



**ქ. თბილისის ივ.ჯავახიშვილის სახელობის
სახელმწიფო უნივერსიტეტი
ზუსტ და საბუნებისმეტყველო მეცნიერებათა
ფაკულტეტი**

გაგა გოგოლაური

ავტომატური შეტყობინების სისტემა

**ნაშრომი შესრულებულია ინფორმაციული ტექნოლოგიების მაგისტრის
აკადემიური ხარისხის მოსაპოვებლად**

ხელმძღვანელი: პაპუნა ქარჩავა

თბილისი 2015

ა ნ ო ტ ა ც ი ა

ნაშრომში განხილულია ავტომატური შეტყობინების სისტემების არსი, მათი გამოყენების სფეროები და რეალიზაციის ტექნოლოგიები. უმთავრესი ყურადღება გამახვილებულია ავტომატური შეტყობინების სისტემის ერთერთ სახეზე Email-ზე. მის მთავარ უპირატესობებს წარმოადგენს: ადამიანების კომუნიკაციის სიმარტივე, როგორც ოფისის შიგნით ასევე მსოფლიოს ნებისმიერ წერტილში; შეტყობინებების სწრაფი გადაგზავნის შესაძლებლობა; კლიენტების დასმულ შეკითხვებზე პასუხის გაცემის სიმარტივე და სისწრაფე; დაბალი ღირებულება (რომელიც გულისხმობს ინტერნეტთან დაკავშირებულობას); გამოყენების სიმარტივე; სხვადასხვა ფორმატის ფაილების (მაგალითად, როგორიცაა, სურათები, ვიდეო- და აუდიო- ფაილები, მონაცემთა ბაზები და ა.შ.); დროის დაზოგვა, აუცილებლობას არ წარმოადგენს ძვირადღირებული დროის ხარჯვა ოფისში სასიარულოდ; იზოგება ქალაქის და პრინტერის ხარჯები. ასეთი შეტყობინების სისტემები გამოიყენება ბანკებში, ორგანიზაციებში, საავადმყოფოებში, საწარმოებში, სახელმწიფო დაწესებულებებში, ონლაინ მაღაზიებში და ა.შ. მისი რეალიზაციისათვის უმთავრესად გამოიყენება სხვადასხვა პროგრამული ენები (მაგალითად, .Net, PHP, Python). ნაშრომში აღწერილია ავტომატური შეტყობინების სისტემის ერთერთი სახის (Email) შექმნის ინსტრუმენტები და მოცემულია მისი რეალიზაცია პროგრამული კოდის სახით. აღნიშნული ძრავი იძლევა ელექტრონული ფოსტის გაგზავნის შესაძლებლობას. ის არის ღია (Open Source) გამოყენებისთვის და ნებისმიერს შეუძლია მისი გამოყენება და, აუცილებლობის შემთხვევაში, საკუთარი მოთხოვნილების შესაბამისად ცვლილებების შეტანა მასში.

ნაშრომში განხილულია ელექტრონული ფოსტის გამოყენების არქიტექტურა და მისი ოთხი სცენარი. მომხმარებლის აგენტი (UA), სერვისები, რომელსაც ის გვთავაზობს და მომხმარებლის აგენტის ორ ტიპი. შეტყობინების გაგზავნის აგენტი (MTA) და შეტყობინების წვდომის აგენტი (MAA). MIME, როგორც ფუნქციების ნაკრები, რომელიც გარდაქმნის non-ASCII მონაცემებს ASCII მონაცემებად. განხილულია ელექტრონული ფოსტის გაგზავნის და მიღების მექანიზმი, მისი გადაცემის ფაზები. განმარტებულია web- შეტყობინების სისტემის ორი შემთხვევა და ელექტრონული ფოსტის უსაფრთხოება.

A n n o t a t i o n

The article deals with the essence of automated message systems, their application and implementation technologies. The main focus is on the automatic warning system on the face of one of the Email. the advantages are that It is very easy to communicate effectively with anybody within the office or anywhere in the world regardless of where they are situated. you can respond to clients quickly and easily. This means that you no longer need to spend hours on the phone, trying to get through and leaving messages with receptionists. Messages can be sent quickly, in an instant. The cost of sending an email is very low, unlike postage and other methods of communication. It is easy to use, simply type the name of the recipient, a subject line and your message and click the send button. You can copy others on correspondence. A large number of people can be copied and communicated with at once. You can send attachments, such as photographs, files and spreadsheets. Email saves time. No need to spend valuable time going to someone else's office. Emails can be responded to in your own time, when it is convenient for you. It is not invasive like a visitor or phone call which requires immediate attention. It speeds up the workflow process, documents can be sent for comment, corrections can be made quickly. Emailing saves paper and printing costs. Such private systems are used in banks, offices, hospitals, enterprises, public institutions, online stores, etc. For its realization used a variety of programming languages (.Net, PHP, Python). The article describes one of the types of automated message system-Email, its creation tools and the realization of its software code. This allows the engine to send e-mail. The engine is an Open Source, and anyone can use, copying, distribution and change.

The article describes the architecture of electronic mail using four scenarios. The user agent (UA), services provided by it, and two types of user agents. The role of a message transfer agent and Simple Mail Transfer Protocol (SMTP) as the formal protocol that handles MTA. Two message access agents (MAAs): POP and IMAP. MIME as a set of software functions that transforms nonASCII data to ASCII data and vice versa. Reviewed the mechanism of sending and receiving e-mails and its transfer phases. discussed the idea of Web-based e-mail and the security of the e-mail system.

შესავალი	5
§1. ელექტრონული შეტყობინების სისტემა - ელექტრონული ფოსტა	9
1.1. არქიტექტურა	9
1.2. მომხმარებლის აგენტი (UA)	12
1.3. შეტყობინების გაგზავნის აგენტი (MESSAGE TRANSFER AGENT)	14
1.4. ფოსტის გაგზავნის ფაზები	17
1.5. შეტყობინების წვდომის აგენტი (MESSAGE ACCESS AGENT)	20
1.6. MIME	22
§2. web- შეტყობინების სისტემა	26
§3. ელ-ფოსტის უსაფრთხოება	27
§4. ავტომატური შეტყობინების სისტემა - პროგრამული რეალიზაცია	30
დასკვნა	32
გამოყენებული ლიტერატურა	34
ავტომატური შეტყობინების სისტემა (პროგრამული კოდი)	35

შესავალი

ოცდამეერთე საუკუნეში ადამიანებს შორის კომუნიკაციის განუყრელ საშუალებას წარმოადგენს ავტომატური შეტყობინების სისტემა. არსებობს მისი შემდეგი ფორმები: SMS, MMS, EMAIL. აღნიშნულ სერვისებს ფართოდ გამოიყენება ბანკებში, ორგანიზაციებში, საავადმყოფოებში, სანარმოებში, სახელმწიფო დაწესებულებებში, ონლაინ მაღაზიებში და ა.შ.

SMS (Short Messaging Service) [8] ეს არის მოკლე შეტყობინების სისტემა. მისი უპირატესობებია:

1. მარტივი ხელმისაწვდომობა;
2. კომუნიკაციის კარგი საშუალებაა;
3. ეფექტური ღირებულება;
4. დროის დანაზოგი.

რაც შეეხება ნაკლოვანებებს შემდეგია :

1. Spam შეტყობინებები;
2. სიმბოლოების ლიმიტი (140-160 სიმბოლო);
3. არაპროფესიონალური მეთოდი ბიზნეს ურთიერთობებისთვის.

MMS (Multimedia Messaging Service) [9] ეს არის მულტიმედია შეტყობინების სისტემა. მისი უპირატესობები არის ის, რომ მარტივია გამოსაყენებლად. შესაძლებელია არა მხოლოდ ტექსტის, არამედ სურათების, მოკლე ვიდეოების, ხმოვანი შეტყობინების და ზოგიერთი ფაილის გაგზავნა. SMS-ისგან განსხვავებით შესაძლებელია 160-ზე მეტი სიმბოლოს გაგზავნა.

მისი ნაკლოვანებები არის ის, რომ თუ MMS შექმნილია ერთი ბრენდის ტელეფონზე, შეიძლება არ იყოს თავსებადი სხვა ბრენდის ტელეფონთან. იმის გამო, რომ ხდება დიდი რაოდენობით MMS-ების გაგზავნა სხვადასხვა ქსელში, მომხმარებლების დიდი ჯგუფისთვის MMS-ის გაგზავნის პროცესი შეიძლება იყოს ძალიან ნელი.

ერთ-ერთ ყველაზე პოპულარულ უფასო ინტერნეტ სერვისს წარმოადგენს ელექტრონული ფოსტა (e-mail) [5]. უმსხვილესი ონლაინ მაღაზიების 66% იყენებს ელ-ფოსტას მისასალმებელი შეტყობინების გასაგზავნად. სწორედ ამის გამო ჩვენი ყურადღება შევაჩერეთ ელ-ფოსტის სერვისზე.

ელექტრონული ფოსტა [2], რომელსაც ყველაზე ხშირად მოიხსენიებენ, როგორც email ან e-mail, არის ციფრული შეტყობინებების გაცვლის მეთოდი გამგზავნისა და ერთ ან რამდენიმე მიმღებს შორის. თანამედროვე ელექტრონული ფოსტა გადაიცემა ინტერნეტის ან კომპიუტერული ქსელების გავლით. ზოგიერთ ადრეულ ელექტრონულ სისტემაში ელექტრონული ფოსტის გადაგზავნის საჭიროებისას აუცილებლობას წარმოადგენდა გამგზავნისა და მიმღების ერთდროული ყოფნა ონლაინ სივრცეში. თანამედროვე ელექტრონული სისტემები ეფუძნება store-and-forward მოდელს, რომლის მიხედვითაც ელექტრონული ფოსტის გადაგზავნის მომენტში აუცილებლობას არ წარმოადგენს არამხოლოდ ონლაინ სივრცეში მომხმარებლის ყოფნა, არამედ კომპიუტერის ჩართვაც (ის შეიძლება იყოს გამორთული).

ელექტრონული ფოსტა შედგება სამი ნაწილისგან: კონვერტი, სათაური და ტანი.

ელექტრონული ფოსტის ძირითადი კომპონენტებია:

- მომხმარებლის აგენტი (UA);
- შეტყობინების გაგზავნის აგენტი (MTA);
- შეტყობინების წვდომის აგენტი (MAA).

პროტოკოლები SMTP, POP, IMAP, MIME რეალიზაციას უკეთებენ ზემოთ აღნიშნულ კომპონენტებს.

თავდაპირველად ინტერნეტ სივრცეში ელექტრონული ფოსტით გაგზავნილი შეტყობინება იყო პატარა და მოიცავდა მხოლოდ ტექსტს. დღეს ელექტრონული ფოსტა გაცილებით უფრო ფუნქციონალურია. მისი საშუალებით შეგვიძლია გავაგზავნოთ ტექსტი, აუდიო და ვიდეო. იგი ასევე იძლევა იმის საშუალებას, რომ შეტყობინება გაეგზავნოს არა ერთ არამედ რამდენიმე მიმღებს. ყველაზე

გავრცელებული ინტერნეტ სერვისებია: Google Gmail, Microsoft Outlook, Yahoo! Mail, Apple iCloud და GMX Mail.

ელექტრონული ფოსტა ფართოდ გამოიყენება ბიზნეს კომუნიკაციების დროს და საერთო ჯამში, ეს არის ძალიან ეფექტური კომუნიკაციის ინსტრუმენტი [4]. ამასთან, იგი არის უფასო. მისი გამოყენებისთვის საჭიროა მხოლოდ ინტერნეტთან დაკონექტება. დაწყებული ორგანიზაციის დირექტორიდან (CEO) დამთავრებული დამლაგებლამდე და ასევე დროებით თანამშრომლებს შეუძლიათ გააგზავნონ და მიიღონ ფოსტა. სამსახურში კომუნიკაცია (ურთიერთობა) ძალიან მნიშვნელოვანია. თანამშრომლები წარმატებით იყენებენ ელექტრონული ფოსტის ბევრ უპირატესობას [3].

1. ეს ძალიან მარტივი გზაა ნებისმიერთან ურთიერთობისთვის, როგორც ოფისში ისე მსოფლიოს ნებისმიერ წერტილში.
2. კიდევ ერთი უპირატესობა არის ის, რომ შენ შეგიძლია მარტივად და სწრაფად უპასუხო კლიენტებს.
3. შეტყობინება იგზავნება სწრაფად.
4. მისი ღირებულება ძალიან დაბალია (ინტერნეტის ღირებულება).
5. მარტივი გამოსაყენებელია. მიუთითებ მიმღებს, ჩანერ საგანს (Subject), შეტყობინებას და აგზავნი.
6. შესაძლებელია ერთი წერილის ბევრ მიმღებთან გაგზავნა.
7. შესაძლებელია დანართების გაგზავნა. მაგალითად, სურათები, ფაილები და ცხრილები.
8. მეილი ზოგავს დროს. არაა აუცილებელი ძვირფასი დროის დახარჯვა ვინმესთან ოფისში სასიარულოდ.
9. იგი აჩქარებს სამუშაო პროცესს.
10. იგი ზოგავს ქალაქის და პრინტერის ხარჯებს.

რაც შეეხება ელექტრონული ფოსტის ნაკლოვანებებს, შემდეგია :

1. შესაძლებელია Spam შეტყობინებების გაგზავნა, რამაც შეიძლება გამოიწვიოს inbox-ის გადავსება.
2. Email hoaxes.

3. ფოსტით შესაძლებელია ვირუსების გაგზავნა.
4. შესაძლებელია მოხდეს გაუგებრობა, თუ შეტყობინებები არ არის სწორად აგებული.
5. ყველას არ აქვს ინტერნეტთან წვდომა.
6. კონფიდენციალური ინფორმაცია შეიძლება გადაიგზავნოს შეცდომით და მოხდეს არასწორ ხელში.

აქვე წარმოდგენილი იქნება ავტომატური შეტყობინების სისტემის პროგრამული კოდი. აღნიშნული ძრავი საშუალებას იძლევა გავაგზავნოთ ელექტრონული ფოსტა. ძრავი არის Open Source და ნებისმიერს შეუძლია მისი გამოყენება, გადანერა, ცვლილება და გავრცელება.

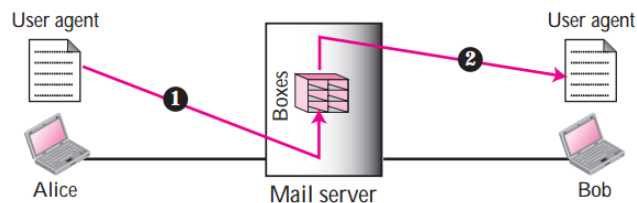
§1. ელექტრონული შეტყობინების სისტემა - ელექტრონული ფოსტა

1.1. არქიტექტურა

ელ-ფოსტის არქიტექტურის ასახსნელად განვიხილავთ ოთხ სცენარს [1]. დავიწყებთ უმარტივესი სიტუაციის განხილვით. მეოთხე სიტუაცია წარმოადგენს ელ-ფოსტის გაცვლის ყველაზე გავრცელებულ ფორმას.

პირველი სცენარი

ამ შემთხვევაში გამგზავნი და მიმღები არიან ერთიდაიგივე საფოსტო სერვერზე. ისინი პირდაპირ არიან დაკავშირებული საერთო საფოსტო სერვერთან. ადმინისტრატორის მიერ შექმნილია ერთი საფოსტო ყუთი, რომელშიც ინახება შემოსული შეტყობინებები. საფოსტო ყუთი არის მყარი დისკის ნაწილი, ფაილი გარკვეული შეზღუდვებით. მასზე წვდომა მხოლოდ საფოსტო ყუთის მფლობელს აქვს. როდესაც ალისას უნდა ბობისთვის შეტყობინების გაგზავნა ის უშვებს მომხმარებლის აგენტის (UA) პროგრამას, რომ მოამზადოს შეტყობინება და შეინახოს საფოსტო ყუთში. შეტყობინებას აქვს გამგზავნის და მიმღების საფოსტო მისამართი. ბობს შეუძლია მიიღოს და წაიკითხოს შეტყობინება მომხმარებლის აგენტის გამოყენებით.

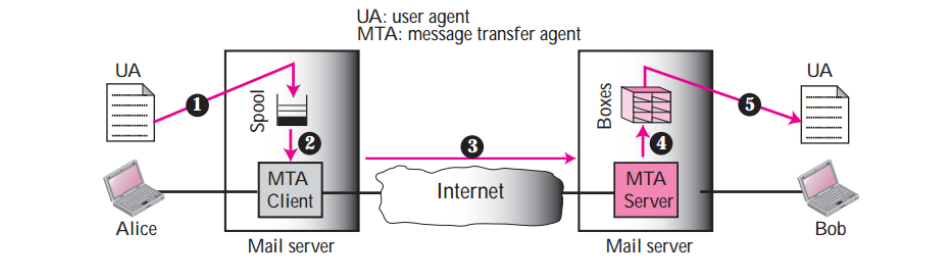


როდესაც მეილის გამგზავნი და მიმღები არიან ერთიდაიგივე საფოსტო სერვერზე, გვჭირდება მხოლოდ ორი მომხმარებლის აგენტი (UA).

მეორე სცენარი

მეორე სცენარში ფოსტის გამგზავნი და მიმღები არიან სხვადასხვა საფოსტო სერვერზე. შეტყობინება იგზავნება ინტერნეტის გავლით. ამ შემთხვევაში გვჭირდება მომხმარებლის აგენტები (UAs) და შეტყობინების გაგზავნის აგენტები (MTAs). ალისას სჭირდება მომხმარებლის აგენტის პროგრამა, რომ გააგზავნოს შეტყობინება საფოსტო სერვერზე საკუთარი საიტიდან. ბობსაც ასევე სჭირდება მომხმარებლის აგენტის

პროგრამა საფოსტო ყუთში არსებული შეტყობინების მისაღებად თავის საიტზე. შეტყობინება ალისას საიტიდან ბობის საიტამდე იგზავნება ინტერნეტის საშუალებით. ამ შემთხვევაში საჭიროა ორი შეტყობინების გაგზავნის აგენტი. ერთი კლიენტი და მეორე სერვერი.

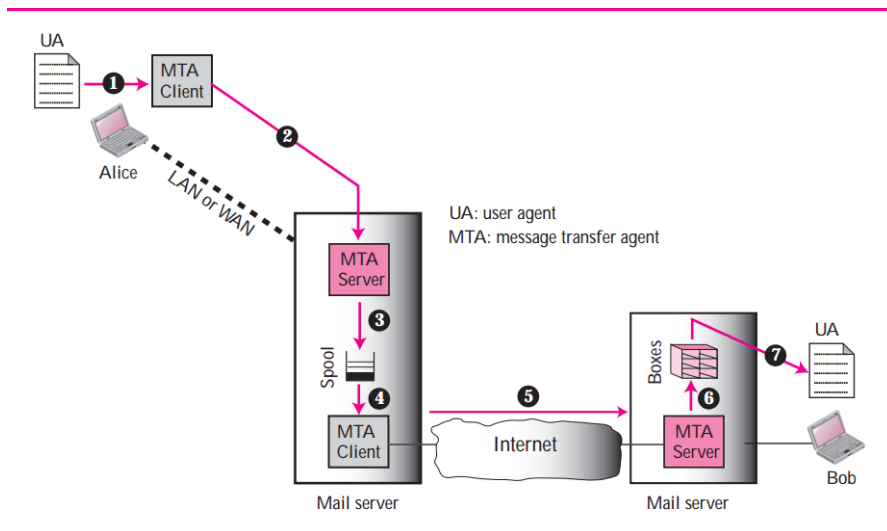


როდესაც მეილის გამგზავნი და მიმღები სხვადასხვა საფოსტო სერვერებზე არიან, საჭიროა ორი მომხმარებლის აგენტი (UA) და ორი შეტყობინების გაგზავნის აგენტი (MTA) (კლიენტი და სერვერი).

მესამე სცენარი

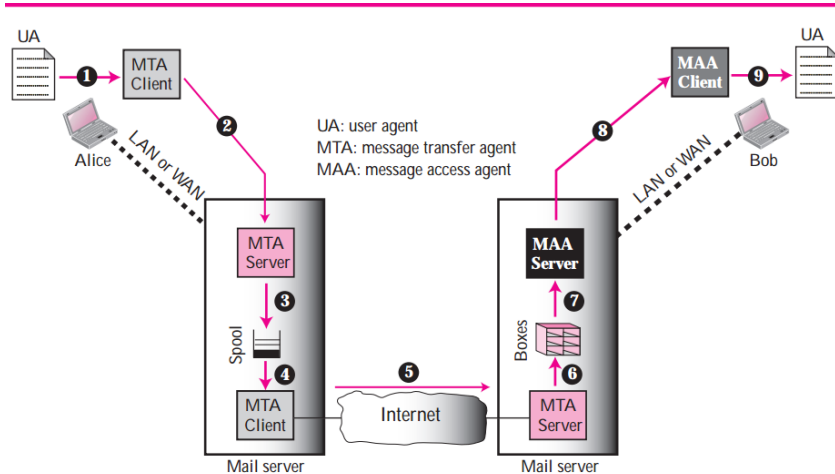
როგორც სურათზე ჩანს ბობი პირდაპირაა დაკავშირებული მის საფოსტო სერვერთან. ალისა კი LAN-ის ან WAN-ის მეშვეობით. ალისას სჭირდება მომხმარებლის აგენტი შეტყობინების მოსამზადებლად. შემდეგ მან უნდა გააგზავნოს შეტყობინება LAN-ის ან WAN-ის საშუალებით. ამისთვის საჭიროა ორი წყვილი შეტყობინების გაგზავნის აგენტი (კლიენტი და სერვერი). როცა ალისა აგზავნის შეტყობინებას, ის იძახებს მომხმარებლის აგენტს, რომელიც თავისმხრივ იძახებს MTA კლიენტს და ამყარებს კავშირს MTA სერვერთან. ალისას სისტემა ინახავს რიგში შეტყობინებებს. ამის შემდეგ იგი იყენებს MTA კლიენტს და უგზავნის შეტყობინებას ბობის სისტემას. სისტემა მოსულ შეტყობინებებს ინახავს ბობის საფოსტო ყუთში. ბობი იყენებს თავის მომხმარებლის აგენტს შეტყობინების მისაღებად და წასაკითხად.

როდესაც გამგზავნი დაკავშირებულია საფოსტო სერვერთან LAN-ის ან WAN-ის მეშვეობით, გვჭირდება ორი მომხმარებლის აგენტი (UA) და ორი წყვილი შეტყობინების გაგზავნის აგენტი (MTA) (კლიენტი და სერვერი).

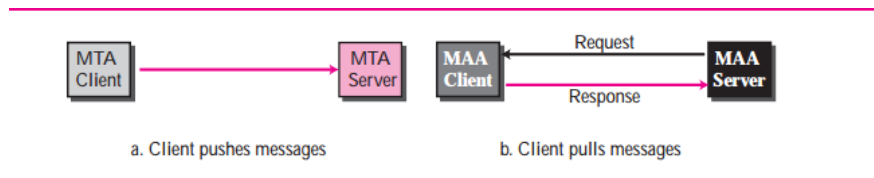


მეოთხე სცენარი

მეოთხე და ყველაზე გავრცელებული სცენარში ბობიც და ალისაც თავის საფოსტო სერვერთან დაკავშირებულია WAN-ის ან LAN-ის საშუალებით. მას შემდეგ რაც შეტყობინება გაეგზავნა ბობის საფოსტო სერვერს, ბობს სჭირდება მისი მიღება. ამისთვის საჭიროა შეტყობინების წვდომის აგენტი (MAA-კლიენტი და სერვერი). იმიტომ, რომ MTA კლიენტ-სერვერის პროგრამა არის გამგზავნი (push) პროგრამა: კლიენტი უგზავნის შეტყობინებას სერვერს. ბობს კი სჭირდება წამომღები (pull) პროგრამა. კლიენტს სჭირდება მიიღოს შეტყობინება სერვერიდან. განსხვავება push და pull პროგრამებს შორის ნაჩვენებია ნახ. 1-ზე.



ნახ. 1. განსხვავება push და pull პროგრამებს შორის



როდესაც გამგზავნიც და მიმღებიც დაკავშირებულია საფოსტო სერვერთან LAN-ის ან WAN-ის მეშვეობით, გვჭირდება ორი მომხმარებლის აგენტი (UA), ორი წყვილი შეტყობინების გაგზავნის აგენტი (MTA) (კლიენტი და სერვერი) და ერთი წყვილი MAA (კლიენტი და სერვერი). ეს არის დღეს ყველაზე გავრცელებული სიტუაცია.

1.2. მომხმარებლის აგენტი (UA)

ელექტრონული ფოსტის სისტემის უპირველეს კომპონენტს წარმოადგენს მომხმარებლის აგენტი (UA). ის გვთავაზობს სერვისებს, რომლებიც ამარტივებს შეტყობინების გაგზავნას და მიღებას.

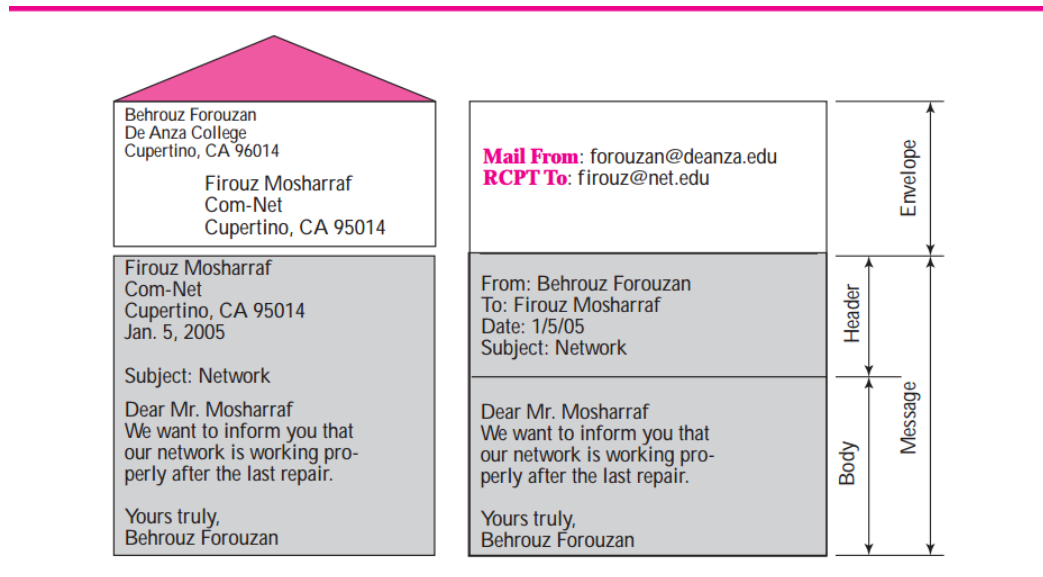
მომხმარებლის აგენტის ტიპები

არსებობს მომხმარებლის აგენტის ორი ტიპი: ბრძანებაზე ორიენტირებული და GUI-ზე დაფუძნებული.

ბრძანებაზე ორიენტირებული მომხმარებლის აგენტის მაგალითია: mail, pine და eml. ხოლო GUI-ზე ორიენტირებული მომხმარებლის აგენტის მაგალითია: Eudora, Outlook Netscape.

ელექტრონული ფოსტის გაგზავნა

ელექტრონული ფოსტის ფორმატი :



კონვერტი

კონვერტი შეიცავს გამგზავნის მისამართს, მიმღების მისამართს და სხვა ინფორმაციას.

შეტყობინება

შეტყობინება შეიცავს header-ს და body-ის. Header განსაზღვრავს გამგზავნის მისამართს, მიმღების მისამართს, შეტყობინების შინაარსს და სხვა ინფორმაციას, ხოლო body შეიცავს ინფორმაციას, რომელიც მიმღებმა უნდა წაიკითხოს.

ფოსტის მიღება

როცა მომხმარებელს მიღებული აქვს ფოსტა, მომხმარებლის აგენტი (UA) უგზავნის შეტყობინებას მას ამის შესახებ. ფოსტა შეიცავს გამომგზავნის მისამართს, შეტყობინების საგანს და გამოგზავნის დროს.

მისამართი

მისამართი შედგება ორი ნაწილისგან : local part და domain name, რომლებიც ერთმანეთისგან გამოყოფილია @ ნიშნით.



Local part

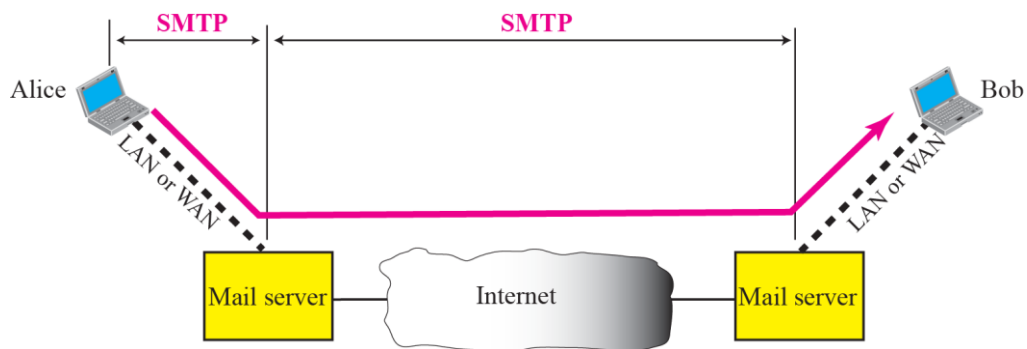
იგი განსაზღვრავს სპეციალური ფაილის სახელს, რომელსაც მომხმარებლის საფოსტო ყუთი ჰქვია. სადაც ინახება ამ მომხმარებლის მიერ მიღებული შეტყობინება.

Domain name

მისამართის მეორე ნაწილია დომენური სახელი. ორგანიზაცია როგორც წესი ირჩევს ერთ ან მეტ host-ს, რომ მიიღოს და გააგზავნოს ფოსტა. მათ ხშირად საფოსტო სერვერებს უწოდებენ.

1.3. შეტყობინების გაგზავნის აგენტი (MESSAGE TRANSFER AGENT)

ელ-ფოსტის გაგზავნა ხორციელდება შეტყობინების გაგზავნის აგენტის საშუალებით (MTAs). იმისათვის რომ სისტემამ გააგზავნოს შეტყობინება საჭიროა MTA კლიენტი, ხოლო შეტყობინების მისაღებად საჭიროა MTA სერვერი. პროტოკოლი, რომელიც განსაზღვრავს MTA კლიენტს და სერვერს ეწოდება Simple Mail Transfer Protocol (SMTP).



SMTP გამოიყენება ორჯერ, გამგზავნისა და გამგზავნის საფოსტო სერვერს შორის და ორ საფოსტო სერვერს შორის. როგორც სურათზე ჩანს საჭიროა სხვა პროტოკოლი საფოსტო სერვერსა და მიმღებს შორის.

ბრძანებები და პასუხები

SMTP იყენებს ბრძანებებს და პასუხებს MTA კლიენტსა და MTA სერვერს შორის შეტყობინებების გასაგზავნად.



ბრძანებები:

ბრძანებები იგზავნება კლიენტიდან სერვერზე. ბრძანების ფორმატი ნაჩვენებია სურათზე.

Keyword: argument(s)

Table 23.1 *Commands*

<i>Keyword</i>	<i>Argument(s)</i>	<i>Keyword</i>	<i>Argument(s)</i>
HELO	Sender's host name	NOOP	
MAIL FROM	Sender of the message	TURN	
RCPT TO	Intended recipient	EXPN	Mailing list
DATA	Body of the mail	HELP	Command name
QUIT		SEND FROM	Intended recipient
RSET		SMOL FROM	Intended recipient
VERFY	Name of recipient	SMAL FROM	Intended recipient

HELLO. ეს ბრძანება გამოიყენება კლიენტის იდენტიფიკაციისათვის. არგუმენტი არის კლიენტის დომენური სახელი.

MAIL FROM. ეს ბრძანება გამოიყენება კლიენტის მიერ შეტყობინების გამგზავნის იდენტიფიკაციისათვის. არგუმენტი არის გამგზავნის ელ-ფოსტა.

RCPT TO. ეს ბრძანება გამოიყენება კლიენტის მიერ შეტყობინების მიმღების იდენტიფიკაციისათვის. არგუმენტი არის მიმღების ელ-ფოსტა. ეს ბრძანება მეორდება თუ არის რამდენიმე მიმღები.

DATA. ეს ბრძანება გამოიყენება ფაქტობრივი შეტყობინების გასაგზავნად.

QUIT. ეს ბრძანება გამოიყენება შეტყობინების გასანადგურებლად.

RSET. ეს ბრძანება წყვეტს მიმდინარე საფოსტო ტრანზაქციას. ინფორმაცია გამგზავნის და მიმღების შესახებ იშლება. ხდება კავშირის გადატვირთვა.

VERFY. ეს ბრძანება გამოიყენება მიმღების მისამართის შემოწმებისთვის. გამგზავნს შეუძლია მოსთხოვოს მიმღებს დაადასტუროს, რომ მიმღების სახელი სწორია.

NOOP. ეს ბრძანება გამოიყენება კლიენტის მიერ, მიმღების სტატუსის შესამოწმებლად. საჭიროა პასუხი მიმღებისგან.

TURN. ეს ბრძანება საშუალებას აძლევს გამგზავნს და მიმღებს პოზიციები გაცვალონ. ანუ, გამგზავნი ხდება მიმღები და მიმღები გამგზავნი.

EXPN. ეს ბრძანება სთხოვს მიმღებს, რომ გაათავართოვოს საფოსტო ლისტი და დაუბრუნოს მიმღებების საფოსტო ყუთის მისამართები.

HELP. ეს ბრძანება სთხოვს მიმღებს ბრძანების შესახებ იმფორმაციის გაგზავნას.

SEND FROM. ეს ბრძანება განსაზღვრავს იმას, რომ ფოსტა მივიდეს მიმღების ტერმინალზე და არა საფოსტო ყუთში. თუ მიმღები არ არის შესული სისტემაში, ფოსტა უკან ბრუნდება. არგუმენტი არის გამგზავნის მისამართი.

SMOL FROM. ეს ბრძანება განსაზღვრავს, რომ ფოსტა მივიდეს მიმღების ტერმინალზე ან საფოსტო ყუთში. რაც ნიშნავს იმას, რომ თუ მიმღები შესულია სისტემაში, ფოსტა მიდის მხოლოდ ტერმინალზე. თუ მიმღები არ არის შესული სისტემაში, ფოსტა მიდის საფოსტო ყუთში. არგუმენტი არის გამგზავნის მისამართი.

SMOL FROM. ეს ბრძანება განსაზღვრავს, რომ ფოსტა მივიდეს მიმღების ტერმინალზე და საფოსტო ყუთში. თუ მიმღები შესულია სისტემაში, ფოსტა მიდის ტერმინალზე და საფოსტო ყუთში. თუ მიმღები გამოსულია სისტემიდან, ფოსტა მიდის მხოლოდ საფოსტო ყუთში. არგუმენტი არის გამგზავნის მისამართი.

პასუხები:

პასუხები ეგზავნება სერვერიდან კლიენტს. იგი არის სამნიშნა კოდი, რომელსაც დამატებით შეიძლება მოყვებოდეს ტექსტური ინფორმაცია.

Table 23.2 Responses

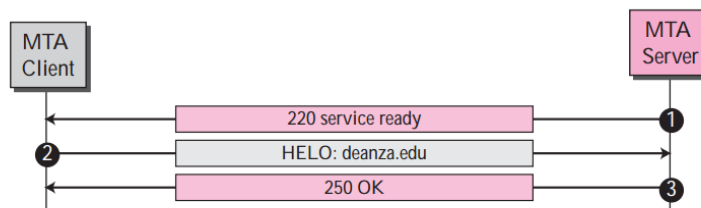
Code	Description
Positive Completion Reply	
211	System status or help reply
214	Help message
220	Service ready
221	Service closing transmission channel
250	Request command completed
251	User not local; the message will be forwarded
Positive Intermediate Reply	
354	Start mail input
Transient Negative Completion Reply	
421	Service not available
450	Mailbox not available
451	Command aborted: local error
452	Command aborted; insufficient storage
Permanent Negative Completion Reply	
500	Syntax error; unrecognized command
501	Syntax error in parameters or arguments
502	Command not implemented
503	Bad sequence of commands
504	Command temporarily not implemented
550	Command is not executed; mailbox unavailable
551	User not local
552	Requested action aborted; exceeded storage location
553	Requested action not taken; mailbox name not allowed
554	Transaction failed

1.4. ფოსტის გაგზავნის ფაზები

გამოყოფენ ფოსტის გაგზავნის სამ ფაზას: კავშირის დამყარება, ფოსტის გაგზავნა და კავშირის შეწყვეტა.

კავშირის დამყარება

იგი შედგება სამი საფეხურისგან, რომელიც ნაჩვენებია სურათზე.

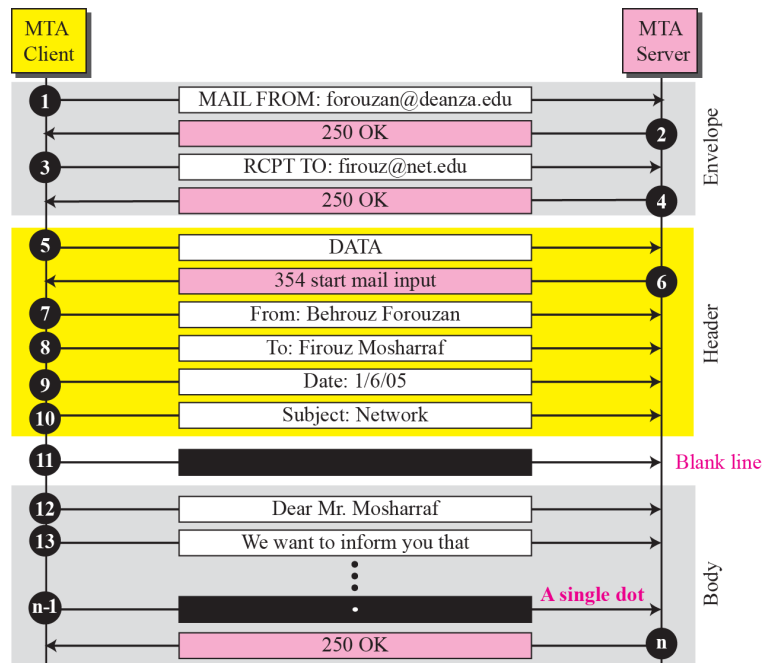


1. სერვერი აგზავნის 220 კოდს (სერვისი მზადაა) რომ უთხრას კლიენტს, რომ ის მზადაა ფოსტის მისაღებად. წინააღმდეგ შემთხვევაში ის აგზავნის 421 კოდს (სერვისი მიუწვდომელია).

- კლიენტი აგზავნის HELO შეტყობინებას იდენტიფიკაციისთვის, რისთვისაც იყენებს ღომეინ სახელს.
- სერვერი პასუხობს 250 კოდს (მოთხოვნილი ბრძანება დაშრულებულია) ან სხვა კოდს სიტუაციიდან გამომდინარე.

შეტყობინების გაგზავნა

SMTP კლიენტს და სერვერს შორის კავშირის დამყარების შემდეგ შესაძლებელია შეტყობინების გაგზავნა ერთ ან რამდენიმე მიმღებთან. ეს პროცესი მოიცავს 8 ფაზას, რომელიც ნაჩვენებია სურათზე. 3 და 4 საფეხური მეორდება თუ არის ერთზე მეტი მიმღები.

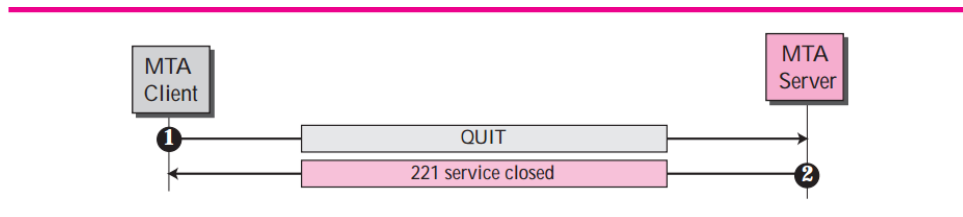


- კლიენტი აგზავნის MAIL FROM შეტყობინებას შეტყობინების გამგზავნის იდენტიფიკაციისათვის. იგი შეიცავს გამგზავნის საფოსტო მისამართს.
- სერვერი უბრუნებს 250 კოდს ან სხვა შესაბამის პასუხს.
- კლიენტი აგზავნის RCPT TO შეტყობინებას, რომელიც შეიცავს მიმღების საფოსტო მისამართს.
- სერვერი უბრუნებს 250 კოდს ან სხვა შესაბამის პასუხს.
- კლიენტი აგზავნის DATA შეტყობინებას შეტყობინების გაგზავნის ინიციალიზაციისათვის.

6. სერვერი პასუხობს 354 კოდით ან სხვა შესაბამისი პასუხით.
7. კლიენტი აგზავნის შეტყობინების შინაარსს თანმიმდევრული ხაზებით.
8. სერვერი უბრუნებს 250 კოდს ან სხვა შესაბამის პასუხს.

კავშირის შეწყვეტა

შეტყობინების წარმატებით გაგზავნის შემდეგ კლიენტი წყვეტს კავშირს. ეს პროცესი მოიცავს ორ საფეხურს, რომელიც ნაჩვენებია სურათზე.



1. კლიენტი აგზავნის QUIT ბრძანებას.
2. სერვერი პასუხობს 221 კოდით ან სხვა შესაბამისი კოდით.

კავშირის შეწყვეტის შემდეგ აუცილებელია TCP კავშირის დახურვა.

მაგალითი

განვიხილოთ მაგალითი, თუ როგორ შეგვიძლია გამოვიყენოთ SMTP ფოსტის გაგზავნისთვის და მიღებისთვის და გავაკეთოთ იმ ბრძანებებისა და პასუხების სიმულაცია, რომელიც ზემოთ აღვწერთ. ჩვენ ვიყენებთ TELNET-ს 25-ე პორტზე შესასვლელად (SMTP-ს პორტი). შემდეგ კი ვიყენებთ ბრძანებებს ფოსტის გასაგზავნად. ამ მაგალითში, forouzanb@adelphia.net თავის თავს უგზავნის ფოსტას. ქვემოთ ნაჩვენებია თუ როგორ უკავშირდება TELNET adelphia-ს საფოსტო სერვერს.

```

$ telnet mail.adelphia.net 25
Trying 68.168.78.100...
Connected to mail.adelphia.net (68.168.78.100).
  
```

დაკავშირების შემდეგ შეგვიძლია გავაგზავნოთ SMTP ბრძანებები და მივიღოთ პასუხები, როგორც ნაჩვენებია ქვევით. ბრძანებები ნაჩვენებია შავი ფერით და პასუხები ვარდისფერი ფერით. კომენტარები აღნიშნულია “=” ნიშნით. ისინი არ წარმოადგენენ საფოსტო პროცედურის ნაწილს.

```

===== Connection Establishment =====
220 mta13.adelphi a.net SMTP server ready Fri, 6 Aug 2004 . . .
HELO mail . adel phi a.net
250 mta13.adelphi a.net
===== Envelope =====
MAIL FROM: forouzanb@adel phi a.net
250 Sender <forouzanb@adel phi a.net> Ok
RCPT TO: forouzanb@adel phi a.net
250 Recipient <forouzanb@adel phi a.net> Ok
===== Header and Body =====
DATA
354 Ok Send data ending with <CRLF>. <CRLF>
From: Forouzan
TO: Forouzan

This is a test message
to show SMTP in action.
.

```

```

===== Connection Termination =====
250 Message received: adelphia.net@mail.adelphia.net
QUIT
221 mta13.adelphia.net SMTP server closing connection
Connection closed by foreign host.

```

1.5. შეტყობინების წვდომის აგენტი (MESSAGE ACCESS AGENT)

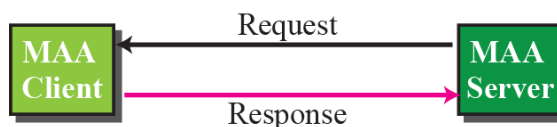
ფოსტის გაგზავნის პირველი და მეორე ეტაპები იყენებენ SMTP-ს (push protocol). თუმცა, SMTP არ გამოიყენება მესამე ეტაპზე, რადგან SMTP მხოლოდ აგზავნის შეტყობინებას კლიენტიდან სერვერზე, სხვა სიტყვებით რომ ვთქვათ, შეტყობინებების მიმართულება არის კლიენტიდან სერვერზე. მესამე ეტაპზე კი საჭიროა შეტყობინებების სერვერიდან მიღება (pull protocol); შეტყობინებების მიმართულება არის სერვერიდან კლიენტზე. მესამე ეტაპი იყენებს შეტყობინებაზე წვდომის აგენტს(MAA).

push protocol :



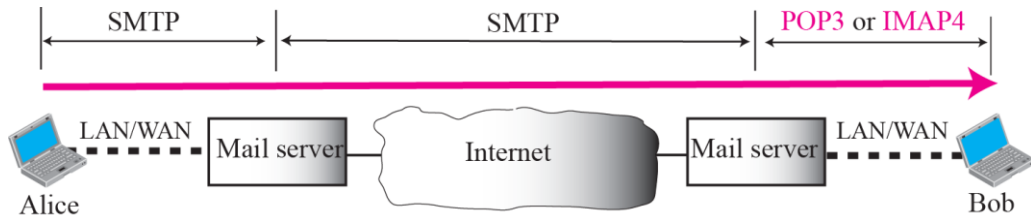
a. Client pushes messages

pull protocol :



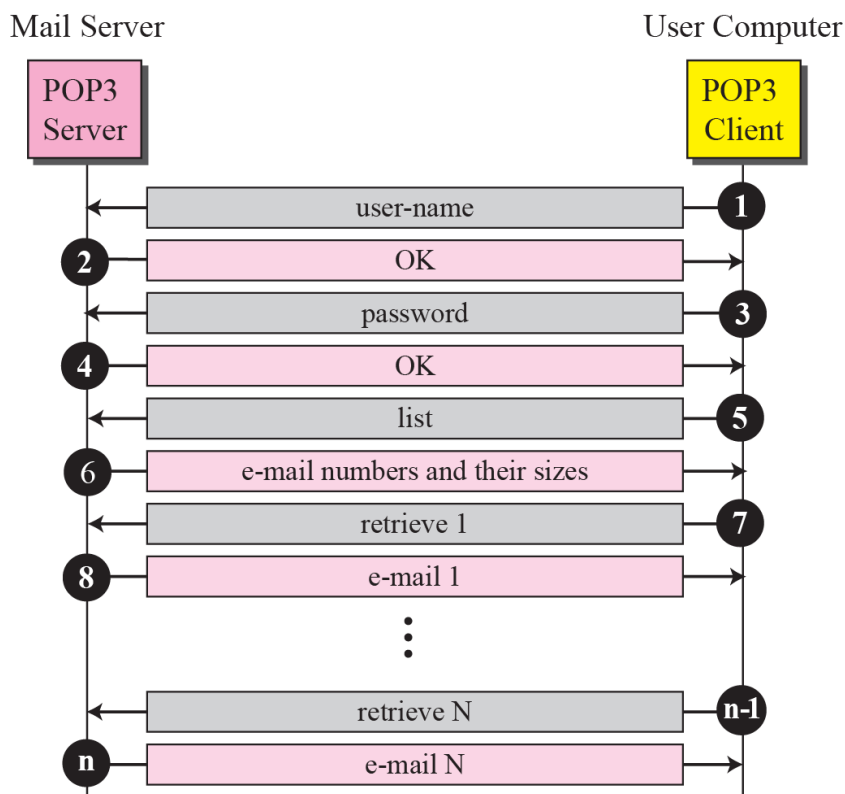
b. Client pulls messages

ამჟამად არსებობს ორი შეტყობინების წვდომის პროტოკოლი: Post Office Protocol version 3 (POP3) და Internet Mail Access Protocol version 4 (IMAP4).



Pop3

კლიენტის Pop3 პროგრამა ინსტალირდება მიმღების კომპიუტერზე, სერვერის Pop3 პროგრამა კი საფოსტო სერვერზე. ამის შემდეგ კლიენტი ხსნის კავშირს 110 TCP პორტზე. შემდეგ აგზავნის თავის username-ს და პაროლს საფოსტო ყუთზე წვდომისთვის. ყოველივე ეს ნაჩვენებია სურათზე.



Pop3-ს აქვს ორი რეჟიმი: წაშლის და შენარჩუნების. წაშლის რეჟიმში შეტყობინება წაშლილია საფოსტო ყუთიდან ყოველი შემდგომი წაკითხვის შემდეგ. შენარჩუნების რეჟიმში შეტყობინება რჩება საფოსტო ყუთში ყოველი შემდგომი წაკითხვის შემდეგ.

IMAP4

სხვა შეტყობინების წვდომის პროტოკოლია IMAP4. IMAP4 პროტოკოლი მსგავსია Pop3-ის, მაგრამ უფრო მეტი ფუნქციებით. IMAP4 უფრო ძლიერი და კომპლექსურია. ეს ფუნქციებია:

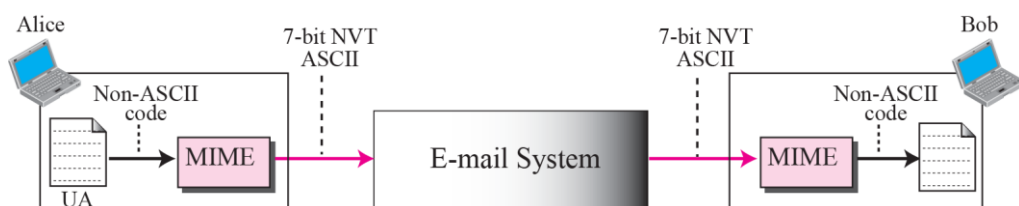
1. მომხმარებელს შეუძლია შეამოწმოს ფოსტის header ჩამოტვირთვამდე.
2. მომხმარებელს შეუძლია ფოსტის შინაარსის მოძებნა კონკრეტული ტექსტის მიხედვით ჩამოტვირთვამდე.
3. მომხმარებელს შეუძლია ნაწილობრივ ჩამოტვირთოს შეტყობინება. ეს მოსახერხებელია იმ შემთხვევაში თუ სიჩქარე შეზღუდულია და შეტყობინება ითხოვს მაღალ სიჩქარეს.
4. მომხმარებელს შეუძლია შექმნას, წაშალოს ან სახელი შეუცვალოს საფოსტო ყუთებს საფოსტო სერვერზე.
5. მომხმარებელს შეუძლია საქალაქში შექმნას საფოსტო ყუთების იერარქია.

1.6. MIME

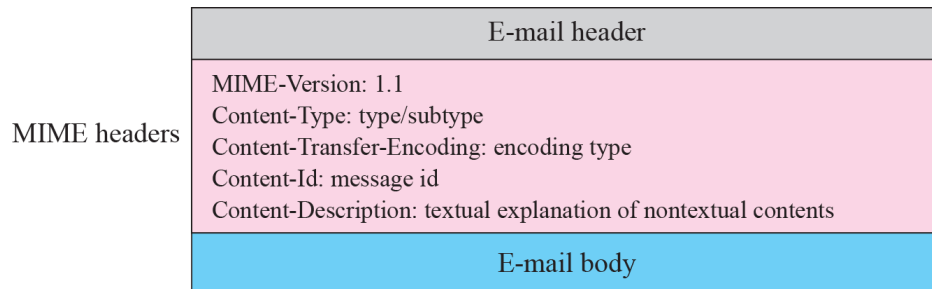
ელექტრონულ ფოსტას აქვს მარტივი სტრუქტურა. მას შეუძლია შეტყობინების გაგზავნა მხოლოდ NVT 7-ბიტანი ASCII ფორმატში. Multipurpose Internet Mail Extensions (MIME) არის დამატებითი პროტოკოლი, რომელიც საშუალებას გვაძლევს ელ-ფოსტის საშუალებით გავაგზავნოთ non-ASCII მონაცემები. MIME გარდაქმნის non-ASCII მონაცემებს გამგზავნის მხარეს NVT ASCII მონაცემებად და აწვდის მას MTA კლიენტს ინტერნეტით გასაგზავნად. შეტყობინება მიმღების მხარეს გარდაიქმნება non-ASCII ფორმატში.

MIME Headers

MIME განსაზღვრავს ხუთ header-ს, რომლებიც შეიძლება დაემატოს ორიგინალ ელ-ფოსტის header-ების განყოფილებას. ესენია :



1. MIME-Version
2. Content-Type
3. Content-Transfer-Encoding
4. Content-Id
5. Content-Description



MIME-Version

ეს header განსაზღვრავს MIME-ის ვერსიას.

Content-Type

იგი განსაზღვრავს შეტყობინების ტანში გამოყენებული მონაცემების ტიპს. გამოიყოფა შვიდი ტიპი, რომლებიც ნაჩვენებია სურათზე.

Type	Subtype	Description
Text	Plain	Unformatted
	HTML	HTML format (see Appendix E)
Multipart	Mixed	Body contains ordered parts of different data types
	Parallel	Same as above, but no order
	Digest	Similar to Mixed, but the default is message/RFC822
	Alternative	Parts are different versions of the same message
Message	RFC822	Body is an encapsulated message
	Partial	Body is a fragment of a bigger message
	External-Body	Body is a reference to another message
Image	JPEG	Image is in JPEG format
	GIF	Image is in GIF format
Video	MPEG	Video is in MPEG format
Audio	Basic	Single channel encoding of voice at 8 KHz
Application	PostScript	Adobe PostScript
	Octet-stream	General binary data (eight-bit bytes)

1. **Text.** ორიგინალი შეტყობინება არის 7 ბიტის ASCII ფორმატში და არ არის საჭირო გარადქმნა MIME-ის მიერ. გვაქვს ორი ქვეტიპი, რომლებიც ამჟამად გამოიყენება: plain და HTML.

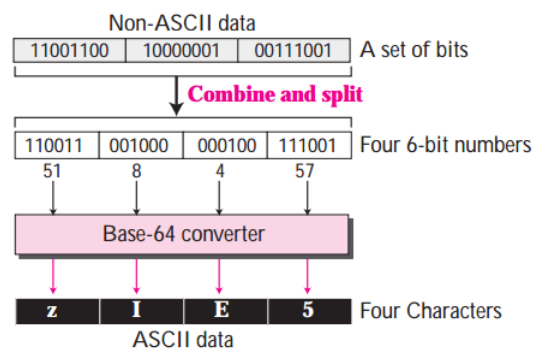
2. **Multipart.** ტანი შეიცავს მრავალ დამოუკიდებელ ნაწილს. აუცილებელია საზღვრების განსაზღვრა მრავალნაწილიანი სათაურისთვის. პარამეტრი, რომელიც გამოიყენება ამისთვის ეწოდება ტოკენი.
3. **Message.** შეტყობინებაში ტანი არის ან მთლიანი ფოსტა, ან ფოსტის ნაწილი, ან კიდევ ფოსტაზე მიმთითებელი. ამჟამად გამოიყენება სამი ქვეტიპი: RFC822, ნაწილობრივი და გარე ნაწილი.
4. **Image.** ორიგინალი შეტყობინება არის სტაციონარული სურათი, რომელიც მიუთითებს იმაზე, რომ არ არსებობს ანიმაცია. ამჟამად გამოიყენება ორი ქვეტიპი: Joint Photographic Experts Group (JPEG) და Graphics Interchange Format (GIF).
5. **Video.** ორიგინალი შეტყობინება არის ანიმაცია. არსებობს მხოლოდ ერთი ქვეტიპი: Moving Picture Experts Group (MPEG). თუ ანიმაციური სურათი შეიცავს ხმას, იგი უნდა გაიგზავნოს ცალკე, როგორც audio.
6. **Audio.** ორიგინალი შეტყობინება არის ხმა. არსებობს მხოლოდ ერთი ქვეტიპი: basic.
7. **Application.** აქ გამოიყოფა ორი ქვეტიპი: PostScript და octet-stream. PostScript გამოიყენება, მაშინ, როდესაც მონაცემები არის Adobe PostScript ფორმატში. octet-stream გამოიყენება იმ შემთხვევაში, როდესაც მონაცემები უნდა იყოს 8 ბიტის ფორმატში.

Content-Transfer-Encoding

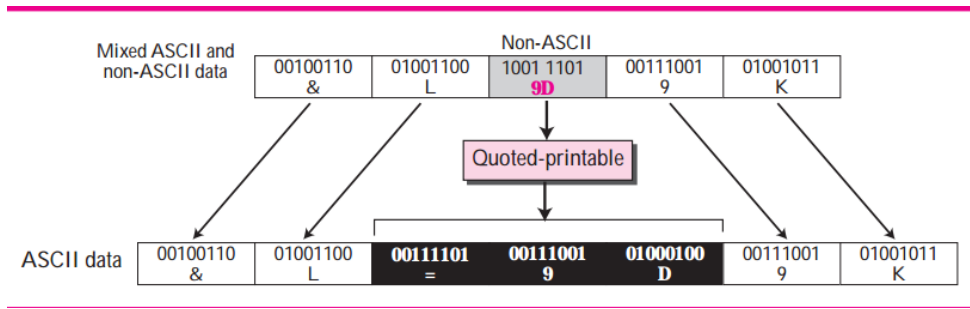
იგი განსაზღვრავს მეთოდებს, რომლებიც გამოიყენება გასაგზავნი შეტყობინების 0 და 1 -ზე კოდირებისთვის. გამოიყოფა კოდირების ხუთი ტიპი, რომელიც ნაჩვენებია სურათზე.

Type	Description
7bit	NVT ASCII characters and short lines
8bit	Non-ASCII characters and short lines
Binary	Non-ASCII characters with unlimited-length lines
Base64	6-bit blocks of data are encoded into 8-bit ASCII characters
Quoted-printable	Non-ASCII characters are encoded as an equal sign plus an ASCII code

1. **7bit.** ეს არის 7 ბიტისანი NVT ASCII კოდირება. ერთი ხაზის სიგრძე არ უნდა აღემატებოდეს 1000 სიმბოლოს.
2. **8bit.** ეს არის 8 ბიტისანი კოდირება. ამ შემთხვევაში შეგვიძლია Non-ASCII სიმბოლოების გაგზავნა. აქაც ხაზის სიგრძე არ უნდა აღემატებოდეს 1000 სიმბოლოს. MIME-მა აქ არ უნდა განახორციელოს არანაირი კოდირება. SMTP პროტოკოლს უნდა შეეძლოს 8 ბიტისანი Non-ASCII სიმბოლოების გარდაქმნა.
3. **Binary.** ეს არის 8 ბიტისანი კოდირება. ამ შემთხვევაში შეგვიძლია Non-ASCII სიმბოლოების გაგზავნა და ხაზის სიგრძე არ უნდა აღემატებოდეს 1000 სიმბოლოს. MIME-მა აქ არ უნდა განახორციელოს არანაირი კოდირება. SMTP პროტოკოლს უნდა შეეძლოს 8 ბიტისანი Non-ASCII სიმბოლოების გარდაქმნა.
4. **Base64.** Base64 გარდაქმნის მონაცემებს ბეჭდვად სიმბოლოებად, რომლებიც შემდგომ შეგვიძლია გაიგზავნოს, როგორც ASCII სიმბოლოები ან ნებისმიერი სხვა ტიპის სიმბოლოები, რომელსაც აქვს ფოსტის გაგზავნის მექანიზმის მხარდაჭერა. Base64 ჰყოფს ორობით მონაცემებს 24 ბიტის ბლოკებად. თითოეული ბლოკი იყოფა 4 სექციად. (1 სექცია 6 ბიტი). ყოველივე ეს ნაჩვენებია ნახაზზე.



5. **Quoted-printable.** თუ მონაცემები შეიცავს უმეტესად ASCII სიმბოლოებს და ნაწილობრივს Non-ASCII სიმბოლოებს, შეგვიძლია გამოვიყენოთ Quoted-printable კოდირება. თუ სიმბოლო არის ASCII, ის გაიგზავნება ისე როგორც არის. წინააღმდეგ შემთხვევაში იგზავნება სამ სიმბოლოდ. პირველი სიმბოლოა (=). შემდეგი ორი სიმბოლო წარმოდგენილია თექვსმეტობით ფორმატში. ყოველივე ეს ნაჩვენებია ნახაზზე.



Content-Id

ეს header ცალსახად განსაზღვრავს მთელ შეტყობინებას მრავალი შეტყობინების გარემოში.

Content-Description

იგი განსაზღვრავს შეტყობინების ტანი სურათია, აუდიო თუ ვიდეო.

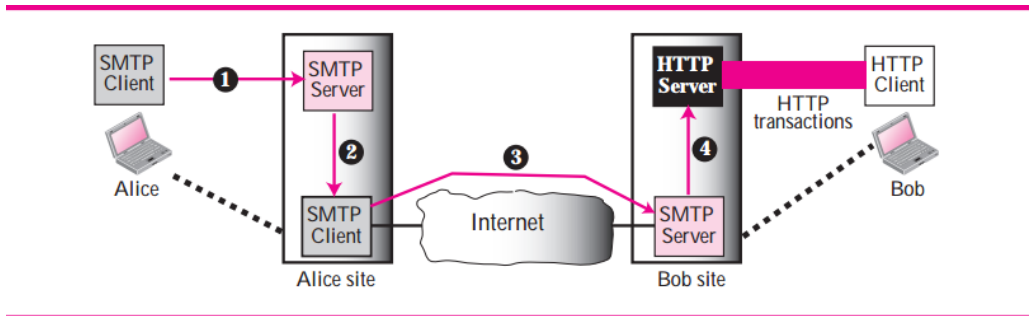
§2. web- შეტყობინების სისტემა

ელ-ფოსტა ისეთი გავრცელებული აპლიკაციაა, რომ ზოგიერთი web-საიტი სთავაზობს ამ სერვისს მათ, ვისაც წვდომა აქვს ამ საიტზე. სამი ძირითადი საიტია Hotmail, Yahoo და Google. იღეა ძალიან მარტივია. განვიხილოთ ორი შემთხვევა:

web- შეტყობინების სისტემა, შემთხვევა 1

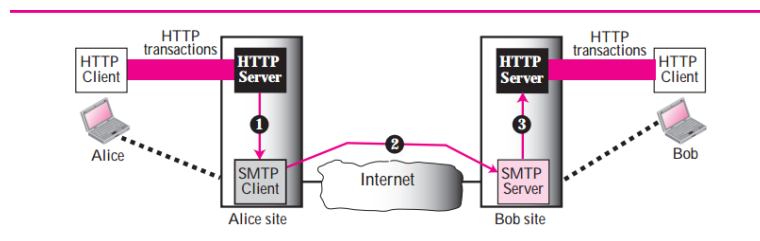
პირველ შემთხვევაში ალისა, გამგზავნი, იყენებს ჩვეულებრივ საფოსტო სერვერს. ბობს, მიმღებს, აქვს ექაუნთი Web-Based სერვერზე. ალისა შეტყობინებას თავის საფოსტო სერვერზე აგზავნის SMTP-ს საშუალებით. ხოლო შეტყობინების გაგზავნა მიმღები სერვერიდან ბობის ბრაუზერამდე ხდება HTTP-ის გავლით.

ყოველივე ეს ნაჩვენებია სურათზე.



web- შეტყობინების სისტემა, შემთხვევა 2

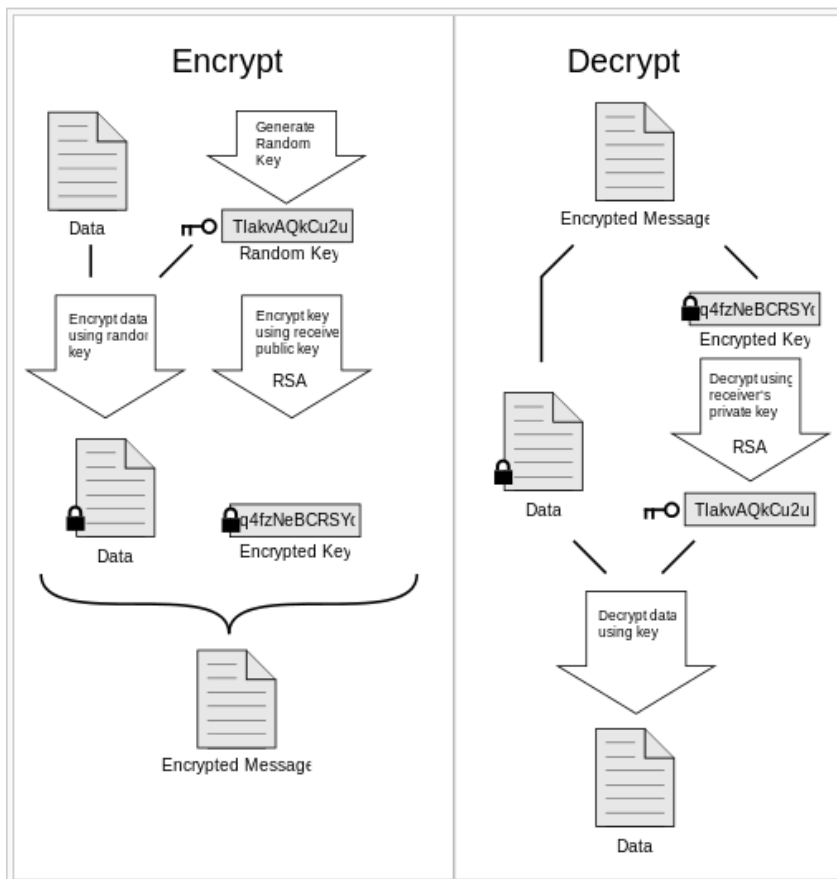
მეორე შემთხვევაში ორივე, ალისაც და ბობიც იყენებენ ვებ სერვერებს. (არაა აუცილებელი ერთიდაიგივე სერვერის გამოყენება). ალისა უგზავნის შეტყობინებას ვებ სერვერს HTTP ტრანზაქციების გამოყენებით. იგი აგზავნის HTTP მოთხოვნას ბობის საფოსტო ყუთის სახელისა და მისამართის გამოყენებით, როგორც URL. შემდეგ ვებ სერვერი გადასცემს შეტყობინებას SMTP კლიენტს და უგზავნის ბობის SMTP სერვერს SMTP პროტოკოლის გამოყენებით. ბობი იღებს შეტყობინებას HTTP ტრანზაქციების გამოყენებით. ეს სიტუაცია ნაჩვენებია სურათზე.



§3. ელ-ფოსტის უსაფრთხოება

პროტოკოლი რომელიც ზემოთ განვიხილეთ, არ გვანდის უსაფრთხოების არანაირ დებულებებს. თუმცა, ელ-ფოსტის გაცვლა შეიძლება დავიცვათ ორი პროტოკოლის გამოყენებით, რომლებიც შექმნილია ელ-ფოსტის სისტემისთვის. ესენია Pretty Good Privacy (PGP) და Secure MIME (SMIME).

Pretty Good Privacy [6] არის მონაცემთა შიფრაციისა და დეშიფრაციის კომპიუტერული პროგრამა, რომელიც უზრუნველყოფს კრიპტოგრაფიულ კონფიდენციალურობას და ავტორიზაციას მონაცემთა გადაცემისთვის. PGP ხშირად გამოიყენება ხელმოწერისთვის, ტექსტების, ელექტრონული ფოსტების, ფაილების, კატალოგების შიფრაციის და დეშიფრაციისთვის ელ-ფოსტის უსაფრთხოების გაზრდის მიზნით. იგი შექმნა ფილ ზიმერმანმა 1991 წელს. PGP და მსგავსი პროგრამები იყენებენ OpenPGP სტანდარტს შიფრაციისა და დეშიფრაციისთვის. სურათზე ნაჩვენებია PGP-ს მუშაობის პრინციპი.



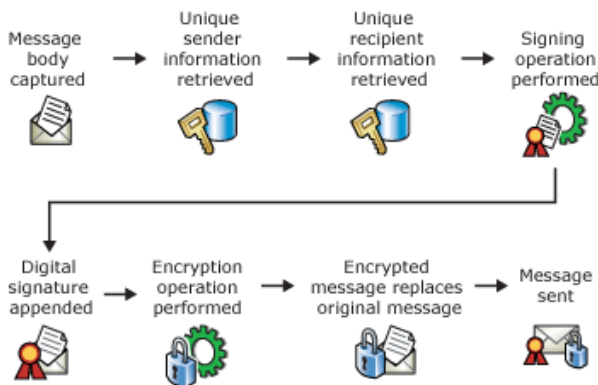
Secure MIME [7] არის ელექტრონული ფოსტის გაგზავნის უსაფრთხო მეთოდი, რომელიც იყენებს RSA (Rivest-Shamir-Adleman) შიფრაციის სისტემას. S/MIME არ არის შეზღუდული, რაც ნიშნავს იმას, რომ მისი გამოყენება შეგვიძლია ნებისმიერ სატრანსპორტო მექანიზმში, რომელიც აგზავნის MIME მონაცემებს, ისეთებს როგორცაა HTTP.

იგი გვთავაზობს შემდეგ მომსახურებებს :

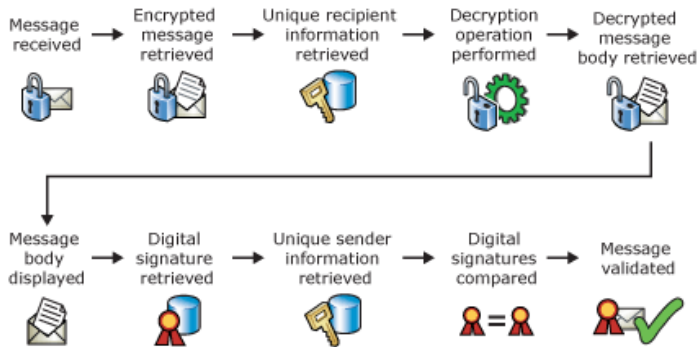
1. Authentication.
2. Message Integrity.
3. Non-repudiation of origin.
4. Privacy and data security.

ქვემოთ ნაჩვენებია S/MIME-ს მუშაობის პრინციპი.

ამ სურათზე ნაჩვენებია ელექტრონული ფოსტის ციფრული ხელმოწერა და შიფრაცია :



ამ სურათზე ნაჩვენებია ელექტრონული ფოსტის დეშიფრაცია და ციფრული ხელმოწერის დადასტურება :



ამჟამად არსებობს S/MIME-ის სამი ვერსია. ესენია :

- S/MIME ვერსია 1 - რომელიც ოფიციალურად გამოქვეყნდა 1995 წელს.
- S/MIME ვერსია 2 - რომელშიც განსაზღვრული იყო RFC-ს წყვილი ინფორმაციული დოკუმენტი : RFC 2311 და RFC 2312, რომელიც ოფიციალურად გამოქვეყნდა 1998 წლის მარტში.
- მუშაობა გაგრძელდა IETF-ში და შედეგად S/MIME ვერსია 3-ში განსაზღვრულია RFCs 2630-დან 2634-მდე. აღნიშნული ვერსია ოფიციალურად გამოქვეყნდა 1999 წლის ივნისში.

§4. ავტომატური შეტყობინების სისტემა - პროგრამული რეალიზაცია

როგორც ზემოთ აღწერილი ანალიზიდან ჩანს ელექტრონული ფოსტა წარმოადგენს ყველაზე გავრცელებულ და მოსახერხებელ ინტერნეტ სერვისს.

ასეთი ტიპის ავტომატური შეტყობინების სისტემა ხშირ შემთხვევაში წარმოადგენს კონკრეტულ web-გვედში ჩაშენებულ ძრავს, რომლის რეალიზაცია ხდება: .Net, Php, Python და სხვა პროგრამული ენების გამოყენებით.

მიუხედავად ასეთი ტიპის ძრავის პოპულარობისა ინტერნეტ სივრცეში ნაკლებად მოიძებნება მარტივი გამოყენების (ჩაშენების თვალსაზრისით) ღია წვდომის (Open Source) პროგრამული ძრავები.

სამაგისტრო ნაშრომის ფარგლებში შეირჩა ელექტრონული ფოსტის გაგზავნის სისტემა. შეიქმნა შესაბამისი ალგორითმი ამ სცენარის რეალიზაციისათვის, და მოხდა მისი პროგრამული რეალიზაცია (დაპროგრამების ენა - C#-ზე). რომელიც ნაჩვენებია ქვემოთ მოცემულ სურათზე.

```
using System;
using System.Net;           ბიბლიოთეკების ჩამონათვალი
using System.Net.Mail;
using System.Web.UI;

private string host = "smtp.gmail.com"; - SMTP პროტოკოლი
private int _port = 587;           - SMTP პროტოკოლის პორტის ნომერი

private readonly NetworkCredential
_credentials = new
NetworkCredential("myMail@gmail.com", "Password"); SMTP კლიენტის ლოგინი და პაროლი

var sendFrom = new MailAddress("SenderMail@gmail.com"); - გამგზავნის საფოსტო მისამართი
var sendTo = new MailAddress(tbxMailTo.Text);           - მიმღების საფოსტო მისამართი

var myMessage = new MailMessage(sendFrom, sendTo)
{
    Subject = "This is an email subject",           - შეტყობინება
    Body = "this content is in the body"
};
```

```
var emailClient = new SmtpClient(host, _port) - SMTP კლიენტი
{
    EnableSsl = true, - Ssl-ის ჩართვა
    Credentials = _credentials
};
emailClient.Send(myMessage); - შეტყობინების გაგზავნა

litStatus.Text = "Message Sent"; - შეტყობინების გაგზავნის სტატუსი
```

დ ა ს კ ვ ნ ა

როგორც აღვნიშნეთ, ელექტრონული ფოსტა არის ყველაზე გავრცელებული აპლიკაცია ინტერნეტში. მისი არქიტექტურა შედგება რამდენიმე კომპონენტისაგან: მომხმარებლის აგენტი (UA), შეტყობინების გაგზავნის აგენტი (MTA) და შეტყობინების წვდომის აგენტი (MAA).

მომხმარებლის აგენტი (UA) ამზადებს შეტყობინებას, ქმნის კონვერტს და ღებს შეტყობინებას კონვერტში. ფოსტის მისამართი შედგება ორი ნაწილისგან : **local part** და **domain name**. ფორმატი კი შემდეგნაირია: **localpart@domainname**.

შეტყობინების გაგზავნის აგენტი (MTA) აგზავნის ფოსტას ინტერნეტის, LAN-ის ან WAN-ის გავლით. პროტოკოლი, რომელიც განსაზღვრავს MTA-ს ენოდება Simple Mail Transfer Protocol (SMTP). იგი იყენებს ბრძანებებს და პასუხებს MTA კლიენტსა და MTA სერვერს შორის. ფოსტის გაგზავნის საფეხურებია: **კავშირის დამყარება, ფოსტის გაგზავნა და კავშირის შეწყვეტა**.

შეტყობინების წვდომის აგენტი (MAA)-ს იმპლემენტაციისთვის გამოიყენება ორი პროტოკოლი: **Post Office Protocol version 3 (POP3)** და **Internet Mail Access Protocol version 4 (IMAP4)**. ეს პროტოკოლები გამოიყენება სერვერიდან შეტყობინებების მისაღებად.

Multipurpose Internet Mail Extension (MIME)-ის საშუალებით შეაძლებელია მულტიმედია შეტყობინებების გაგზავნა. MIME ცვლის მულტიმედია სიმბოლოებს ASCII სიმბოლოებად.

დიდი პოპულარობით სარგებლობს Web-based ელექტრონული ფოსტები. ამ სისტემებში მონაცემების გადაცემა ხდება ნაწილობრივ SMTP პროტოკოლით და ნაწილობრივ HTTP პროტოკოლით.

ელექტრონული ფოსტის დაცვა შესაძლებელია ორი ტექნოლოგიით: **Pretty Good Privacy (PGP)** და **SMIME (Secure MIME)**-ით.

სამაგისტრო ნაშრომის ფარგლებში შეირჩა ელექტრონული ფოსტის გაგზავნის სისტემა. შეიქმნა შესაბამისი ალგორითმი ამ სცენარის რეალიზაციისათვის, და მოხდა მისი პროგრამული რეალიზაცია. აღნიშნული ძრავი საშუალებას იძლევა გავაგზავნოთ

ელექტრონული ფოსტა. ძრავი არის Open Source და ნებისმიერს შეუძლია მისი გამოყენება, გადანერა, ცვლილება და გავრცელება.

გ ა მ ო ყ ე ნ ე ბ ე უ ლ ი ლ ი ტ ე რ ა ტ უ რ ა

1. <http://www.mikeownage.com/mike/ebooks/TDC463.pdf>
2. <https://en.wikipedia.org/wiki/Email>
3. <http://www.workplace-communication.com/advantages-disadvantages-email.html>
4. <http://smallbusiness.chron.com/use-email-business-communication-118.html>
5. <http://emailstatcenter.com/AutomatedMessages.html>
6. https://en.wikipedia.org/wiki/Pretty_Good_Privacy
7. [https://technet.microsoft.com/en-us/library/aa995740\(v=exchg.65\).aspx](https://technet.microsoft.com/en-us/library/aa995740(v=exchg.65).aspx)
8. <http://www.atuljain7.com/sms-marketing-%E2%80%93-advantages-and-disadvantages>
9. <http://txt180.com/blog/tips/sms-and-mms/>

ავტომატური შეტყობინების სისტემა (პროგრამული კოდი)

```
using System;
using System.Net;
using System.Net.Mail;
using System.Web.UI;
namespace SMTP
{
    public partial class SmtplibWebForm : Page
    {
        private string host = "smtp.gmail.com";
        private int _port = 587;
        private readonly NetworkCredential _credentials =
            new NetworkCredential("myMail@gmail.com", "Password");
        protected void Page_Load(object sender, EventArgs e)
        {
        }
        protected void btnSendMail_OnClick(object sender, EventArgs e)
        {
            try
            {
                var sendFrom = new MailAddress("SenderMail@gmail.com");
                var sendTo = new MailAddress(tbxMailTo.Text);
                var myMessage = new MailMessage(sendFrom, sendTo)
                {
                    Subject = "This is an email subject",
                    Body = "this content is in the body"
                };
                var emailClient = new SmtplibClient(host, _port)
                {
                    EnableSsl = true,
```

```
        Credentials = _credentials
    };
    emailClient.Send(myMessage);
    litStatus.Text = "Message Sent";
}
catch (Exception ex)
{
    litStatus.Text = ex.ToString();
}
}
}
```