

Email Essentials

A printable lesson plan for teaching email as both an everyday tool and a working Internet system. This updated version deliberately builds on students' knowledge of the Internet and IP addresses.

Core question: If I know your email address, how does my computer discover the IP address of the server that should receive the message?

Duration	100-120 minutes
Prior knowledge	Internet basics and IP addresses
Format	Discussion + live terminal demo + hands-on email activity
Big ideas	DNS, MX records, IP addresses, TCP ports, protocols, client-server systems, store-and-forward, security

Learning outcomes

- Explain the mailbox and domain parts of an email address.
- Use To, CC, BCC, Reply, Reply All, subjects and attachments appropriately.
- Trace delivery from an email domain through DNS/MX to a mail server hostname and IP address.
- Explain why an IP address is not enough by itself, and why ports identify services.
- Describe the roles of TCP/IP and SMTP in moving an email between servers.
- Explain why the recipient computer does not need to be online when a message is sent.
- Recognize basic phishing/spoofing warning signs and explain SPF at a conceptual level.

Teaching principle

Use email as a case study for the Internet concepts students already know. The target mental model is not simply "SMTP sends mail"; it is **address -> domain -> DNS -> MX -> server name -> IP -> TCP/port -> SMTP -> mailbox**.

1. Lesson flow

Time	Topic	Teacher action / learning goal
0-10	Hook	Send an email live. Ask: "Does my laptop contact the recipient laptop directly?"
10-22	Everyday email	Address anatomy, subject, To/CC/BCC, Reply/Reply All, attachments and threads.
22-35	The network problem	We have a domain, but computers communicate using IP addresses. How do we find the right server?
35-55	DNS + MX + IP	Resolve recipient domain -> MX hostname -> A/AAAA -> IP. Live dig/nslookup demo.
55-70	IP + port + TCP	IP identifies the host; a port identifies the service. Build the TCP connection to SMTP.
70-82	SMTP + store/forward	Protocol conversation, server-to-server delivery, mailbox storage.
82-94	Headers + security	Inspect Received headers; phishing, spoofing, SPF/DKIM/DMARC at high level.
94-112	Hands-on	Compose/send, CC/BCC, reply, forward, inspect original/header, trace one message.
112-120	Recap	Students redraw the full journey from memory.

2. Opening hook

Open a mail client and compose a tiny message. Before pressing Send, ask:

"When I click Send, does my computer directly contact the recipient's computer?"

Collect guesses. Because they already know IP addresses, follow with: **"If computers need IP addresses, where is the IP address inside ali@cs17.org?"** Do not answer yet.

```
My laptop
|
v
Email app
|
?
?
?
v
Ali's inbox
```

3. Fast practical foundation

Email address: ali@cs17.org -> mailbox = ali; domain = cs17.org. The domain tells us which mail system is responsible for the address.

4. To, CC, BCC, reply and threads

Field	Use it when...	Key idea
To	Someone is expected to act or respond.	Primary recipient.
CC	Someone should be informed but is not the main actor.	Visible to all recipients.
BCC	Recipients should not see one another.	Useful for privacy in mass mail.
Reply	Respond only to the sender.	Avoid unnecessary group noise.
Reply All	Everyone genuinely needs the response.	Use deliberately.

Mini discussion

Show a bad subject such as Hi and a useful one such as CS17 Boolean Algebra Assignment - Ahmed. Ask which is easier to find a week later.

5. The central network problem

Now return to the technical mystery. Suppose sara@gmail.com sends to ali@cs17.org. The sender can extract the domain cs17.org, but Internet routing ultimately needs an IP address.

Question for the class: “How do we turn cs17.org into the IP address of the machine that receives email for it?”

Do not jump straight to an A record. Email uses a special DNS record first: **MX (Mail Exchange)**.

```
ali@cs17.org
  |
  v
Domain: cs17.org
  |
  v
DNS query: MX cs17.org
  |
  v
Mail server hostname
  |
  v
DNS A / AAAA lookup
  |
  v
IP address
```

Teacher checkpoint

Students should be able to say: “The email address gives us a domain. DNS MX tells us which hostname receives mail for that domain. DNS then resolves that hostname to an IP address.”

6. DNS -> MX -> hostname -> IP

Run a live terminal demo. Choose a public domain and ask students to predict each step before you run it.

```
dig MX gmail.com
```

```
# or on systems without dig:
nslookup -type=mx gmail.com
```

Pick one returned mail-server hostname, then resolve it:

```
dig A <mail-server-hostname>
dig AAAA <mail-server-hostname>
```

Important: an MX record usually points to a *hostname*, not directly to an IP address. That hostname is then resolved using A (IPv4) and/or AAAA (IPv6).

7. Why an IP address is still not enough

A single computer/server can run many network services at once. The IP address identifies the host; the **port number** identifies which service on that host should receive the connection.

```
203.0.113.25
|
+-- port 22    -> SSH
+-- port 80    -> HTTP
+-- port 443   -> HTTPS
+-- port 25    -> SMTP (server-to-server mail)
+-- port 587   -> mail submission
```

Mental model: IP address = which machine? | Port = which service on that machine?

Do not make the class memorize ports. The purpose is to show why knowing an IP address alone does not completely identify the network service.

8. Where TCP fits

At a simplified level, the sending mail server establishes a **TCP connection** to the receiving mail server, usually on SMTP port 25 for server-to-server transfer. SMTP then defines what the two applications say over that connection.

```
EMAIL
  v
SMTP    <- application protocol / rules
  v
TCP     <- reliable byte stream between endpoints
  v
IP      <- addressing and routing across networks
  v
Network
```

This is enough layering for this lesson; there is no need to detour into all seven OSI layers.

9. Put the entire journey together

```
Sara writes: ali@cs17.org
|
v
[ Email client / webmail ]
|
v
[ Sender mail server ]
|
| 1. Extract domain: cs17.org
| 2. DNS MX lookup
| 3. MX -> mail-server hostname
| 4. A/AAAA -> IP address
| 5. Open TCP connection to SMTP service
v
[ Recipient mail server ]
|
| SMTP transfers message
v
[ Ali's mailbox ]
|
| Webmail / IMAP
v
[ Ali's phone or laptop ]
```

Ask: “**What happens if Ali's laptop is switched off?**” The message can still arrive because the recipient mail server stores it in Ali's mailbox. This is a classic **store-and-forward** system.

10. SMTP is a protocol, not magic

Explain that a protocol is an agreed set of rules for communication. SMTP servers exchange structured commands. A simplified conversation might look like this:

```
S: 220 mail.example.com ready
C: EHLO sender.example
C: MAIL FROM:<sara@example.com>
C: RCPT TO:<ali@cs17.org>
C: DATA
C: ...message contents...
C: .
C: QUIT
```

The exact details are less important than the idea: **SMTP defines the application-level conversation; TCP carries it; IP gets packets across networks.**

11. SMTP vs IMAP vs POP3

Protocol	Main role	Simple mental model
SMTP	Sending/relaying mail.	“Move this message to another mail server.”
IMAP	Access and synchronize mailbox state.	“Keep my phone and laptop in sync with the server mailbox.”
POP3	Traditionally download messages from a mailbox.	“Fetch mail down to this device.”

12. Email headers: turn students into detectives

Open a real message and use the mail client's "Show original" / "View source" option. Ask students to find familiar pieces and anything that looks like a server or IP address.

```
From: ...
To: ...
Date: ...
Message-ID: ...
Received: from ... by ...
Received: from ... by ...
Content-Type: ...
```

Focus on Received:. Mail infrastructure adds these headers as the message moves through servers. Ask: **"How many hops can you identify?"** This connects visible metadata to the server path they just learned.

13. MIME and attachments

Email began as text-oriented communication. **MIME** lets a message describe different content types and attachments. This is a useful bridge to later HTTP/API concepts such as Content - Type.

```
Email
+-- text/plain
+-- text/html
+-- image/jpeg
+-- application/pdf
```

14. Security: why "From:" alone is not enough

Introduce spoofing with a simple question: **"If a message says From: admin@cs17.org, how can the receiver gain confidence that a legitimate cs17.org mail system sent it?"**

SPF is now easy to connect to their IP knowledge: a domain can publish which servers/IP ranges are authorized to send mail for it. **DKIM** adds a cryptographic signature. **DMARC** specifies policy and reporting around authentication results. Keep all three conceptual; configuration is outside this lesson.

```
Message claims: @cs17.org
Connection came from: 203.0.113.50
      |
      v
Check domain authentication / policy
      |
      v
Is this sending system expected / authorized?
```

Phishing mini-challenge

```
From: Google Security <security@google-account-alert.xyz>
Subject: URGENT - Account closes in 30 minutes
```

Click now: <https://google-security-login.xyz>

Ask students to identify: strange domain, urgency/threat, suspicious link, unexpected request, credential request, attachment risk, and whether the displayed sender name proves identity.

15. Hands-on mission

CS17 EMAIL + NETWORK MISSION

Students complete both the practical email task and the network reasoning task.

Part A - Send a real email

- Send a professional 4-6 sentence email to the teacher with a useful subject.
- CC one classmate and BCC another.
- Attach a PDF or image.
- Receive a reply, use Reply appropriately, then forward the message.
- Find the message later using search.
- Open "Show original" / message source and find at least one Received header.

Part B - Trace the network journey

- Take the recipient email address and identify its domain.
- State which DNS record should be queried first for email delivery.
- Explain what the MX result represents.
- Explain how the MX hostname becomes an IP address.
- Explain why the sender also needs a port number.
- Name the application protocol used to transfer mail between servers.
- Explain why the recipient laptop can be offline and still receive the email later.

16. Recap challenge

Without looking at notes, students redraw this chain and explain every arrow:

```
EMAIL ADDRESS
  v
DOMAIN
  v
DNS MX
  v
MAIL SERVER HOSTNAME
  v
A / AAAA
  v
IP ADDRESS
  v
TCP + PORT
  v
SMTP
  v
RECIPIENT MAIL SERVER
  v
MAILBOX
```

Success criterion

If a student can explain this chain in their own words - and distinguish the recipient mail server from the recipient laptop - the technical objective of the class has been met.

17. Printable homework - Email Network Detective

Name: _____ Date: _____

Use the lesson concepts to answer in your own words.

1. In the address `ali@cs17.org`, what is the mailbox part and what is the domain?
2. Why can't a sending mail server simply stop after learning the recipient domain?
3. Which DNS record tells a sender which server handles email for a domain?
4. Does an MX record normally give an IP address directly? If not, what does it give?
5. Which DNS records can map a mail-server hostname to IPv4 or IPv6 addresses?
6. What does an IP address identify? What does a port number identify?
7. What is the main role of TCP in this simplified email journey?
8. What is SMTP used for?
9. Why can you send an email to someone whose laptop is switched off?
10. What can a Received header reveal about an email?
11. In one sentence each, what are SPF, DKIM and DMARC trying to help with?
12. List four warning signs that might indicate a phishing email.

Final diagram

Draw the journey of a message from `sara@gmail.com` to `ali@cs17.org`. Your diagram must include: sender mail server, DNS, MX, mail-server hostname, IP address, TCP/port, SMTP, recipient mail server and mailbox.
