Email Administration

Don Porter

Theme

- Email systems started with a pretty simple design
  - Everyone trusted each other, it was nice
- Then spam came along
  - Lots of complexity and distrust to try to reduce spam

What is an email?

- A simple message format:
  - Envelope (server-internal routing info, not user-visible)
  - Headers (basically the history of the message handling)
    - These are viewable in your email app
  - Body (the text you see in your email app)

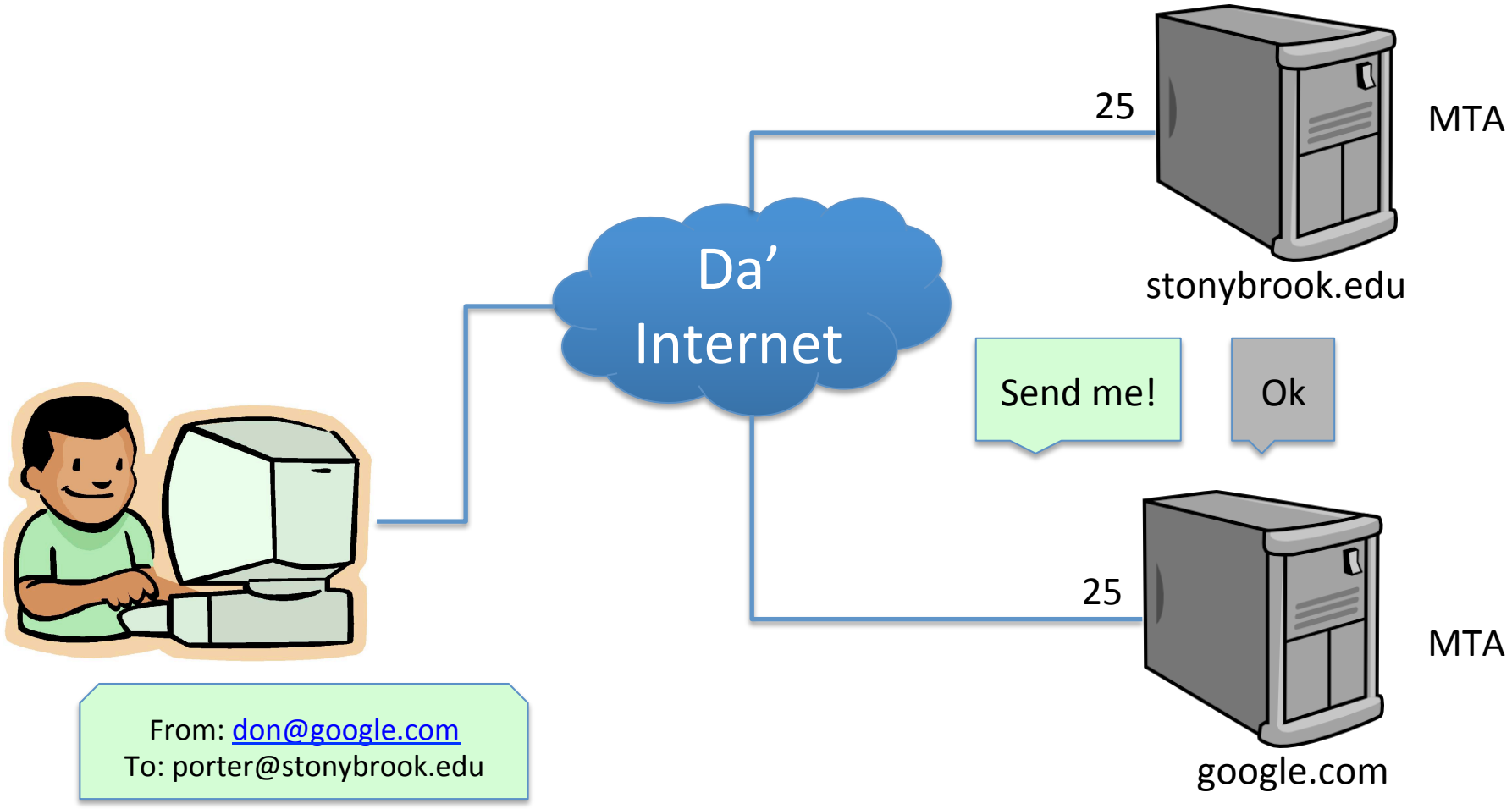
Example email header:

Received: from edge2.cs.stonybrook.edu (130.245.9.211) by hubcas2.cs.stonybrook.edu (130.245.9.209) with Microsoft SMTP Server (TLS) id 14.3.158.1; Thu, 13 Feb 2014 14:46:08 -0500
Received: from sbmta2.cc.stonybrook.edu (129.49.2.199) by edge2.cs.stonybrook.edu (130.245.9.211) with Microsoft SMTP Server (TLS) id 14.3.174.1; Thu, 13 Feb 2014 14:46:05 -0500
Received: from mailrelay.stonybrook.edu (mrs.stonybrook.edu [129.49.1.206]) by sbmta2.cc.stonybrook.edu (8.14.4/8.14.4) with ESMTP id s1DJk7Z7008862; Thu, 13 Feb 2014 14:46:07 -0500 (EST)
Received: from chalk2.ic.sunysb.edu (bbgw.ic.sunysb.edu [129.49.1.57]) by mailrelay.stonybrook.edu (8.14.2/8.14.2) with ESMTP id s1DJk7lm019593; Thu, 13 Feb 2014 14:46:07 -0500 (EST)
Received: from chalk2 (localhost.localdomain [127.0.0.1]) by chalk2.ic.sunysb.edu (8.13.8/8.13.8) with ESMTP id s1DJk6Zq029777; Thu, 13 Feb 2014 14:46:06 -0500
Date: Thu, 13 Feb 2014 14:46:06 -0500
From: Donald Porter <Donald.Porter@stonybrook.edu>
To: -CSE 311.01 / ISE 311.01 Systems Administration - Spring 2014:;
Message-ID: <1552521034.1214.1392320766984.JavaMail.bbuser@chalk2>
Subject: -CSE 311.01 / ISE 311.01 Systems Administration - Spring 2014: 2 Clarifications on Lab 1
Content-Type: multipart/alternative; boundary="----=_Part_1213_176163800.1392320766979"
X-Brightmail-Tracker: AAAAAgAAAUAAAAFS
X-Brightmail-Tracker: AAAAAgAAAUAAAAFU
Return-Path: Donald.Porter@stonybrook.edu
X-MS-Exchange-Organization-PRD: stonybrook.edu
Received-SPF: None (edge2.cs.stonybrook.edu: Donald.Porter@stonybrook.edu does not designate permitted sender hosts)
X-MS-Exchange-Organization-Antispam-Report: v=2.1 cv=JqnI8qIC c=1 sm=1 tr=0 a=pn1Q8qIfytiHm8Yza5mcig==:117 a=763spsVpnPbe6MGV/ff7eQ==:17 a=8V5aTH39tWIA:10 a=1Ov9T8keMbsA:10 a=ynS6Qj0ZrXwA:10 a=smqkL0zb6yZfvFi7FqQA:9 a=QEXdDO2ut3YA:10 a=CB515-41zfoA:10 a=sP84-5C6CZIA:10;OrigIP: 129.49.2.199;SCL:-1
X-MS-Exchange-Organization-AVStamp-Mailbox: MSFTFF;1;0;0 0 0
X-MS-Exchange-Organization-SCL: -1
X-MS-Exchange-Organization-SenderIdResult: NONE
X-MS-Exchange-Organization-AuthSource: edge2.cs.stonybrook.edu
X-MS-Exchange-Organization-AuthAs: Anonymous
X-MS-Exchange-Organization-Antispam-Report: MessageSecurityAntispamBypass
MIME-Version: 1.0

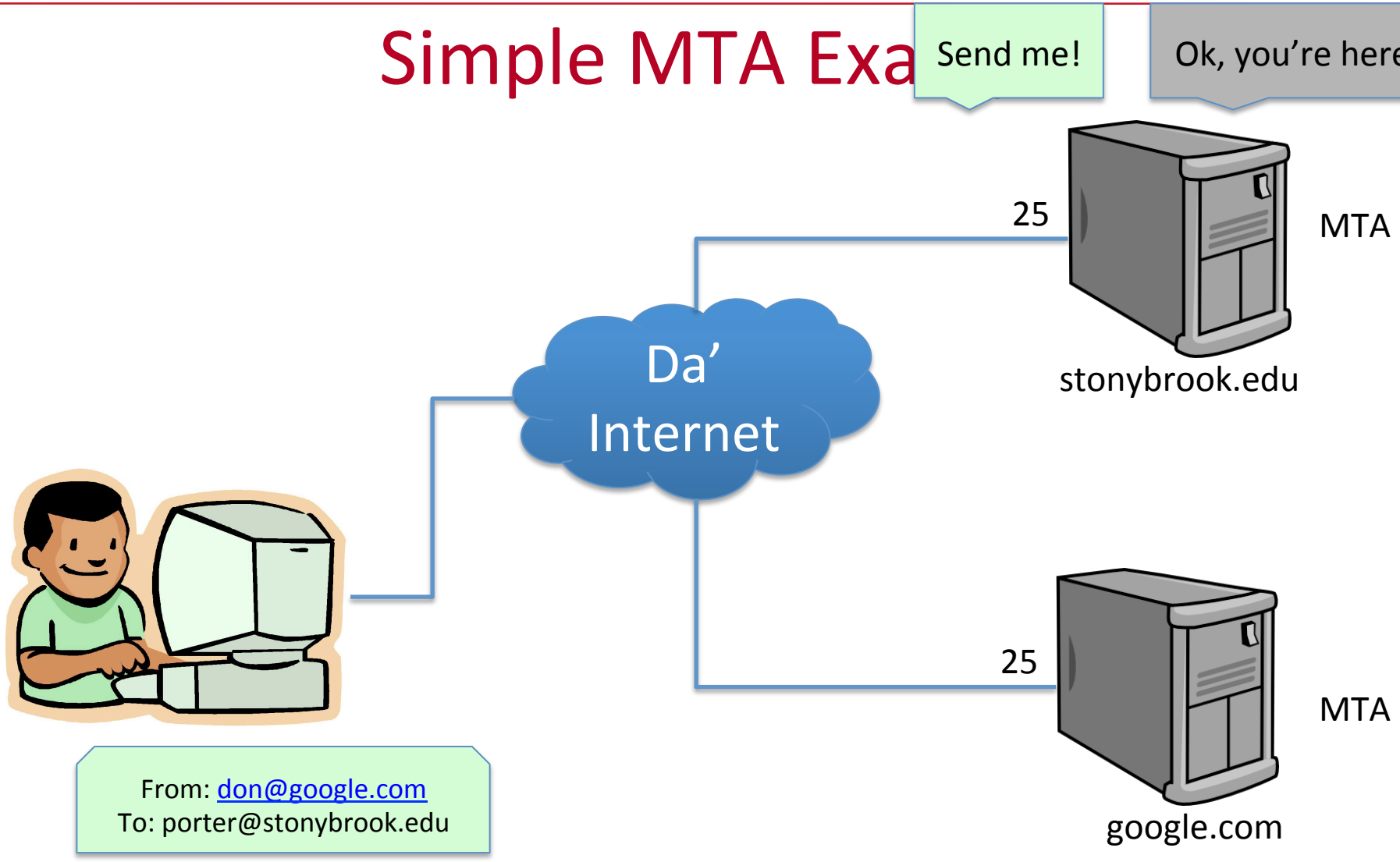
Mail Transport Agent (MTA)

- Accepts emails from the Internet
  - Delivers to local users
  - Or sends outgoing messages
- Every site that accepts email runs one
  - Identified with an MX record in DNS
- Listens on Port 25

Simple MTA Example



Simple MTA Exa



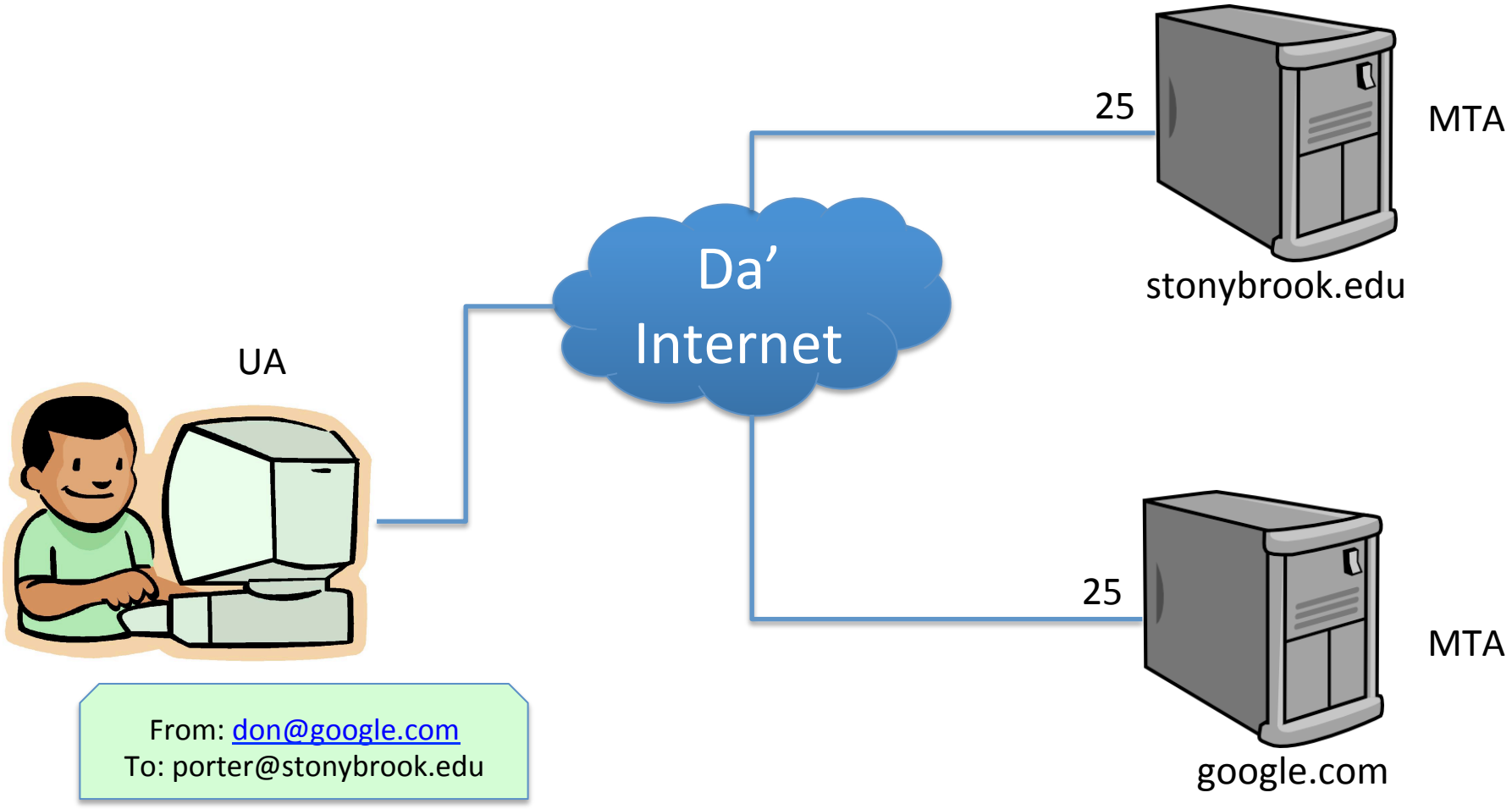
SMTP

- Simple Mail Transport Protocol
- It really is simple.
- Main operations:
  - Send a message
  - Check if an address is valid
  - Expand an address (for lists and forwarding)
- Email basically works by lots of MTA servers passing messages to each other

So what is Apple Mail, or Outlook?

- A User Agent (UA)
- Usually provides a program to type emails
- Ultimately packaged and sent to an MTA using SMTP

Simple MTA Example, redux



So how does email get to my inbox?

- Once an email reaches its destination MTA,
  - Handed off to a mail delivery agent (MDA)
  - MDA can be the same program or a different daemon as MTA
- MDA finds inbox for the recipient and stores the email

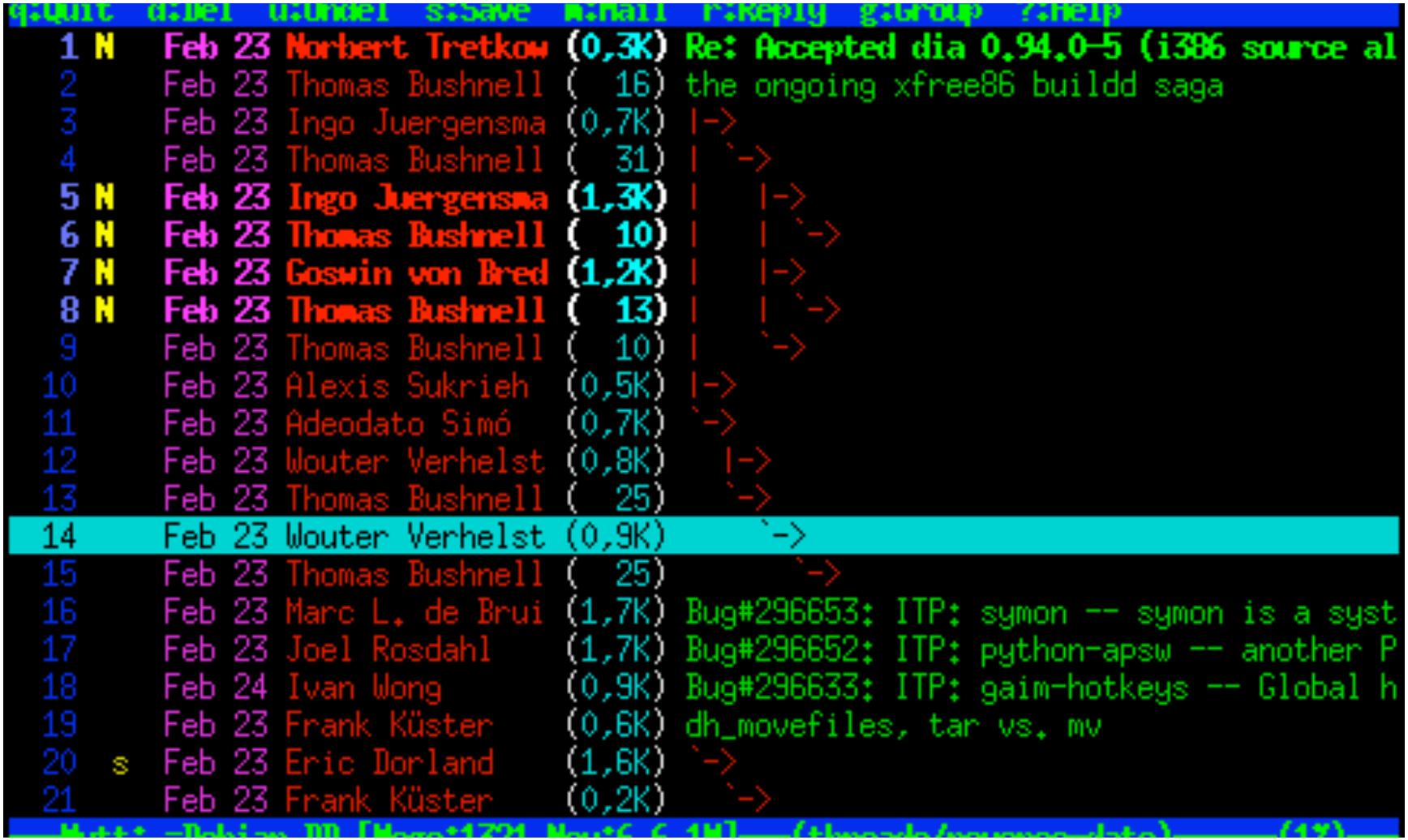
Where *is* my inbox?

- On Unix/Linux, it is either a file or directory
- 2 Popular formats:
  - mbox – single file
  - maildir – directory with one file per-message
- Literally, a file like: /home/porter/mail/mbox
  - Each message has a special delimiter between it
- Maybe shared with other machines over NFS
  - Users can get their mail without logging into MDA machine

Old-School Mail Reading

- Programs like mutt, pine, elm would read these mail files directly on a server
  - You could even open them in vi or emacs!

Text-based email reading (mutt)



From: http://greek0.net/mutt.html

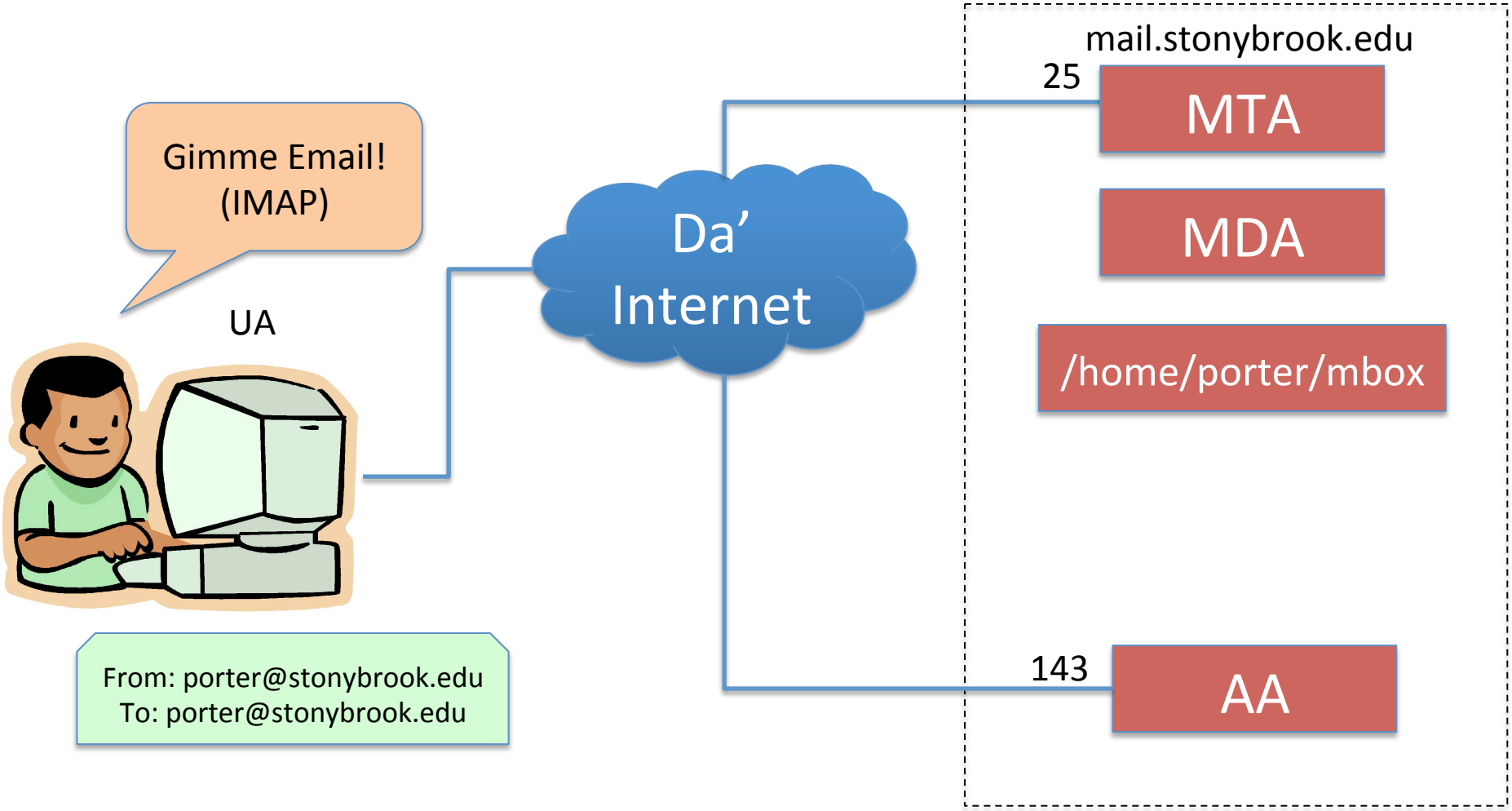
Pointy-Clicky Mail Reading

- Not everyone runs a command line on a Linux server
- Mail for the rest of us?
  - 2 popular protocols: POP and IMAP
  - Make a nice GUI app, like Thunderbird or Apple Mail
  - Download inbox using POP or IMAP
  - Still send using SMPT
- Access Agent (AA) – serves POP or IMAP

POP vs. IMAP

- Post Office Protocol (POP)
  - Download everything and (usually) delete it from inbox on server
  - Designed when people had one PC
- Internet Mail Access Protocol (IMAP)
  - Sync a local copy of mailbox with server
    - Work offline and sync later
  - Multiple clients
    - Laptop, desktop, iphone, ipad
  - Better choice for most modern users

Putting it together



Recap so far…

- User Agent: Program for human user
- Sending mail?
  - Protocol: SMTP
  - Emails exchanged by MTAs
  - Delivered to inboxes by MDAs
  - Inboxes are just simple files (mostly)
- Receiving mail?
  - Read inbox through an AA
  - Protocols: POP or IMAP

SMTP Review

- Main operations:
  - Send a message
  - Check if an address is valid
  - Expand an address (for lists and forwarding)
- **Anything missing?**

Are my emails private?

- No
- Sent in plain text over internet, inbox in plain text
- If you want privacy, use encryption in your UA
  - PGP/GPG are good choices

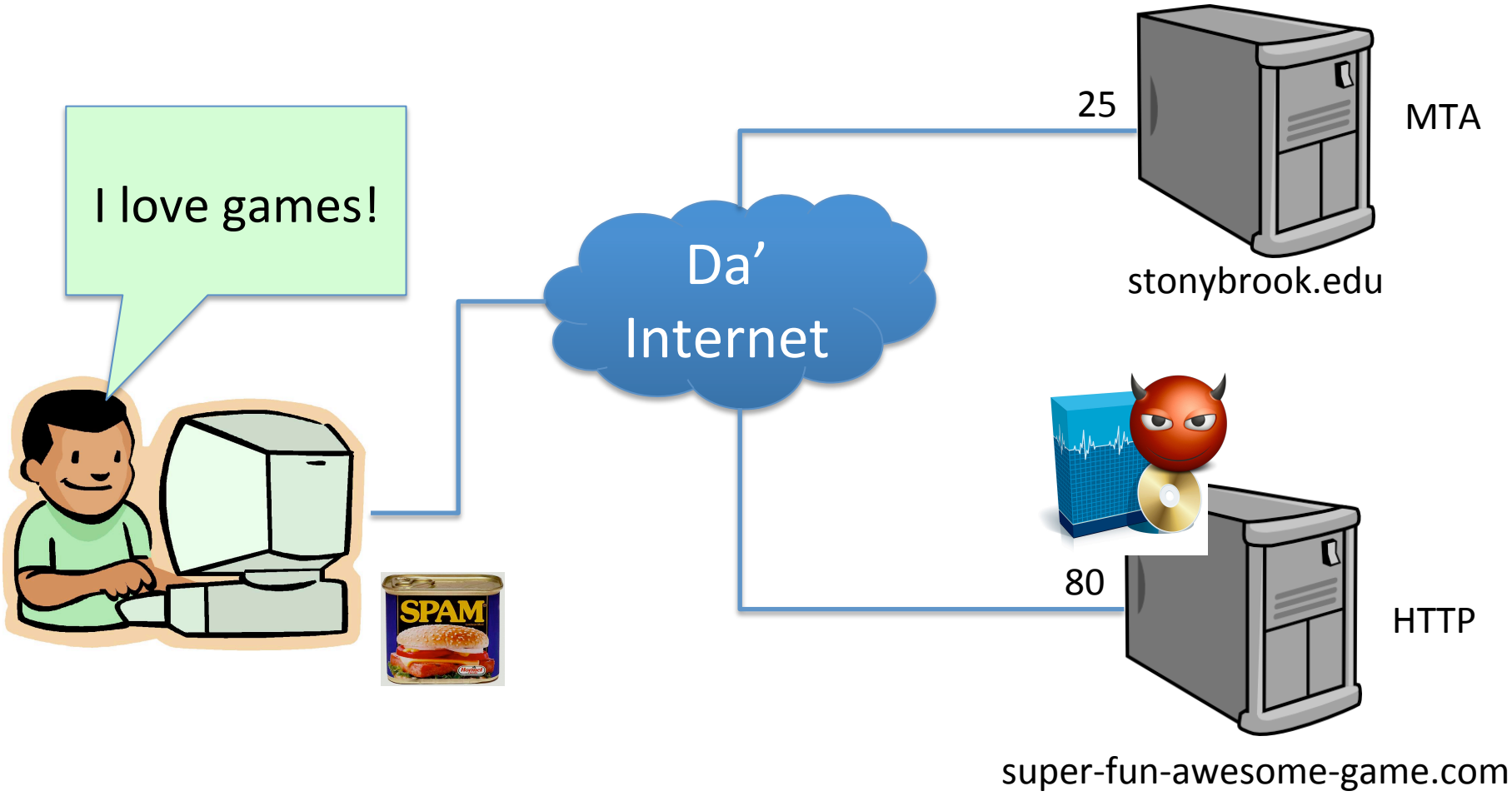
How do I know email came from sender?

- You don't
- Anyone can put any name/email address in the 'from' field
  - Of course, replies may not go to you…
- This is why people who care use PGP/GPG to *sign* messages

Spam

- Junk email
  - Unwanted ads, harassing emails, etc.
  - Often selling illegal products
- Why is it called spam?
- A massive nuisance
  - But also a massive business
- A lot of modern system organization designed to limit/prevent spam
  - We will refine previous model of email to cope with spam

An all-too-common example



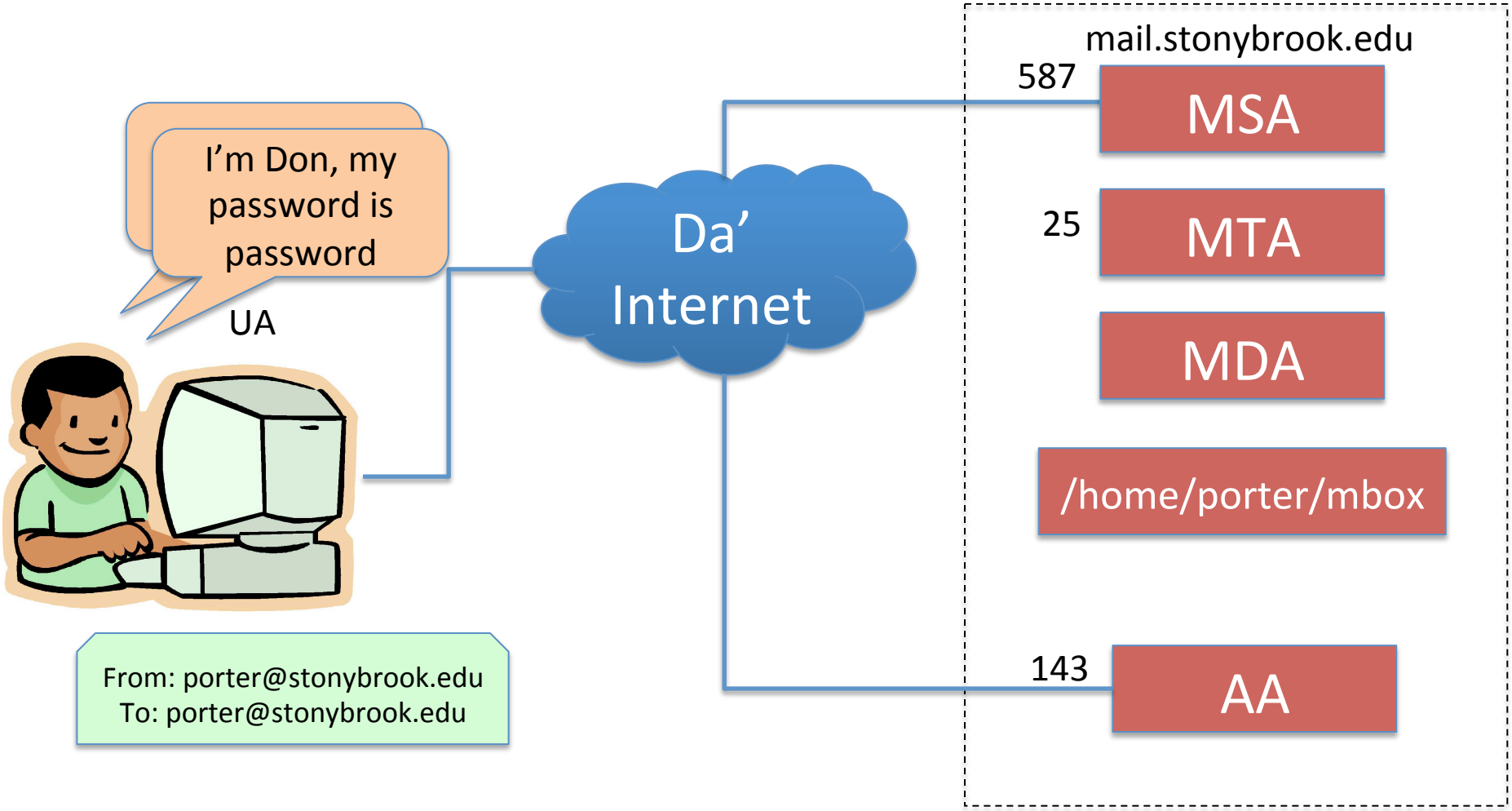
Example Recap

- User downloads software
  - Software includes “trojan horse” malware
  - Malware connects to email servers and sends spam from user’s computer over SMTP
  - Other SMTP servers then relay the email throughout the internet
- Ideas how to fix this?

Refinement 1

- SMTP servers (as presented) are dumb
  - Accept and relay mail from anyone!
- Idea: Only send mail from authenticated users
- Mail Submission Agent (MSA)
  - Basically, provides a log-in system, and then forwards mail to MTA
  - MTA configured to only relay mail from MSA
  - MSA often listens on port 587

Putting it together



Refinement, recap

- In the “good old days”, any email server would send your mail for you
  - Made email (and spam) easy
- Now, only servers that know you will relay email for you
  - Spam program at least has to steal your account info now

Problem 2

- The first example was about *relaying*
  - Using stonybrook.edu to relay email to lots of other addresses (gmail, hotmail, etc)
  - Fixed in part by:
    - Only accepting new email for legitimate users in stonybrook.edu
    - Only relaying email from authenticated stonybrook.edu users
- What about spam for users of stonybrook.edu?

Refinement 2

- Be choosier about who you accept email from
- Whitelists: known-good email servers (accept)
- Blacklists: known-bad email servers (reject)
  - Services that provide both for administrators
- Unknown servers? (Gray lists)
  - Reject: May lose email
  - Delay: Spammers often impatient

Refinement 3: Spam Detection

- Before the MDA delivers spam, run it through a filter
- If it passes, deliver to inbox
- If it fails, put in Junk folder
- How do spam filters work?

2 Strategies for Spam Filters

- Fuzzy matching against known spam
  - Verbatim matches foiled by including time of day
  - Services track these things for admins
- Bayesian learning filters
  - Users mark things as ‘spam’ or ‘ham’
  - Basically training a fancy-dancy machine learning classifier
  - Classifier applied to new mail
  - Learns user preferences over time

2 Phase Spam Prevention

- My advice (and the book's):
  - Do quick checks at the MTA (white list, black list)
  - Then do detailed checks (spam filtering) in MDA
- Why?
  - Quickly drop obvious garbage
  - Shift load for heavier-weight scanning to MDA

Summary

- MTAs exchange mail all over internet
  - Only relay outgoing mail from MSA (to prevent spam)
  - Only accept incoming mail from white/gray list (to prevent spam)
- MDA delivers mail
  - After it passes a filter (to prevent spam)
  - Stores it in a simple inbox file
- UA gets mail using IMAP/POP
  - Sends it via MSA using SMTP