's Dilemma tournament. In this game, players are given points based on their chosen action (cooperate or defect). Players participate in multiple rounds against their opponent, adding points to their overall score based on the outcome of each round. A description of the four possible outcomes and their point values is listed in Fig. 8.

As in the previous Prisoner's Dilemma problem, mutual cooperation yields the highest score for both players, but if one player defects, the payoff is much greater for him or her. The ultimate goal after all rounds is to have the highest average score. Players in this model can only access direct experiences and do not communicate with each other to share results globally. Since this Prisoner's Dilemma game has multiple rounds, players establish trust based on previous game results. Ideally, two agents who quickly learn to trust one another will select mutual cooperation for all future rounds, resulting in the best-case score for the pair.

Interestingly, Axelrod's tournament yielded several preferred conditions for a successful strategy: niceness, retaliation, forgiveness, and nonenviousness. To maximize points in a single game, defection is the most obvious choice. However, to succeed after multiple iterations of the game, a more dynamic strategy must be implemented. The strategy must be nice in that one will not defect before his or her opponent and will retaliate if the opponent does defect. The strategy must also be forgiving if the opponent defects once and then cooperates for several rounds. Finally, it cannot be envious, striving to score more than its opponent in every round. Tit-for-tat strategies embody these four traits and, consequently, are most robust in iterated Prisoner's Dilemma games.

Seiffertt et. al. first tested two categories of strategies: pure and behavioral. Pure strategies are Boolean, so players using this method cooperate or defect without degrees of probability. The simplest pure strategies are to always cooperate or always defect regardless of the opponent's action. Players who use more adaptable pure strategies factor in their opponent's decisions, altering their choices based on opponent behavior. Two examples of adaptable pure strategies are copying the opponent's last move or making the opposite choice of the opponent. Behavioral strategies allow players to combine pure strategies to gain the upper hand. Instead of a Boolean response, behavioral strategies have probabilities associated with cooperation and defection. For example, a particularly optimistic player could have a 0.9 chance of cooperating and a 0.1 chance of defecting, assuming his or her opponent chose to cooperate in the last round.

The primary contribution of this paper was to observe that if the iterated Prisoner's Dilemma problem is analyzed using a Markov decision process, the solution yields a "best course of action" for each player during every decision period. Using this framework, both pure and behavioral strategies can be ranked based on optimality. Pure strategies with fixed values for cooperation and defection are the simplest to evaluate. Assuming an arbitrary starting move, eight possible strategies exist. When expressed as an ordered pair $[i, c, d]$, i is the initial move, c is the player's response if his or her opponent chose cooperation last, and d is the player's response if his or her opponent defected last. Table 1 describes all eight strategies. Against an opponent who acts randomly, a player who always defects will end up with the highest score.

People do not behave like fixed players, ignoring their opponent's actions and responding objectively. To simulate intelligent players, behavioral strategies must be incorporated. To explore a large number of strategies and foster a dynamic environment, a 100-square grid was populated with different strategies, none of them absolute. After each round, every "cell" (individual strategy) compared its score with its neighbors' scores and adopted the highest-scoring strategy. After 50 rounds, the strategy with the highest average score was declared the winner.

The basic idea of TRAVOS involves binary interactions, that is, situations in which an outcome is successful or not.

Interestingly, strategies that favor defection were only dominant in early-game rounds. Toward mid- and end-game, a more unstable tit-for-tat strategy, in which players repeatedly alternate between cooperation and defection, appeared most powerful. The more cooperative tit-for-tat strategies became very strong in the later game, defeating defection strategies entirely. At this late-game stage, any cooperation-based strategies that survived the early game could be fairly powerful. Determining which strategy is more optimal depends on the player's goals. To dominate early on, a defection strategy has the best likelihood. To succeed in the mid-game phase, a tit-for-tat strategy is ideal. To make it to the end, more cooperative tit-for-tat strategies nearly always result in victory but risk early elimination by defection strategies.

Quality of service thresholds and Dempster-Shafer theory

Yu and Singh's 2002 trust model determined trust using direct or indirect information. In this model, direct interactions were stored in a set of values called the *quality of service* (QoS). The QoS values reflect the quality of an agent's most recent

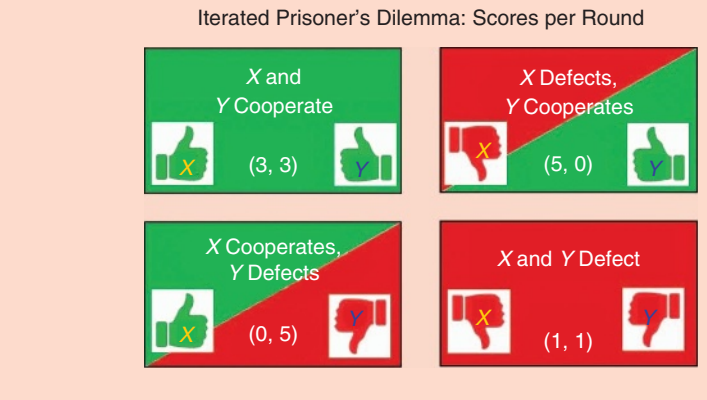


FIG8 A summary of possible player scores for each Prisoner's Dilemma outcome. Mutual cooperation is the most desirable outcome, but the temptation to defect is high enough to incentivize a range of actions. Relating these results to the $a < b < c < d$ relationship, all Prisoner's Dilemma setups share, $a = 0$, $b = 1$, $c = 3$, $d = 5$.

TABLE 1. Pure strategies for the iterated Prisoner's Dilemma game.

[0,0,0]	Always defect, initially defect
[1,0,0]	Always defect, initially cooperate
[0,0,1]	Defect unless the other player defects, initially defect
[1,0,1]	Defect unless the other player defects, initially cooperate
[0,1,0]	Cooperate unless the other player defects, initially defect
[1,1,0]	Cooperate unless the other player defects, initially cooperate
[0,1,1]	Always cooperate, initially defect
[1,1,1]	Always cooperate, initially cooperate

interactions. Every agent determined an upper and lower QoS threshold for three types of agents: trustworthy, nontrustworthy, and unclassified. From here, the Dempster-Shafer theory of evidence, in which a degree of belief is generated from combining different sources of evidence, allowed agents to calculate the probability that their target agent's services belong to one of the three categories. If the service's probabilities of being in the trustworthy and untrustworthy group differed by more than a set trust threshold, the agent was considered trustworthy.

When witnesses provided information in this model, they returned either information about the target agent (if the target is an acquaintance) or a referral to the target agent. Each referral could provide information or a new referral to get to the desired information in a chain that continued until the requested information was obtained. Once all indirect information had been obtained, Dempster's rule of combination was used to complete analysis.

This model provided no mechanism to combine direct information and indirect information. If present, direct information was the only source considered when computing trust.

Linking direct and indirect information with confidence metrics in Bayesian systems

The Trust and Reputation system for Agent-based Virtual Organizations (TRAVOS), was developed in 2006 by Teacy, Patel, Jennings, and Luck. This model is based on probability theory and Bayesian systems and used both direct interactions and indirect information to assess trustworthiness. Unlike previously mentioned models, TRAVOS can link the two information-gathering methods using a confidence metric.

TABLE 2. Social networking application description: Assuming the target agent is called "User A," each item in the left column represents a list of data pairs that the researchers collected. Each list was multiplied by each weight set and the number of interactions to yield four unique top-ten friends lists.

INFORMATION GATHERING	CALCULATION ALGORITHM WEIGHT SETS			
ITEM DESCRIPTION	SET 1	SET 2	SET 3	SET 4
List of friends who are tagged in the same Facebook photo with User A	6	random()	2	5
List of friends who write on User A's Facebook wall	4	random()	4	2
List of friends who leave comments on User A's Facebook wall	3	random()	3	2
List of friends who like posts on User A's Facebook wall	2	random()	2	1
List of friends who write to User A's Facebook inbox	1	random()	5	1
List of friends on whose Facebook walls User A writes or comments	5	random()	5	1
List of friends who like User A's Facebook photos	1	random()	1	1
List of friends who leave comments on User A's Facebook photos	1	random()	1	1
List of all Facebook friends of User A	–	–	–	–

The basic idea of TRAVOS involves binary interactions, that is, situations in which an outcome is successful or not. The set of all observed interactions of an agent are accumulated over time to generate an agent's behavior tendency. If agent Q tends to fulfill half of its obligations to agent P, then agent Q's behavior tendency is 0.5. Because these measures are in regards to two specific agents, all parameters in TRAVOS were subjective. TRAVOS defined trust as a probabilistic, experience-based measure of how likely an agent is to perform in a certain way. To compute trust, they used the expected value, or average, of the behavior given the experience set of the agent.

Each agent has an additional metric called *confidence*. Confidence is the accuracy of a trust computation based on the number of observations the agent uses when computing. In this way, more evidence increases an agent's confidence in its assessment. For an agent with a fixed observation set, confidence is the probability that the actual behavior of an agent lies within an acceptable error threshold of the calculated trust value. If the set of evidence is unchanged, increasing the error threshold can make an agent more confident in its decisions.

By including confidence, TRAVOS could implement a decision-making process that leads agents to seek more information when a minimum confidence threshold is not met. As a baseline, all agents calculated trust based on direct experiences. However, when an agent realized its confidence level was too low (i.e., that it did not have enough data to reach a reliable decision), it sought out the opinions of other agents to boost its confidence. These secondary opinions are an indirect information source and form the target agent's reputation. In this model, agents had a single value for the truthfulness of received opinions that did not vary from agent to agent. Assuming all witness opinions are independent, the reputation score was determined by adding up the number of successful and unsuccessful interactions of all reports. TRAVOS filtered inaccurate data by judging the

perceived accuracy of each witness agent's past opinions.

Application to social networking

Pododnik et. al. developed a trust-based model that calculates a Facebook user's closest friends using weighted nodes. These calculations are useful for social recommender systems that suggest products based on what a user's closest friends have previously liked. In social networking, the closeness of a friend is directly related to that friend's trustworthiness. Since Facebook is a direct social network, its users build friend networks using explicitly defined connections. As a result, researchers can calculate the trust between two Facebook friends using their recorded social activities.

To test this assumption, Pododnik et. al. created a Facebook application that used social activities to calculate trust. The list of social activities considered is in Table 2. The social activities set is multiplied by the number of recorded activities and four unique weight sets to generate four possible "closest friends" lists. Users must select which list best represents their top friends. Results showed that Set 3 and Set 1 best matched the users' top friends, revealing that writing and commenting on Facebook walls is important.

Conclusion

As networks grow, so does the necessity of reliable trust models. The five aforementioned models are a small sample of the algorithms developed to quantify trust in computer networks. As research continues, graphical trust models have expanded to cover not only direct experiences and indirect data but also a range of sociological information. This information can be used in social network models, online reputation models, and more.

As research continues, graphical trust models have expanded to cover not only direct experiences and indirect data but also a range of sociological information.

Read more about it

- G. Lu, J. Lu, S. Yao, and J. Yip, "A review on computational trust models for multi-agent systems," *Open Inform. Sci. J.*, vol. 2, pp. 18–25, 2009.
- P. Sant and C. Maple, "A graph theoretic framework for trust—From local to global," Tenth International Conference on Information Visualization. London, 2006, pp. 497–503.
- J. Sabater and C. Sierra, "Review on computational trust and reputation models," *Artif. Intell. Rev.*, vol. 24, no. 1, pp. 33–60, 2005.
- S. Marsh, "Formalising trust as a computational concept," Ph.D. dissertation, Department of Computational Science and Mathematics, Univ. of Stirling, 1994.
- J. Seiffertt, S. Mulder, R. Dua, and D. Wunsch, "Neural networks and Markov models for the iterated prisoner's dilemma," in *Proc. Int. Joint Conf. Neural Networks*, Atlanta, GA, 2009, pp. 2860–2866.
- W. T. L. Teacy, P. Jigar, R. Nicholas, and M. L. Jennings, "Coping with inaccurate reputation sources: Experimental analysis of a probabilistic trust model," in *Proc. Int. Conf. Autonomous Agents Multiagent Systems*, 2005, pp. 997–1004.
- M. Schillo, P. Funk, and M. Rovatsos, "Using trust for detecting deceitful agents in artificial societies," *Appl. Artif. Intell.*, vol. 14, no. 8, pp. 825–848, 2000.
- B. Yu and M. Singh, "An evidential model of distributed reputation management," in *Proc. Int. Conf. Autonomous Agents Multiagent Systems*, Bologna, Italy, 2002, pp. 294–301.
- S. Kuhn. (2014). Prisoner's Dilemma. *The Stanford Encyclopedia of Philosophy*, E. N. Zalta, Ed. [Online]. Available: http://plato.stanford.edu/archives/fall2014/entries/prisoner-dilemma/
- F. Harary, *Graph Theory*. Reading, MA: Addison-Wesley, 1969.
- V. Podobnik, D. Striga, A. Jandras, and I. Lovrek, "How to calculate trust between social network users?" in *Proc. Int. Conf. Software Telecommunications Computer Networks*, 2012, pp. 1–6.

About the authors

Emily Hernandez (e.g.hernandez@ieee.org) is an electrical engineering undergraduate student at Missouri University of Science and Technology (Missouri S&T). She is a National Merit finalist and involved with the Society of Women Engineers, Society of Hispanic Professional Engineers, and IEEE-HKN chapters at Missouri S&T. She was the 2015–2016 president of the Robotics Competition Team and has interned at Molex, Intel, and Garmin.

Donald Wunsch (wunsch@ieee.org) earned his B.S. and M.S. degrees in applied mathematics from the University of New Mexico, his M.B.A. degree from Washington University in St. Louis, and his Ph.D. degree in electrical engineering from the University of Washington. He is the Mary K. Finley Missouri Distinguished Professor and director of the Applied Computational Intelligence Laboratory at Missouri University of Science and Technology. He is an IEEE Fellow.

P

Dynamic time-warping dissimilarity matrices

Ana Lorena Uribe-Hurtado, Mauricio Orozco-Alzate, and Efraín Alberto Rodríguez-Soto

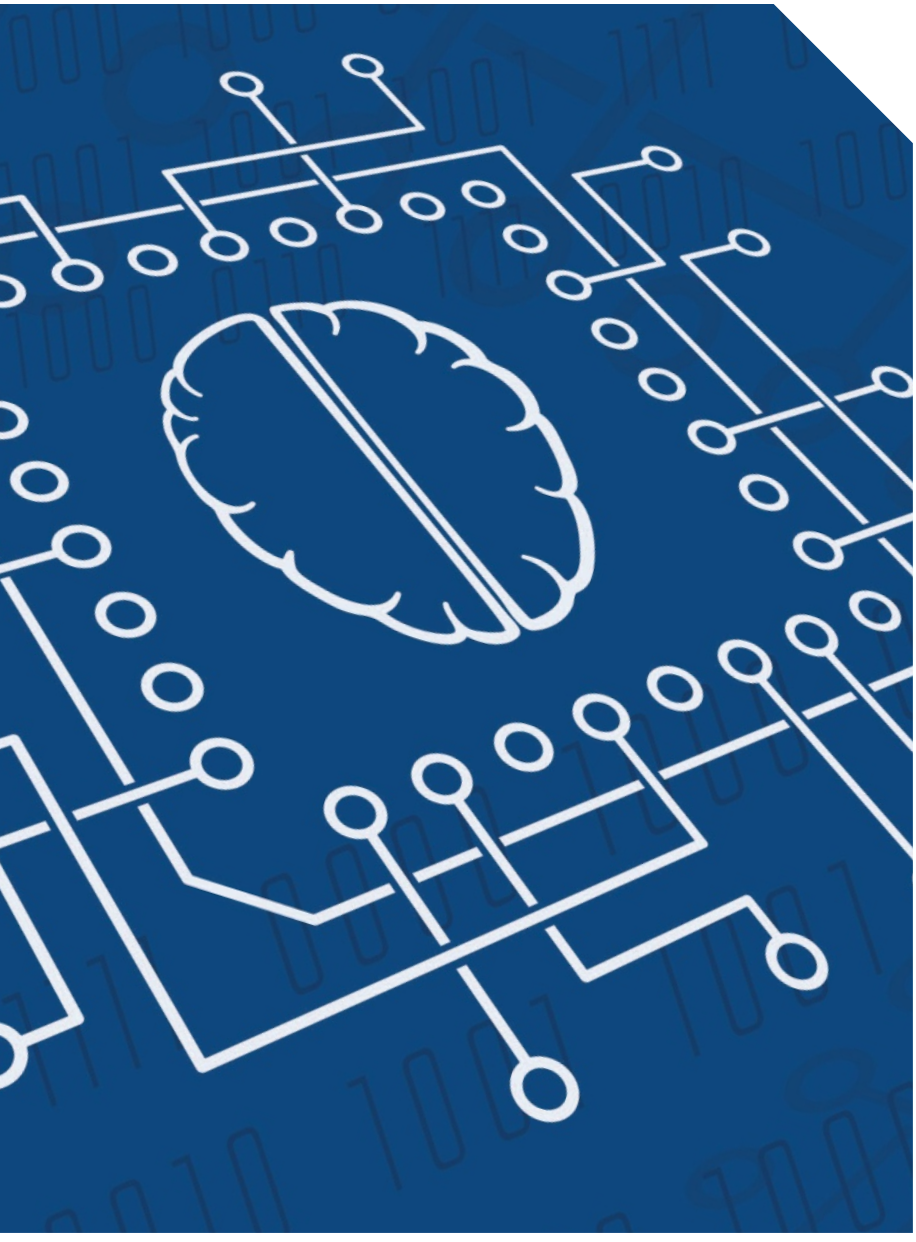


IMAGE COURTESY OF ING IMAGES

Digital Object Identifier 10.1109/MPOT.2014.2358696
Date of publication: 6 September 2018

The dynamic time-warping measure (DTW), also known as the DTW distance, is a widely used method that quantifies how dissimilar two time series are from each other. That quantification of the degree of dissimilarity is often motivated by tasks such as data exploration/visualization, clustering, and classification. Practical examples based on the DTW measure include:

- plotting a dissimilarity matrix, resulting from pairwise comparisons between two sets of hydrologic time series, as an intensity image to visualize emerging cyclical and nontrivial patterns
- splitting one or more years into groups of days, according to the behavior observed from electricity price time series
- assigning class labels to biomedical signals by using a template matching approach with a set of prototype pathological examples.

Even though several alternative dissimilarity measures have been proposed—such as longest common subsequence (LCSS), edit distance with real penalty (ERP), edit distance on real sequence (EDR), and time-warp edit distance (TWED)—the DTW distance remains one of the most widely applied dissimilarity measures for comparing time series; its popularity is due to both its effectiveness and simplicity of implementation.

A number of software implementations of the DTW dissimilarity measure are freely available on the

Internet. Some are written in MATLAB: one of the most popular and powerful packages for numerical computation. MATLAB offers several attractive features to scientists and engineers, such as sophisticated graphics capabilities, many proprietary as well as free toolboxes, extremely simple language syntax and refined interface, cross-platform compatibility, and an always-growing set of built-in and contributed functions.

MATLAB programs are interpreted and, therefore, slower than equivalent versions written in compiled languages such as C, C++, or Fortran. This disadvantage might not be very important when running low-cost algorithms, but it becomes very relevant for large-scale numerical computations. It is precisely the case of computing the DTW dissimilarity measure between pairs of long time series and, moreover, between large training and test sets for clustering or classification experiments. The computational cost of the DTW measure, for comparing time series of length m and n, is $O(mn)$; that is, it has a quadratic time complexity for equal-length time series: $m = n$. A typical application implies the computation of all pairwise DTW-based comparisons from a set of test time series to another set of training (reference) ones or even the computation of all pairwise comparisons between two time series from the union of the two sets.

MATLAB offers several attractive features to scientists and engineers, such as sophisticated graphics capabilities, many proprietary as well as free toolboxes, extremely simple language syntax and refined interface, cross-platform compatibility, and an always-growing set of built-in and contributed functions.

Fortunately, several solutions to both problems—the high computational cost of the DTW measure and the slowness of MATLAB—have been proposed. The former has been faced by imposing global constraints (for example the Sakoe-Chiba and the Itakura bands) to the warping path and by using the so-called multiscale DTW approach. The latter has been tackled, among other strategies, by: 1) preallocation of the arrays and vectorization of the loops, 2) compilation and linking of C, C++, or Fortran codes into a binary (.MEX) file callable from MATLAB, and 3) taking advantage, with the Parallel Computing Toolbox, of the multiple cores or processors available in most modern computers.

The purpose of this article is to explain how to speed up publicly available MATLAB implementations of the DTW dissimilarity measure by using the last strategy mentioned in the previous paragraph and, particularly, when computing the DTW dissimilarities for many time series contained in two separated sets for training and test, respectively. An explanation on how to effectively use the multiple cores in a computer is provided, together with experimental evidence of the improvements in the execution speed as the number of cores is increased. For the sake of reproducibility, a freely available data set from the UCR Time Series Classification/Clustering Page (http://www.cs.ucr.edu/~eamonn/time_series_data/) is used: the synthetic control data set.

The DTW dissimilarity measure and its implementations

We previously mentioned that one of the reasons for the popularity of the DTW dissimilarity measure is its effectiveness to compare pairs of time series which, depending on the problem, is reflected in either low classification errors or good cluster validity indexes. In addition to this fact, the DTW measure is also preferred

LISTING 1. The algorithm to compute the DTW dissimilarity measure by using dynamic programming and a band of width w. This pseudocode is based on the MATLAB implementation by Quan Wang.

```
1:  procedure DTW (A, B, W)
2:    w ← max(w, |m − n|)
3:    Initialize a matrix D, of size (m+1) x (n+1), with all Inf entries
4:    D[1, 1] ← 0
5:    for i ← 1 to m do
6:      for j ← max(i − w, 1) to min(i+w, n) do
7:        D[i+1, j+1] ← (A[i] − B[j])^2 + min(D[i, j+1], D[i+1,j], D[i,j])
8:      end for
9:    end for
10:    return D[m + 1, n + 1]
11: end procedure
```

The DTW measure is also preferred because it does not require that both time series are of the same length as needed by typical distances; such a feature is called *elasticity*.

because it does not require that both time series are of the same length as needed by typical distances; such a feature is called *elasticity*. The DTW measure is, thereby, an elastic dissimilarity measure that quantifies the best alignment between two time series by minimizing a distance between them. A typical implementation of the DTW dissimilarity measure uses 1) dynamic programming and 2) a window size to constrain the search of the alignment to a band (e.g., the Sakoe-Chiba band and the Itakura band); see Listing 1. Both strategies are used in a publicly available MATLAB implementation of the DTW measure, which is available at http://www.mathworks.com/matlabcentral/fileexchange/43156-dynamic-time-warping-dtw.

Parallel programming in MATLAB

An existing MATLAB routine can be easily adapted to make use of several cores or processors by employing the Parallel Computing Toolbox. The first required instruction is named `matlabpool` and is used at both the beginning of the code to declare how many cores are going to be used, as well as at the end of it to close them; see the source code in Listing 2. Check how many cores you have in your machine; that number is the maximum amount you can specify as parameter for the first call to `matlabpool`.

Loops might also be parallelized by using the `parfor` instruction. A restriction is that, to be able to parallelize a loop, all the iterations must be independent from each other. Refer again to the pseudocode in Listing 1 and notice that, unfortunately, it is not the case for the DTW dissimilarity measure, since the computation for each entry of the matrix depends on values computed in previous steps. However, since the DTW distance is typically computed among large sets of signals, a useful parallelization is to distribute the effort at the data level instead of doing it at the program level; in other words, the parallelization consists of assigning different calls of the DTW function to different processing cores. This parallelization strategy, according to the Flynn's taxonomy, is known as the *single program multiple data* (*SPMD*) approach.

The source codes

The source codes of our experiments are shown in two separate listings: Listing 2 (`script.m`) and Listing 3 (`dtwParallel.m`). For the sake of readability, all comments are placed right above the lines to be explained. Listing 2 shows the main script that manages the experiments for parallel computation. An entire dissimilarity matrix from a training set to a test set was computed; the computation was performed for 11 different parallel configurations (from two cores up to 12 cores) plus a sequential run whose execution time is the reference to be beaten. In the latter case—the sequential execution—`script.m` is not `run before` `dtwParallel.m`. Notice in Listing 2 that each run was repeated 25 times to ensure the stability of the reported results. The average elapsed times are reported in Fig. 1.

The comments in both listings were included in such a manner that extra explanations are hopefully not needed to understand the code. However, we provide some additional remarks about specific lines in Listing 3.

- Notice that all of the body of the function, except for the first line, is enclosed in an `spmd` statement. It tells MATLAB that the code will be executed in several workers (cores) simultaneously.
- We previously indicated that, typically, the DTW dissimilarity measure is computed for all pairs from a training (reference) set of time series to another set of test ones. As an example, we have considered a publicly available set of time series named "`synthetic_control`" (freely available at <http://www.cs.ucr.edu/~eamonn/time_series_data/>), which is separated into two independent sets for training and test, each one having 300 time series of length 60. In our code, they are loaded in variables `ref` and `signal`, respectively.
- The program `dtwParallel` is divided into two sessions. The first executes the master node; the second is in charge of the worker node. The master node is executed on the first available node in the pool, reserving it until all the jobs are done. This reservation is carried out by setting the `idle` flag to `-1`. The number of workers depends on the number of cores that have been previously launched by the user in the `matlabpool` minus one, since the core where the master code is executed is only dedicated to its task.
- The parallel execution is defined by the `PARALELL` variable, which is only activated in case that a `matlabpool` having more than one processing unit is defined. The portion of the master code sends data to the workers whether there are jobs in queue and available workers to execute them. The list of pending jobs is managed through the `workerStatus` vector. Variable `i` carries the ID of the test signal to be processed and variable `j` contains the ID of the training signal (reference) against with the DTW measure is going to be computed.
- The `while` loop in the master session permanently verifies the vector of `jobStatus` to check if there are jobs to be attended. In

such a case, an idle worker is found, and the job is assigned to it. The status in the vector is then changed to `1`. The master node is constantly checking the buffer through the `labProbe` function to verify if a worker delivered its result of the DTW measure and, in that case, it labels `jobStatus` as "done" by assigning a `2` to it; similarly, it assigns a `0` to the corresponding entry of the `workerStatus` vector. When all of the jobs of the DTW computation are complete, the master sends a stopping signal to all the workers and, in such a way, they end the waiting process.

The sequential execution is faster than the parallel one with only two cores; this behavior is easily explained by the fact that, when using two cores, one of them—the master—is entirely dedicated to communications while the whole processing load is assigned to the worker.

- Workers perform another `while` loop and read the data signals in their buffers to process them by computing the complete DTW

LISTING 2. The main script.

```
%  File     : script.m
%
%  Author   : A.L. Uribe-Hurtado
%  Date     : May 2014
%  Teammates: M. Orozco-Alzate and E.A. Rodríguez-Soto
%  Project  : Semillero de Investigación en Computación de Alto Rendimiento
%             Universidad Nacional de Colombia - Sede Manizales
%
%  Description :
%
%    This script performs 25 repetitions (outer loop) per each parallel
%    configuration (number of cores), ranging from 2 to 12 cores.
%
%
%
% Used Variables and what they mean:
%    results : matrix to store execution times of each experiment
%    i       : loop variable through the repetitions of the experiment
%    j       : loop variable through the number of cores
%

clear all;
results = zeros(25,11);
for i = 1:25
    for j = 2:12

        % Opening of the parallel computing with j cores
        eval(['matlabpool ' int2str(j)]);

        % Calling of the dtwParallel function and measuring of the elapsed
        % time by using tic and toc
        tic
        dtwParallel();
        results(i,j) = toc;

        % Closing of the parallel computing
        eval('matlabpool close');
    end
  end
```

LISTING 3. The script for parallel computing administration.

```
% function dtwParallel()
%
%  File     : dtwParallel.m
%
%  Author   : A.L. Uribe-Hurtado
%  Date     : May 2014
%  Teammates: M. Orozco-Alzate and E.A. Rodríguez-Soto
%  Project  : Semillero de Investigación en Computación de Alto Rendimiento
%             Universidad Nacional de Colombia - Sede Manizales
%
% Function   : dtwParallel
%
% Purpose    : The function performs either a parallel or a sequential
%              execution of the DTW. The first case (parallel execution)
%              takes place if script.m was called which, in turn, calls
%              dtwParallel
%
% Credits    : This code is based on a template by Carlos Edmundo Murillo-
%              Sánchez, later modified by Claudio Zapata-Arias

function  dtwParallel()
clc; clear all;

% single program, multiple data statement: asks MATLAB to execute the body
% on several workers simultaneously
spmd

    % numlabs is the total number of CPU nodes participating on this pool
    % Initially, sequential run is assumed (PARALLEL = 0). If numlabs >= 2,
    % parallel execution can be performed (therefore, PARALLEL = 1)
    PARALLEL = 0;
    if ~isempty(numlabs) && numlabs >= 2
        PARALLEL = 1;
    end

    % Sequential execution of the experiment. Its elapsed time is measured
    % by using tic and toc
    if ~PARALLEL
        tic
        fprintf(' Runs sequentially \n')

        % Load data from the file with test time series
        signal=load('synthetic_control_TEST' ,'-ascii');

        % Load data from the file with training time series
        ref  =load('synthetic_control_TRAIN','-ascii');

        % Check the the number of time series (rows) and their length
        % (columns) in the test and training sets, respectively. A matrix
        % "answers" is allocated to store all the pairwise DTW
        % dissimilarity measures from the training time series to the test
        % time series.
        [NFA,NCA]=size(signal);[NFB,NCB]=size(ref);
        answers=zeros(NFA,NFB);
```

(Continued)

LISTING 3. The script for parallel computing administration. *(continued)*

```
    % Sequential exectution of the DTW. The "dtw" function is called
    % NFA x NFB times in order to compute all the pairwise distances
    for i=1:NFA
        A=ref(i,2:end);
        for j=1:NFB
            B=signal(j,2:end);
            answers(i,j) = dtw(A, B, Inf); % see Listing 1
        end
    end
    toc
else % runs in many workers in parallel

     % --CODE TO RUN only on M A S T E R   L A B (labindex#1)
    if labindex == 1
        % Load data from the file with test time series
        signal=load('synthetic_control_TEST' ,'-ascii');
        % Load data from the file with training time series
        ref  =load('synthetic_control_TRAIN','-ascii');

        % Check the the number of time series (rows) and their length
        % (columns) in the test and training sets, respectively. A matrix
        % "answers" is allocated to store all the pairwise DTW
        % dissimilarity measures from the training time series to the test
        % time series.
        [NFA,NCA]=size(signal);[NFB,NCB]=size(ref);
        answers=zeros(NFA,NFB);

        nJobs=NFA*NFB; % total number of jobs
        jobStatus = zeros(nJobs, 1);

        % jobStatus= 0 if sitting waiting, 1 if in processing now and 2
        % if finished
        workerStatus = zeros(numlabs, 1);
        workerStatus(1)=1; %mark the master node

        % workerStatus= 0 if idle, -1 if error, other if busy with i_job
        %-----------------------------------------------

        while any(jobStatus ~= 2) % if still there are no finished jobs
            i_Job = find(jobStatus == 0);  % lists next jobs to work on
            if ~isempty(i_Job)
                % choose the first waiting job to be next
                i_Job = i_Job(1);
            end

            % look for first available worker
            i_idle = find(workerStatus == 0);
            if ~isempty(i_idle)
                i_idle = i_idle(1); % choose the first idle worker
            end

            % schedule idle worker with the next job
            if ~isempty(i_Job) && ~isempty(i_idle)
                % mark worker as busy with t i_Job
```

(Continued)

LISTING 3. The script for parallel computing administration. *(continued)*

```
            workerStatus(i_idle) = i_Job;
            % mark this job as been served on  worker
            jobStatus(i_Job) = 1;
            t = i_Job;
            % Prepare data to be SENT to each worker
            j=mod(t,NFB);i=floor(t/NFB)+1; % Determine row numbers i and j
            if j==0
                i=i-1; j=NFB;
            end
            A=signal(i,2:end);
            labSend(A,i_idle,1);
            B=ref(j,2:end);
            labSend(B,i_idle,2);

            %frintf('i_Job: %2i SENT to worker: %2i\n',t,i_idle)

            %if no worker is idle, try to RECEIVE some worker's answer
        elseif any(jobStatus == 1) % still no worker is available
            %    Requests
            fin_flag=0;
            while ~fin_flag  %waits to receive some result
                %waits to see which worker already finished
                [ fin_flag, i_worker, tag] = labProbe;
            end

            % Some worker finished..., => receive its answer
            if fin_flag == 1
                % which job was this worker processing?
                t = workerStatus(i_worker);
                % Determine row numbers i and j
                j=mod(t,NFB);i=floor(t/NFB)+1;
                if j==0
                    i=i-1; j=NFB;
                end

                data3 = labReceive(i_worker, tag);
                answers(i,j)=data3;
                workerStatus(i_worker) = 0; % mark worker as idle
                jobStatus(t)       = 2; % Mark this i_Job as finished
                % fprintf('data3: %g  for i_Job: %2i Received...
                % from worker: %2i\n',data3,t, i_worker)
            end % if fin_flag
        end % if can match a job and a worker
    end % while there are pending jobs on queue
    stop=[];
    for i_worker=2:numlabs
        % Order each worker to finish and get out
        labSend(stop,i_worker,1);
    end

    %toc
    %answers
    % -------------- master end -------------
```

(Continued)

LISTING 3. The script for parallel computing administration. *(continued)*

```
        else % ------ code to run only on   W O R K E R S ---------
            % Worker task for multiple processor implementation
            A  = labReceive(1, 1);
            while ~isempty(A)
                B = labReceive(1, 2);
                min_distance = dtw(A, B, Inf);
                labSend(min_distance,   1, 1001);
                A = labReceive(1, 1);
            end
        end % if labindex ==1
        % ------------ worker end ------

    end %if ~PARALLEL
    beep,pause(0.2),beep,pause(0.2), beep
end % spmd
```

measure and return the answer through the `labSend` function (the `1001` was arbitrarily chosen). The worker verifies its buffer and executes its job until the master sends a stopping signal.

- The program determines the status of both each worker node and each job by using the flags in the `workerStatus` and `jobStatus` vectors, respectively. The worker manages three values for `status`: `0` for idle; a number between `1` and the number of jobs determines which one is currently being processed; and `-1` indicates that an error has occurred. The worker status has a maximum length equal to the number of processing units.
- The maximum length of `jobStatus` is equal to the number of jobs to be carried out. For instance, for the training set of 300 time series and the test set of 300 time series used for the experiments, `jobStatus` has 90,000 positions. Notice that `jobStatus` also has three possible values: `0` stands for *waiting to be processed*; `1` means *currently under execution*; and `2` stands for *done*. In this way, an asynchronous control of the computation performed by each worker node is carried out.
- Data that is sent or received by both the master and the worker node is transferred by using message passing interface (MPI).

Experimental results

Experiments were performed on a Dell Workstation Precision T7500 with 50 GB of RAM and two Intel Xeon CPUs X5660 at 2.8 GHz, each one having six cores. Therefore, there are up to 12 cores available in the machine to perform multicore processing. The MATLAB version used in the machine was the 2013a release. The averaged results of the 25 runs are shown in Fig. 1. The standard deviations are shown in a separate figure (Fig. 2) for the sake of clarity; notice that they are always very low in comparison with the averaged execution times.

From Fig. 1, the following general observations are made: the sequential

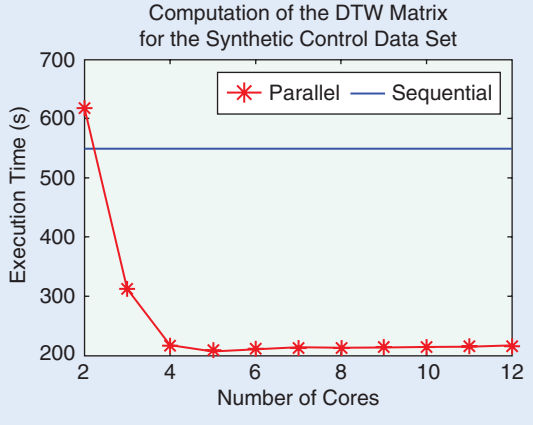


FIG1 The averaged execution times for 25 repetitions of the experiments.

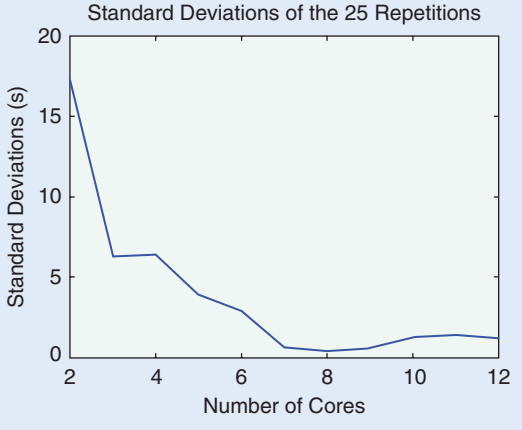


FIG2 The standard deviations associated to the averaged results shown in Fig. 1.

Currently, most of the commercially available desktops and laptops for domestic use are multicore but, in spite of that, people are typically not taking advantage of this feature in their programming applications.

execution is faster than the parallel one with only two cores; this behavior is easily explained by the fact that, when using two cores, one of them—the master—is entirely dedicated to communications while the whole processing load is assigned to the worker. A rough estimation of the communication costs is obtained by subtracting the execution time when using two cores from the time of the sequential execution: this difference is 69.32 s in our case.

In contrast, for more than three cores, the parallel execution is remarkably faster than the sequential one (from three cores, an improvement of 236.57 s and up to 340.13 s for five cores). It seems that using more than five cores does not help for reducing the computation time; therefore, the parallel implementation explored by us for the DTW dissimilarity matrix is not linearly scalable in the number of cores. This, in part, is due to Amdahl's law, which theoretically explains the asymptotic behavior of a parallel implementation as the number of processing units grows; in addition, costs associated to communications eventually become more expensive than the task to be parallelized itself. Further studies on these factors might be an object of future work.

Notice also that most of the contribution in reducing the computation time is observed when going from two cores to three cores, followed by the decrease observed when using four cores. In general, the computation time can be effectively reduced to almost a third by using the parallel implementation. For other applications, in which the `parfor` structure could be used, this improvement must be even more remarkable.

Currently, most of the commercially available desktops and laptops for domestic use are multicore but, in spite of that, people are typically not taking advantage of this feature in their programming applications. In this article, we have shown how to profit from several cores in a machine by using MATLAB parallel programming and illustrated it with an example—the computation of a full DTW dissimilarity matrix—that can be easily replaced by another custom application requiring the SPMD parallelization approach.

Acknowledgments

We would like to express our appreciation for the financial support from Universidad Nacional de Colombia, project (Hermes) 19224, *Semillero de Investigación en Computación de Alto Rendimiento*, Convocatoria: Programa Nacional de Semilleros de Investigación, Creación e Innovación de la Universidad Nacional de Colombia 2013–2015. The anonymous reviewers are also acknowledged for their valuable comments and suggestions to improve the article as well as Carlos Edmundo Murillo-Sánchez and Claudio Zapata-Arias for their collaborations with the source code.

Read more about it

- J. M. Mier and N. Oberg, "Parallel computing with MATLAB: Step by step guide (beginners' edition)," Tech. Report., Civil & Environmental Engineering Dept., Univ. Illinois at Urbana-Champaign, 2010.
- J. Lin, S. Williamson, K. D. Borne, and D. deBarr, "Pattern recognition in time series," in *Advances in Machine Learning and Data Mining for Astronomy*, Boca Raton, FL: CRC, 2012, pp. 617–646.
- E. Keogh, Q. Zhu, B. Hu, Y. Hao, X. Xi, L. Wei, and C. A. Ratanamahatana. (2011). The UCR time series classification/clustering homepage. [Online]. Available: www.cs.ucr.edu/~eamonn/time_series_data/

About the authors

Ana Lorena Uribe-Hurtado (alhurtadou@unal.edu.co) is with Departamento de Informática y Computación at the Universidad Nacional de Colombia—Sede Manizales. She earned her B.Eng. (systems engineering) and M.Eng. (computer science) degrees in 1994 and 2004, respectively. Her main research interests include computer networks, communications, and parallel and distributed computing.

Mauricio Orozco-Alzate (morozcoa@unal.edu.co) is with the Departamento de Informática y Computación at the Universidad Nacional de Colombia—Sede Manizales. He earned his B.Eng. (electronic engineering), M.Eng. (industrial automation) and Dr.Eng. (automatics) degrees in 2003, 2005, and 2008, respectively. His main research interests encompass pattern recognition and digital signal processing and its application to the analysis and classification of seismic, bioacoustic, and hydro-meteorological signals.

Efraín Alberto Rodríguez-Soto (efarodriguezso@unal.edu.co) is pursuing his undergraduate degree in administration of information systems at the Universidad Nacional de Colombia—Sede Manizales.

P

An elegant home automation system using GSM and ARM-based architecture

V.L.K. Bharadwaj Manda, Voona Kushal, and N. Ramasubramanian

©ISTOCKPHOTO/D3DAMON

Home automation systems have been developed recently using various technologies, such as the Internet, wireless networks, Bluetooth, and voice commands, among others. Our proposed research provides a cost-effective system that helps know and control the status of different home appliances. The technology here is the Global System for Mobile Communication (known as *GSM*), and the central processing unit of this system is the NXP LPC11U24 microcontroller unit (the ARM "mbed" microcontroller), which is designed especially to prototype low-cost Universal Serial Bus (USB) devices and other applications that are battery powered. The specific advantages of using short message service (SMS) for intimating the user, over e-mail or any Internet-based messaging techniques, are:

1) Text messages are immediately and directly delivered to the user's mobile phone, which is carried by

Digital Object Identifier 10.1109/MPOT.2016.2515644
Date of publication: 6 September 2018

The research provided here aims at studying the feasibility of implementing an SMS-based control of home appliances using the GSM technology without trying to access other local networks.

him/her almost all of the time. As soon as a message is delivered, an acknowledgment indicating delivery appears on the sender's mobile, thereby providing assurance to the sender. In the case of an Internet-based messaging technique, latency problems of message delivery exist as well as problems such as spam or other e-mail filters, which do not guarantee the delivery of messages.

2) This system can be used in any environment and is free from geographical limitations. It can be used anywhere the GSM network is available. Though this is the same case as that of the Internet, the proposed system does not at all depend on an Internet connection, unlike other systems that require continuous connection, thus incurring additional costs. This makes the proposed system more cost-effective and universal, catering to the needs of any common person.
3) Another major advantage realized by employing the GSM technology in the field of home automation is that the GSM has a higher security infrastructure; in other words, it provides maximum reliability as others cannot monitor the information sent or received. But this is not the case with Internet-based messaging, as the network is vulnerable to attacks. Though there are some industry-proven techniques for facing some threats, incorporating these into a home automation system makes it very complex and also adds to the expenses.

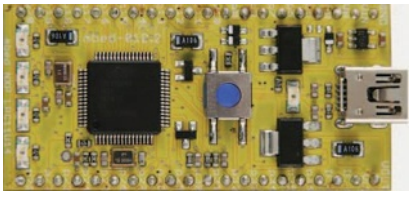


FIG1 An NXP LPC 11U24 microcontroller.

The research provided here aims at studying the feasibility of implementing an SMS-based control of home appliances using the GSM technology without trying to access other local networks. Also, it is affordable to all classes of people as the hardware used in it is inexpensive. Mobile phones are very common these days—almost everyone owns and knows how to send an SMS. This makes the system a real-time application.

Related works

Many home automation systems have been developed previously, which are designed based on the concepts of wireless networks like Zigbee modules, the Internet, voice commands, and infrared rays, to name a few. Baris et al. have proposed a home automation system that uses multiple communication methods, such as GSM, Internet, and speech commands, together in their system. The user can choose any one one of the aforementioned communication methods to control the appliances. This method involves communication between the home appliances using radio-frequency identification communication (RFID).

The research work by Khusvinder et al. presents a home automation system based on Zigbee, where all home appliances are connected to the network and a personal computer as an end user. This system requires an Internet-enabled device with Java support to remotely control the appliances.

Al-Ali et al. have described a home automation system based on Java, wherein all home appliances are connected to an embedded system board and the remote sensing of appliances is achieved using the Internet connection at the house.

Felix et al. offered a system based on the wireless Zigbee technology (based on radio-frequency communications), where the transmitter communicates with every node present in the home. A GSM module is used to facilitate data flow between the microcontroller and user. The user can send commands via SMS to the controller and thus achieve control of the appliances.

Alkar et al. suggest a home automation system where every appliance is connected through a server to one central node and uses the Internet connection to achieve control of appliances. There are other communication protocols being used to control the devices.

System hardware

The major hardware components in the proposed system are:

- *The NXP LPC11U24 microcontroller:* Figure 1 shows the central processing unit of this system is the LPC11U24 microcontroller that is often referred to as the *mbed* microcontroller. It is designed to prototype low-cost USB devices. It comes as a package that contains a small dual in-line package (DIP) form-factor and is used for prototyping with some of the devices such as through-hole printed circuit boards and breadboards and the like. In addition, it includes a built-in USB Flash-programmer. It includes 32-kB flash memory, 16-kB random access memory (RAM), and interfaces like Serial Peripheral Interface (SPI) and analog-to-digital converter (ADC), among others. Its specific features are the ability to run on a low-power ARM Cortex-M0 Core. It runs at a speed of 48 MHz and uses the Inter Integrated Circuit (I^2C) serial computer bus. It has an inbuilt USB drag-and-drop Flash programmer to communicate with other devices (mainly the computer). The prototyping form-factor is as follows: 40-pin 0.1-in pitch DIP package, 5-V

USB, 4.5-9-V supply, and 2.4-3.3-V battery.

- *GSM SIM 300 Module:* The GSM module, shown in Fig. 2, can accept the subscriber identification module (SIM) card of any network operator. It is similar to a cellular phone with a unique phone number. The main advantage of this module is the utilization of serial communication and the resultant development of applications based on embedded systems. This modem can be directly connected to any microcontroller or personal computer. It can send/receive SMS and make/receive phone calls. When used in general packet radio service (GPRS) mode, it should be connected to the Internet, and it can perform tasks like data transfer. Other applications include SMS control, remote control, data logging, security applications and sensor monitoring and highly reliability for continuous monitoring with a matched antenna. It is very inexpensive and simple to use. It also contains a quad-band modem that supports all GSM-operator SIM cards. We use attention (AT) commands to control the various operations of the GSM modem.
- *Four-channel relay board:* Relays are special switches that open and close electrical and electronic circuits, either electro-mechanically or just electronically. They are a type of control device that are used to control an electrical circuit by maintaining or removing its contact with another circuit. When a relay contact is normally open, there is no contact, and when the relay contact is normally closed, there is a contact maintained. In both cases, when electrical current is applied to the contacts, there is a change of state. Relays switch smaller currents present in a control circuit and do not, in general, control the devices that consume power, except for some small motors and a few solenoids that draw significantly less current. However, highly advanced relays can be used to protect electrical systems against troubles and shutdowns, as they can regulate and control the generation and distribution of electric current. To gain control of numerous appliances, we use an n-channel relay board, which contains light-emitting diodes (LEDs) that correspond to the status of appliances. In our work, a four-channel relay board, as shown in Fig. 3, is considered.
- *Sensor modules:* Home automation includes the control of electrical appliances as well as the detection of fire and the leakage of liquid petroleum gas (LPG). For this purpose, we use modules such as fire/smoke and gas sensors to obtain the status of the house.

To gain control of numerous appliances, we use an n-channel relay board, that contains light-emitting diodes (LEDs) that correspond to the status of appliances.

Circuit description and implementation

Appliances or devices in a house are connected to the four-channel relay board. The relay board consists of four independent relays mounted on a single board. Each relay is connected to one LED, which indicates the status of the appliance connected to it. There are six pins on the board, out of which one is for the power supply (12 V), one for the ground connection, and the remaining four are control pins for each of the four relays. This relay board is interfaced with the microcontroller. The control pins of the relay board are connected to the input and output ports of the microcontroller. The GSM SIM 300 modem is interfaced to the microcontroller by means of serial communication. The ground pin of SIM 300 is connected to the microcontroller's ground, and the transmission and receiving pins are connected to the receiving and transmitting pins, respectively, of the microcontroller. The sensor modules, such as the temperature, humidity, and LPG gas sensors, are placed on the same board where the microcontroller is located and connected based on their pin configurations. The power supply is then given to the microcontroller, relay board, and the SIM 300 module. The block diagram is shown in Fig. 4, and the circuit connections are displayed in Fig. 5.

Once the circuit connections are made, a valid SIM card is put in the module and authenticated. The user's mobile number is verified and set as default for sending SMS. When the user wants to turn on/off an appliance, a message is sent from the user's mobile phone to the number placed in the module. This turns on/off the device through the relay board. In addition to controlling the device, we can also learn the status of each appliance and obtain information about the temperature and humidity. The messages used for communication between the user and the system are summarized in Table 1, and the real implementation

FIG2 The GSM SIM 300 module.

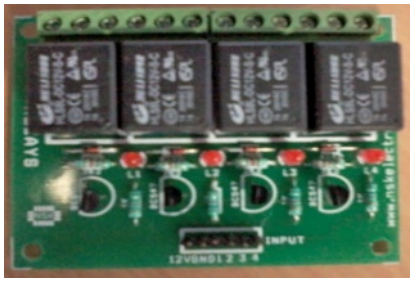


FIG3 A four-channel relay board.

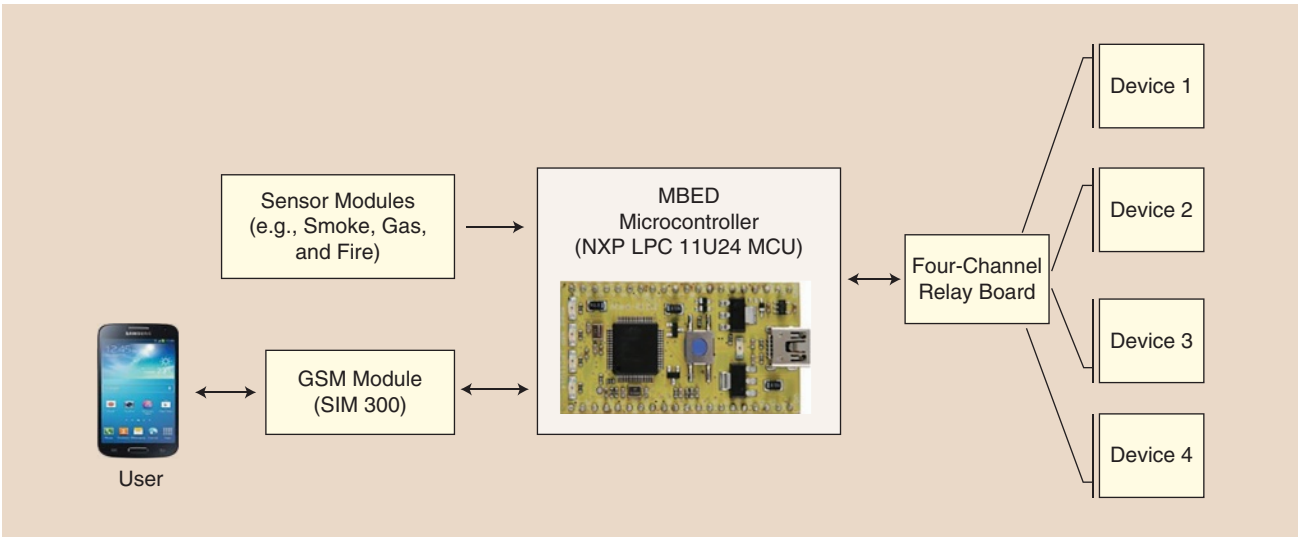


FIG4 A block diagram of our work.

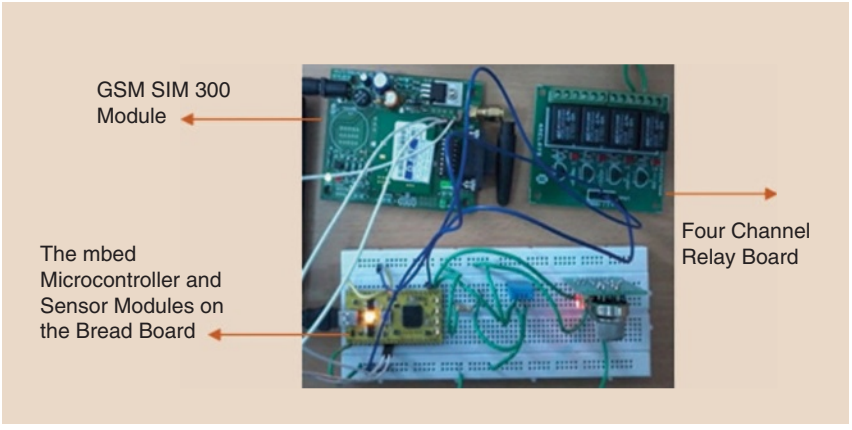


FIG5 The circuit connections.

of switching on and off an appliance is shown in Figs. 6 and 7, respectively. In situations like gas leakage or a fire, the sensors detect changes in the surroundings and alert the user through the microcontroller-interfaced GSM modem, and the user can take necessary and timely action to avoid further damage.

Software used

The online mbed development platform (www.mbed.org) is used for the implementation of this system. It provides free software libraries, the required hardware designs, and other necessary online tools for professional prototyping of the microcontroller. The platform includes an open-source software development kit (SDK), based on C/C++ for rapid project development, and is licensed under Apache 2.0. The SDK is designed to provide enough hardware abstraction to build powerful and complicated projects. Details of hundreds of reusable peripheral device module libraries is provided, in addition to real-time operating system, USB, and networking libraries.

A hardware development kit (HDK) that provides full support for microcontroller subsystem designs is also included for building the development boards and other products that make use of the development platform. The HDK includes

TABLE 1. A list of messages sent and the corresponding actions.

S.NO.	MESSAGE SENT	ACTION
1.	**Status**	Returns the status of the appliances.
2.	**Load 1 OFF**	Turns off the device connected to position 1 on the four-channel relay board.
3.	**Load 2 OFF**	Turns off the device connected to position 2 on the four-channel relay board.
4.	**Load 3 OFF**	Turns off the device connected to position 3 on the four-channel relay board.
5.	**Load 4 OFF**	Turns off the device connected to position 4 on the four-channel relay board.
6.	**Load 1 ON**	Turns on the device connected to position 1 on the four-channel relay board.
7.	**Load 2 ON**	Turns on the device connected to position 2 on the four-channel relay board.
8.	**Load 3 ON**	Turns on the device connected to position 3 on the four-channel relay board.
9.	**Load 4 ON**	Turns on the device connected to position 4 on the four-channel relay board.

all components and circuits that support the mbed microcontroller. A simple drag-and-drop USB Flash programmer is provided to make the programming of the microcontroller very easy. An online compiler is also available for programming the microcontroller, which relies on the ARM standard C/C++ compiler, and is configured to test and compile the code efficiently. One main advantage of the compiler is its private workspace feature, though it still provides access to the developer website. A snapshot of the online compiler is shown in Fig. 8.

Future enhancements

Future improvements include the implementation of home automation using the cloud, where the home automation technology can be extended with a central server for each house that monitors every connected node in the home. Also, an Ethernet connection to home appliances can be established and the audio/video devices added to the already-established network to achieve control (e.g., playing music or video). This provides a higher level of automation while also incurring expenses.

Comparative study

The details mentioned in Table 2 provide information for a comparative study of home automation systems that are currently in use as well as our proposed system. The key feature here is the microcontroller used, the ARM mbed LPC11U24 [up to 32-kB on-chip Flash memory, up to 4-kB on-chip data electrically erasable programmable read-only memory (EEPROM), low working voltage range of 1.8–3.6 V, and a wide operating frequency range of 1–25 MHz], which provides additional advantages to the proposed architecture. The full details are listed in Table 2.

Conclusion

The feasibility of SMS-based control of home appliances using the GSM technology was examined in this article. The conceivable challenges in the practical implementation of it could be packaging the hardware into a single unit and making it a commercial product. Also, if the system was installed at a place that experiences a poor or no signal, the performance of the system might be affected, but GSM signal boosters could be set up at the home so as to enhance the signal for the GSM module and increase performance. The control of home appliances using wireless GSM technology can have a great impact and revolutionize people's lifestyles. The technique used in our system is very simple: through inexpensive GSM technology and mobile phones, which are easily available and used

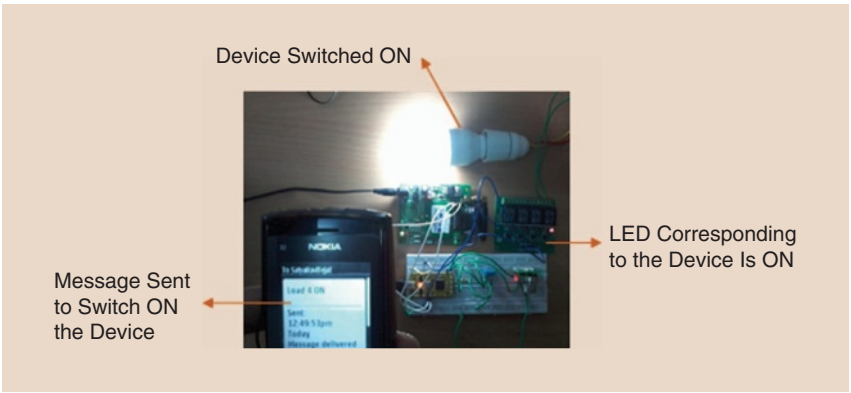


FIG6 Turn ON any device.

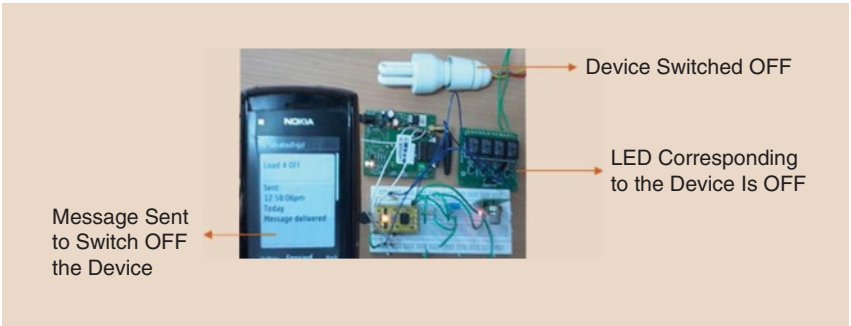


FIG7 Turn OFF any device.

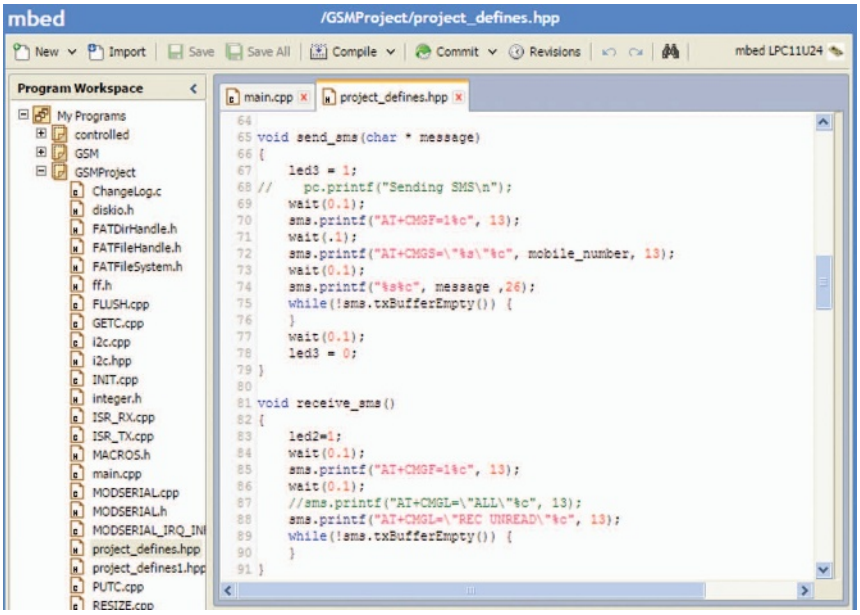


FIG8 The online compiler for programming the microcontroller.

TABLE 2. Comparison of existing system versus proposed system (as specified in product data sheets).

S.NO.	CHARACTERISTIC	EXISTING SYSTEMS	PROPOSED SYSTEM
1.	Microcontroller used	8051, PIC and Arduino	NXP LPC 11U24
2.	Software flexibility	No online support available for software execution	Free, portable online compiler and open-source SDK
3.	Peripheral devices used	Use of additional components for serial communication like MAX232	A simple, built-in, drag-and-drop USB Flash programmer is provided for communication, without the need for additional components
4.	Area occupied	The space occupied by the home automation units is fairly high	Requires less space due to small and simple circuitry
5.	Cost-effectiveness	Use of wireless networks (e.g., Zigbee) and Internet is costly	Use of a GSM module and a simple mobile phone makes it cost effective, reliable, and used by the general public
6.	Memory management	Up to 256 B of data EEPROM and external Flash memory provision	Up to 32-kB on-chip Flash program memory and up to 4-kB on-chip EEPROM data memory, along with 10-kB SRAM data memory
7.	Power consumption	2–5.5 V	1.8–3.6 V
8.	Clock generation	Up to 10 MHz	Wide operating range of 1–25 MHz
9.	Code security	No provision for code security	Uses code read protection mode so the access to the on-chip Flash can be restricted

by a large population, real-time application can be achieved.

Read more about it

- B. Yuksekkaya, A. Alper Kayalar, M. B. Tosun, M. K. Ozcan, and A. Z. Alkar, "A GSM, Internet and speech controlled wireless interactive home automation system," *IEEE Trans. Consum. Electron.*, vol. 52, no. 3, pp. 837–843, Aug. 2006.
- K. Gill, S. H. Yang, F. Yao, and X. Lu, "A Zigbee based home automation system," *IEEE Trans. Consum. Electron.*, vol. 55, no. 2, pp. 422–430, May 2009.
- A. R. Al-Ali and M. AL-Rousan, "Java-based home automation system," *IEEE Trans. Consum. Electron.*, vol. 50, no. 2, pp. 498–504, May 2004.
- C Felix and I. Jacob Raglend, "Home automation using GSM," in *Proc. IEEE Int. Conf. Signal Processing Communication Computing and Networking Technologies*, Thuckafay, India, July 2011, pp. 15–19.
- A. Z. Alkar and U. Buhur, "An internet based wireless home automation system for multifunctional devices," *IEEE Trans. Consum. Electron.*, vol. 51, no. 4, pp. 1169–1174, Nov. 2005.
- M. A. Zamora-Izquierdo, J. Santa, and A. F. Gómez-Skarmeta, "An integral and networked home automation solution for indoor ambient intelligence," *IEEE Pervasive Comput.*, vol. 9, no. 4, pp. 66–77, Jan. 2010.
- K. Balasubramanian and A. Cellatoglu, "Analysis of remote control techniques employed in home automation and security systems," *IEEE Trans. Consum. Electron.*, vol. 55, no. 3, pp. 1401–1407, Oct. 2009.
- R. Teymourzadeh, S. A. Ahmed, K. W. Chan, and M. V. Hoong, "Smart GSM based home automation system," in *Proc. IEEE Conf. Systems Process and Control*, Kuala Lumpur, Malaysia, Dec. 2013, pp. 306–309.

About the authors

V.L.K. Bharadwaj Manda (mvlkbcse@gmail.com) earned his B.Tech degree from the Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli, India. He is currently a direct Ph.D. scholar in the Department of Engineering Design, Indian Institute of Technology Madras, India. His areas of interest include embedded systems, computer architecture, computer vision, and deep learning.

Voona Kushal (kushal002@gmail.com) earned his B.Tech degree in the Department of Instrumentation and Control Engineering, National Institute of Technology, Tiruchirappalli, India. He is currently pursuing his M.S. (by research) degree in interdisciplinary engineering, Indian Institute of Technology Madras, India. His areas of interest include embedded systems, smart structures and systems, and control systems.

N. Ramasubramanian (nrs@nitt.edu) earned his B.E. (electronics and communication), M.E. (computer science), and Ph.D. (multicore computer architecture) degrees from the National Institute of Technology, Tiruchirappalli, India. He is a professor in the Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli, India. His areas of interest include multicore computer architecture, advanced digital design, distributed systems, real-time embedded systems, and reconfigurable computing.

P

NUMBERS—© CAN STOCK PHOTO/AGSANDREW,
ANDROID—© CAN STOCK PHOTO/KIRSTYPARGETER

Problem #1: Flow Rider

An oarsman leaves his boathouse on the river and rows upstream at a steady rate. After 2 km, he passes a log that is floating down the river. He continues on for 1 h and then turns around and rows back downstream. He overtakes the log just as he reaches the boathouse. What is the flow velocity of the river?

Problem #2: Der Vampyr

In South Strangeland, sane humans and insane vampires make only true statements. Insane humans and sane vampires make only false statements. Two sisters, Anna and Betsy, call the suburbs of South Strangeland home. We know that one of the sisters is human and the other is a vampire, but we know nothing about the sanity of either woman. Anna says, "We are both insane!" Betsy replies, "That's not true." Which sister is the vampire?

Problem #3: It's the Magic Number

How can you obtain "3" by using only zeros and common mathematical symbols?

Problem #4: Belly Up to the Bar

There are two straight iron bars that look identical. One is a magnet, and the other is not. How can you tell which is which by just touching them to each other?

Problem #5: Tunnel Vision

An engineer entered the north end of a tunnel of length L. After walking the distance $L/4$ into the tunnel, she noticed a car approaching the north entrance at 40 mi/h. The engineer knew her own speed and calculated that no matter which end of the tunnel she ran to, she would arrive there at the same time as the car. What was her top speed? (Hint: she might do better with a career as a professional athlete than as an engineer.)

P

If you have a problem for the Gamesman, please submit it along with the solution to potentials@ieee.org.
Solutions are on page 6.

Digital Object Identifier 10.1109/MPOT.2018.2846898
Date of publication: 6 September 2018

Some things don't make sense

Some things do.

IEEE Member Group Term Life Insurance Plan — It just makes sense.

IEEE members now have access to a NEW online term life insurance tool—QuickDecision℠! With QuickDecision℠, members can apply for up to $500,000 in Group Term Life Insurance with no medical exam required, just answer some health questions and other information. Plus, you'll find out in as little as one day if you're approved. Everything is handled conveniently and securely online!*

Want to join thousands of your fellow IEEE members?
Apply online with QuickDecision℠ today!

Learn more about the IEEE Member Group Term Life Insurance Plan.
Visit **IEEEinsurance.com/Potentials**.**

IEEE

Program Administered by Mercer Health & Benefits Administration LLC

In CA d/b/a Mercer Health & Benefits Insurance Services LLC

AR Insurance License #100102691
CA Insurance License #0G39709

*QuickDecision℠ is not available in Canada or Puerto Rico. May not available in all states. If you are not approved for coverage under the QuickDecision℠ tool, you may still apply online, but will follow the fully underwritten process.

**For information on features, costs, eligibility, renewability, limitations and exclusions. The Group Term Life Insurance Plan is available only for residents of the U.S. (except territories), Puerto Rico and Canada (except Quebec). This plan is underwritten by New York Life Insurance Company, 51 Madison Ave., New York, NY 10010 on Policy Form GMR. This plan is administered by Mercer Consumer, a service of Mercer Health & Benefits Administration LLC. This coverage is available to residents of Canada (except Quebec). Mercer (Canada) Limited, represented by its employees Nicole Swift and Suzanne Dominico, acts as broker with respect to residents of Canada.