

History of The Internet after 1981

[Click here](#) for further background to the early Internet after 1971

[Click here](#) to go to **Janet (Joint Academic Network)** used in the UK for email after 1985

[Click here](#) for the Rise of the New Networks in the US and Worldwide 1981 - 2010, the lead-up to Google and Cloudflare

How did MTAs work out destination IP address on obscure emails during the 1980s?

Mail Transfer Agents (MTAs) resolve the destination IP address of an email by evaluating the routing format of its delivery envelope. Since 1985, this mechanism has evolved from rigid, localized host files and manual path-routing to the globally automated, multi-tiered Domain Name System (DNS) lookups used today.

When an MTA encounters an "obscure" email address—such as an address with an legacy format, a direct IP fallback, or non-standard syntax—it uses specific programmatic rules to deduce the next hop.

Era 1: Pre-1986 (The Transition from HOSTS.TXT and Bang Paths)

Before 1986, the Internet was transitioning away from centralized host tables and alternative networks like [UUCP](#) (Unix-to-Unix Copy). MTAs did not have a uniform global DNS to dynamically find IP addresses.

- **The "Bang Path" Method:** Emails did not use `user@domain.com`. Instead, they used explicit network routes like `hop1!hop2!hop3!user` (Bang syntax). The MTA did not look up an IP address for the final recipient; it only looked at the *next immediate machine* (`hop1`) in its local routing table, stripped that name off, and passed the message forward.
- **HOSTS.TXT:** For systems on the early ARPANET, MTAs checked a single, master text file (`HOSTS.TXT`) maintained by the Network Information Center (NIC). If a host wasn't manually registered in that text file with a hardcoded IP address, the email bounced immediately.

Era 2: 1986–1990s (The Birth of RFC 974 and MX Records)

The foundational shift occurred in **January 1986** with the publication of [RFC 974: Mail Routing and the Domain System](#). This established the **Mail Exchanger (MX) record** system.

- **Standard MX Lookups:** An MTA takes the domain portion (to the right of the `@`), queries DNS for an `MX` record, receives a list of priority-ranked hostnames, and queries the `A` record of the highest-priority host to get the destination IP address.
- **Handling Obscure CNAMEs:** If a domain did not have an `MX` record, early MTAs would search for a `CNAME` (canonical name alias). If found, the MTA would recursively start the `MX` and `A` record lookups all over again on the underlying true domain name.
- **The Direct Fallback Rule:** If a domain completely lacked an `MX` record, RFC 974 mandated that MTAs fall back to treating the domain name as a direct host name. The MTA would look

up the domain's **A** record directly and attempt an SMTP connection to that IP address, assuming the domain's web server doubled as its mail server.

Era 3: 2000s–Present (Advanced Handling of "Obscure" Syntax)

Modern MTAs (such as Postfix, Exim, and Sendmail) use deterministic parsing chains to extract destination IPs when addresses look unusual:

Obscure Address Type	Example Syntax	How the MTA Resolves the Destination IP
Domain Literals (Direct IP)	<code>user@</code> or <code>user@[IPv6:2001:db8::1]</code>	Bypasses DNS entirely. The MTA recognizes the square brackets <code>[]</code> , extracts the literal IP address, and immediately opens an SMTP socket connection directly to that network address.
The "Smart Host" Relay	Internally routed email or legacy systems	If an MTA encounters a domain it cannot resolve through standard internet DNS, it relies on a local configuration file containing a Smart Host rule. It offloads the email to a designated relay server's IP address, delegating the lookup to a higher-level routing engine.
Sub addressing / Plus Tagging	<code>user+shopping@domain.com</code>	The MTA ignores everything between the <code>+</code> and the <code>@</code> . It extracts <code>domain.com</code> , performs a normal MX lookup to get the IP, and leaves it up to the destination server's internal Mail Delivery Agent (MDA) to sort the label later.
Internationalized Domains (IDN)	<code>user@مثال.إختبار</code> or <code>user@example.com</code>	MTAs translate Unicode characters into an ASCII compatible string via Punycode (e.g., <code>xn--xample-9va.com</code>). The MTA then requests the standard MX and A/AAAA records for the Punycode string to pull the destination IP.

If you are investigating a specific technical scenario, let me know:

- Are you dealing with a **specific error bounce code** or standard?
- Are you trying to route mail over a **private/local network** or the public internet?
- What **MTA software** (e.g., Postfix, Exchange, Sendmail) are you configuring?

Just seeking to document old history. When the hops were entered ca 1985, were these IP addresses? How were they arrived at?

No, around 1985, the hops in a "bang path" address were not IP addresses. They were textual hostnames of individual computers (e.g., `decvax`, `ihnp4`, `ucbvax`) linked together via the UUCP (Unix-to-Unix Copy) network.

Because UUCP was primarily a store-and-forward network based on dial-up telephone lines and direct serial cables, IP addresses did not exist on most of these machines.

How These Hops Were Arrived At

Before globally automated routing, determining how an email travelled from Computer A to Computer Z required human calculation, community mapping projects, and specialized graph-routing software.

1. The Human "Map" (Knowing the Backbones)

In the early 1980s, users had to manually figure out a chain of connections. You couldn't directly call just any computer; you could only talk to a machine with which your system administrator had a direct dial-up agreement (a "neighbor").

To email someone far away, you had to find a path through highly connected "backbone" hubs run by universities or corporations (like AT&T's `ihnp4` or UC Berkeley's `ucbvax`).

- **The Process:** A user at a small college might know: *"My machine (`localvax`) calls `stateuni` every night. I know `stateuni` talks to the backbone hub `ihnp4`. And my friend at a tech company tells me their machine (`foobar`) is connected to `ihnp4`."*
- **The Resulting Address:** The user would manually construct the string:
`stateuni!ihnp4!foobar!user.`

2. The UUCP Mapping Project

As the network grew, manual pathing became impossible. In 1984, volunteers formed the **UUCP Mapping Project** to manage the chaos.

- System administrators from all over the world would periodically email a standardized text entry listing their machine's name, geographical location, and its direct neighbors along with an arbitrary "cost" factor (e.g., how fast the modem was, or how frequently they dialed out).
- These entries were aggregated into files and published monthly in Usenet newsgroups.

3. Automation via `pathalias`

Once an administrator downloaded the global UUCP map files, they ran a specialized command-line utility called `pathalias`.

`pathalias` treated the entire global network as a giant mathematical directed graph. It calculated the "cheapest" (fastest or most reliable) route from the local machine to every other registered machine in the world, generating a local database table. When a user typed a relatively simple domain address, early mail agents would look at the `pathalias` table to inject the required string of "bang" hops automatically.

What Actually Happened at Each Hop

When the local machine processed `hop1!hop2!hop3!user`, it didn't use internet routing protocols. It executed a literal telephone call:

1. The mail agent looked at `hop1`, checked its local configuration file (often `/etc/uucp/L.sys` or `Systems`), and pulled a **physical telephone number** and a modem speed.
2. The computer used a physical modem to **dial the telephone number** of `hop1` over the standard public voice network.
3. Once connected, it used the UUCP protocol to transfer the email data file into a queue on `hop1`.
4. The connection hung up. `hop1` would later read the remaining path (`hop2!hop3!user`), see that `hop2` was its responsibility, look up `hop2`'s phone number, and repeat the process.

Because these connections often happened sequentially on nightly timers to minimize long-distance phone bills, an email with five or six hops could take days to reach its destination.

If you are looking to dig deeper into specific configurations, would you like to explore:

- How the `pathalias` **routing files** were structured?
- How early mailers like **Sendmail** mapped these legacy formats to the emerging internet standards?

Right, that clarifies heaps. What were key dates as it shifted to today's system.

The shift from the manual, dial-up, "bang path" network to the modern, automated, DNS-driven email system happened over a series of crucial technological milestones between 1982 and the early 2000s.

The chronological breakdown of the key dates that shaped today's email routing infrastructure is detailed below.

1. The Core Infrastructure (1982–1983)

- **1982 — RFC 821 & 822 (The Blueprint):** Jonathan Postel published [RFC 821](#) (establishing the **Simple Mail Transfer Protocol - SMTP**) and Dave Crocker published RFC 822 (defining the standard text structure of emails, like `To:`, `From:`, and `Subject:`).
- **January 1, 1983 — The ARPANET Flag Day:** The military/academic ARPANET completely shut down its old NCP network protocol and switched entirely to **TCP/IP**. This provided the baseline internet pipe that allowed SMTP mailers to communicate with stable, numeric IP addresses rather than just physical telephone wires.
- **November 1983 — The Birth of DNS (RFC 882 & 883):** Paul Mockapetris designed the **Domain Name System (DNS)** to eliminate the massive, centralized `HOSTS.TXT` master file. Initially, it used clunky `MD` (Mail Delivery) and `MF` (Mail Forwarder) records that required separate, inefficient lookups.

2. The Invention of Modern Routing (1986–1989)

- **January 1986 — The MX Record is Born (RFC 973 & 974):** Realising that `MD` and `MF` records were broken, Craig Partridge authored [RFC 974](#). This officially introduced the **Mail Exchanger (MX) record**. For the first time, a domain name could point to a completely different server for its email traffic with built-in numerical "priorities".
- **1987 — RFC 1035 (DNS Finalised):** This foundational document solidified the core internet directory structure we use today, tying `A` records (IP lookups) and `MX` records (mail lookups) into an unbreakable standard.
- **1989 — The "Smart Host" Standard (RFC 1123):** This RFC updated requirements for internet hosts. It mandated that modern internet mailers *must* gracefully handle legacy routing syntax (like translating mixed bang paths and domain formats) and fallback elegantly if an `MX` record couldn't be reached.

3. The Great Convergence & The End of UUCP (1990–1997)

- **1990–1992 — MX Overcomes the UUCP Map:** As commercial ISPs (Internet Service Providers) began offering cheap, always-on internet access to businesses and the public, systems abandoned dial-up telephone links. The old UUCP Mapping Project dissolved because it was no longer necessary to "hop" messages across volunteers' modems.
- **1992 — MIME Extensions (RFC 1341):** Email transitioned from purely sending plain text files to supporting attachments, images, and audio.
- **1996–1997 — Webmail and Commercial Domination: Hotmail (1996) and Yahoo Mail (1997)** launched. Because these huge platforms operated directly on the public web using standard DNS lookups, the remaining fringe private networks were forced to adapt to standard internet routing or face obsolescence.

4. The Modern Security Era (1999–2004)

By the turn of the millennium, the absolute trust of the 1980s internet routing system became its greatest weakness: anyone could fake an address. The system had to shift to enforce sender verification.

- **1999 — SMTP Authentication (RFC 2554):** Incorporated a basic extension requiring users to log in with a username and password to send outbound mail, blocking open-relay spam.
- **2004 — SPF (Sender Policy Framework) Introduced:** The ultimate evolutionary step for MTAs dealing with "obscure" or fraudulent emails. Domains began publishing DNS records specifying exactly *which* IP addresses were legally allowed to send mail on their behalf, effectively locking down the global destination matrix.

If you are putting together a historical log or code reference, let me know:

- Would you like an example of a **1980s Sendmail configuration file** (`sendmail.cf`) showing how it parsed these addresses?
- Are you looking for the specific history of a certain network (e.g., **BITNET**, **CSNET**, or **UUCP**)?

More interested how it worked after 1987 from the sender host

By **1987**, the email landscape was a "hybrid" world. The Domain Name System (DNS) and **MX records** were the official internet standards, but many machines still relied on dial-up UUCP networks [RFC974].

Because of this, an MTA (Mail Transfer Agent) on a sender host in the late 1980s—most commonly **Sendmail** running on Unix—operated as a complex sorting engine. It used a brutal series of string-parsing rules called **Rewrite Rules** to figure out whether to pass a message to a local modem, an ARPANET gateway, or directly onto the Internet via TCP/IP [RFC974].

Here is exactly how a sender host worked out the destination after 1987.

Step 1: Parsing the Address (The Sendmail Rulesets)

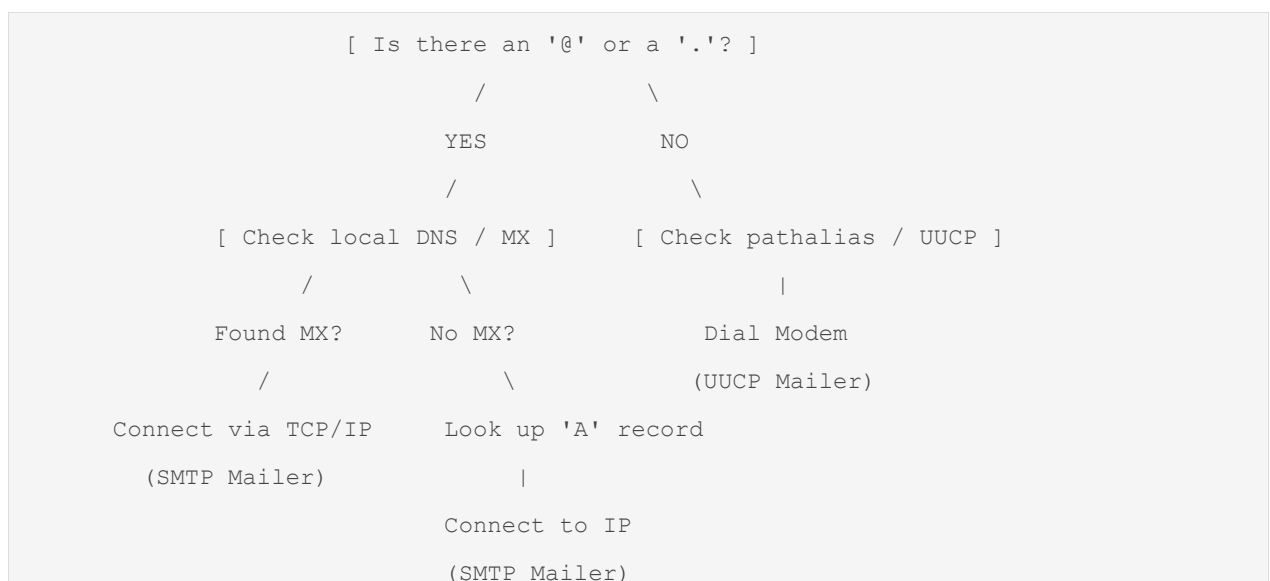
When a user hit "send," the email was handed to the local MTA. The MTA didn't immediately look at DNS. First, it had to figure out what *kind* of address it was looking at, because users frequently mixed syntaxes (e.g., `ucbvax!user@harvard.edu`).

The sender host ran the address through its `sendmail.cf` configuration file, applying mathematical text transformations (pattern matching):

- **Look for an @:** If the address contained an @, the host assumed it was an Internet-style domain address (`user@domain.com`).
- **Look for a !:** If it contained an exclamation mark, it was flagged as a UUCP dial-up address (`host!user`).
- **Look for a %:** The percent sign was used as a "hack" for source routing (`user%finalhost@gateway`). The sender host would rewrite this to target `gateway` first.

Step 2: Choosing the "Mailer" (The Internal Decision Tree)

Once the address syntax was parsed, the sender host selected a specific protocol plugin called a **Mailer** to handle the delivery.



Scenario A: The Target was a Domain (The TCP/IP Path)

If the sender host was lucky enough to have a direct internet connection (via an ethernet cable or leased line) and the address was a domain, it initiated the newly standardized [RFC 974](#) lookup:

1. **The MX Query:** The sender host sent a UDP packet to its local DNS nameserver asking for the **MX** (Mail Exchanger) records of the destination domain.
2. **The Priority Sort:** The nameserver returned a list of hosts with priority numbers (e.g., 10 ://domain.com, 20 ://domain.com). The sender host sorted these, lowest number first.
3. **The IP Lookup (A Record):** The sender host then asked DNS for the **A** record (the actual 32-bit IPv4 address) of ://domain.com.
4. **The Connection:** The sender host opened a TCP connection on **Port 25** to that IP address and used SMTP to transfer the mail.

Scenario B: The Target was a Domain, but had NO MX Record

In 1987, many domains hadn't updated to the new MX standard yet. If the DNS query returned "No MX record found," RFC 974 required the sender host to fallback:

- The sender host immediately requested the **A** record for the domain name itself.
- If it found an IP address, it assumed the company's main web/file server was also running an mail daemon, connected to that IP on Port 25, and handed over the mail.

Scenario C: The Sender Host was "Off-Grid" (The UUCP Path)

If the sender host itself did not have an internet connection (it only had a modem), or if the destination was a legacy UUCP address, it used its local database:

1. It queried its local **pathalias database** (compiled from the monthly UUCP Mapping Project text files).
2. The database would tell the host: *"To get to remotehost, you must first dial neighborhost."*
3. The sender host translated the address into a full path and queued the file locally.
4. The **UUCP Mailer** took over, commanded the physical modem to dial the telephone number for neighborhost, and dumped the mail file over the phone line.

The 1987 Transition Nightmare: "Domainizing" UUCP

The biggest friction point after 1987 was connecting these two worlds. To make UUCP networks play nice with the internet, the community introduced **pseudo-domains** like .uucp or .bitnet.

If a user on the internet sent an email to user@host.uucp:

1. The internet sender's MTA would query DNS for .uucp.
2. DNS would return an MX record pointing to a massive institutional **Gateway Machine** (like relay.cs.net or uunet.uu.net) that possessed *both* an internet connection and a massive bank of modems.
3. The internet host routed the mail via IP to the gateway.
4. The gateway stripped off the internet headers, looked up the host in its phonebook, and dialed the destination computer at 1200 or 2400 baud over a telephone line to finish the delivery.

If you are tracking down how specific code handled this, I can pull up an original **1980s** `sendmail.cf` **rewrite rule block** to show you what the actual configuration syntax looked like. Would you like to see how they used characters like `$X`, `$D`, and `$+` to manipulate those addresses?

No, that joins the links pretty well. Thanks. Just been looking at Janet, 1985 - 1994.

Focusing on **JANET (Joint Academic Network)** between 1985 and 1994 uncovers one of the most unique and distinctive eras in email history.

While the rest of the world was standardising on TCP/IP and the American, "little-endian" Domain Name System (`user@cam.ac.uk`), the UK academic community went its own way. JANET operated on **X.25 packet-switching network infrastructure** and used the UK's proprietary **Coloured Book protocols** (specifically the *Grey Book* for email).

This setup created distinct technical challenges and quirks for a sender host during that period:

1. The Big-Endian vs. Little-Endian "Order" Wars

The most famous aspect of JANET routing was that its **Name Registration Scheme (NRS)** ordered domains backwards relative to the global internet standard. JANET addresses were **Big-Endian** (most significant element first) and capitalized:

- **JANET Format:** `USER@UK.AC.CAM`
- **Internet Format:** `user@cam.ac.uk`

If a sender inside the UK typed `USER@UK.AC.CAM`, a JANET mailer processed it hierarchically from left to right: *Look for the UK network → Look for the Academic sector (AC) → Look for Cambridge (CAM)*.

However, if an American MTA received this, it would try to parse it from right to left, look for a top-level domain named `.CAM`, fail immediately, and bounce the mail.

2. How the Sender Host Resolved Destinations Over X.25

Because JANET did not use TCP/IP natively until the **JIPS (JANET Internet Protocol Service)** rollout began in **1991**, it could not perform standard DNS lookups or establish direct SMTP TCP/IP connections.

Instead, a sender host on JANET in 1987 worked like this:

1. **The Local Static Lookup:** The sender host checked a locally compiled NRS database (downloaded periodically from the central JANET registrar).
2. **Finding the DTE Address:** Instead of looking up a 32-bit IP address, the NRS database mapped `UK.AC.CAM` to an **X.121 DTE address** (a string of up to 14 digits, similar to a telephone number used for X.25 data routing).
3. **Establishing the Call:** The sender host's Grey Book mail software instructed its X.25 network card to open a virtual circuit directly to that digit string across the JANET backbone.

3. The Gateway Eras (1985–1994)

To prevent the UK from becoming a digital island, email had to bridge the gap between JANET's X.25/Grey Book network and the global Internet's TCP/IP/SMTP network. This was achieved through specialized, monumental **Gateway Machines**:

- **The Backbone Interconnect:** The primary gatekeeper during this entire era was run by **University College London (UCL)** via a relay machine named `cs.ucl.ac.uk` (and later, commercial gateways like `uunet`).
- **Outbound (UK to US):** When a British researcher sent an email to `user@harvard.edu`, the local JANET host recognised it wasn't a `UK` address. It routed the mail over X.25 to the UCL gateway. UCL translated the Grey Book headers into standard SMTP, flipped any necessary address components, looked up the US IP address via standard internet DNS, and pushed it across an transatlantic link.
- **Inbound (US to UK):** If an American sent mail to `user@cam.ac.uk`, their US sender host queried global DNS. JANET worked behind the scenes to ensure that DNS entries for UK universities actually pointed to the IP address of the UCL gateway. The gateway received the mail over IP, **completely reversed the domain string to Big-Endian**, converted the message to Grey Book format, and queued it to send over X.25.

4. The 1991–1994 Great Collapse

By **1991**, the explosion of global Internet traffic made maintaining a separate X.25/Coloured Book standard unsustainable.

- **March 1991:** JANET introduced the **JIPS pilot**, allowing standard Internet Protocol (IP) encapsulation to run on top of the existing network lines.
- **November 1991:** IP traffic exploded so fast that it completely overtook traditional X.25 volume, and IP routing officially became the primary standard for JANET.
- **By 1994:** JANET shifted onto **SuperJANET** (a high-speed fibre-optic ATM system). The old Big-Endian NRS database was systematically decommissioned in favour of global Domain Name System (DNS) servers.

The only surviving legacy of this entire decade-long architectural divergence is that the UK uses `.uk` as its country code top-level domain instead of `.gb` (the official ISO country code), because `UK` was already burned into the foundation of the old big-endian JANET Name Registration Scheme.

Ok, to summarize, Janet joined the Internet in 1989 with JIPS forming in 1991

- **1989 Link to the Internet:** The [United Kingdom's Joint Academic Network \(JANET\)](#) originally launched in 1984 utilizing the UK's proprietary "Coloured Book" protocols over X.25. It officially **established its first gateway to the NSFNET in 1989**, establishing its foundational link to the global Internet backbone.
- **1991 JIPS Pilot:** Planning and a pilot project for the **JANET Internet Protocol Service (JIPS)** launched in **March 1991**. This critical move allowed native TCP/IP traffic to run over the existing infrastructure. Within just eight months, IP traffic completely eclipsed old protocol traffic volumes, leading to IP support becoming the official standard for JANET in November 1991.

What did Janet do in 1989 before JIPS in 1991 as regards it's joining the Internet.

In 1989, when JANET first connected to the United States' [NSFNET](#) (the primary Internet backbone at the time), it did not actually use native Internet Protocol (TCP/IP) traffic.

Instead, JANET bridged the gap using **protocol translation and application relays**. Because JANET operated on **X.25 packet-switching infrastructure** and relied on the British [Coloured Book protocols](#), joining the global Internet in 1989 involved a few key workarounds:

- **Application-Level Gateways:** Rather than allowing a computer in the UK to talk directly to a computer in the US using IP packets, specific gateway computers (largely hosted at institutions like [University College London](#)) acted as translators. For example, a UK user sending an email would use the **Grey Book protocol**. The gateway would receive it, convert it into standard Internet **SMTP**, and send it over the NSFNET link.
- **File Transfer Translation:** File transfers between the UK and the global Internet relied on translating JANET's **Blue Book protocol** into the Internet's standard **FTP** at the border.
- **Address Mapping (NRS to DNS):** The UK academic system used the JANET [Name Registration Scheme \(NRS\)](#), which processed addresses in "big-endian" order (e.g., `UK.AC.CAMBRIDGE` instead of `cam.ac.uk`). In 1989, gateways had to map these reverse addresses to the Internet's standard **Domain Name System (DNS)** so international traffic could route correctly.

Essentially, JANET built a "bridge" to the Internet in 1989, but traffic had to change vehicles at the border until **JIPS** turned the entire network into a native IP highway in 1991.

If you want to dig deeper into this era, let me know if you would like to explore:

- The fierce "**Protocol Wars**" of the late 1980s between the Internet's TCP/IP and Europe's preferred OSI/X.25 models.

The fierce "Protocol Wars" of the late 1980s between the Internet's TCP/IP and Europe's preferred OSI/X.25 models

The "**Protocol Wars**" of the late 1980s and early 1990s were a massive, high-stakes battle over the foundational architecture of the global network. It pitted a grassroots, working-code ethos against a top-down, government-mandated standard.

Here is how the battle lines were drawn and why it shaped the modern digital world.

The Two Factions

The conflict was fundamentally an ideological and technical clash between two different networking philosophies:

Feature	The Challenger: TCP/IP (The Internet Model)	The Establishment: OSI / X.25 (The European Model)
Origin	Formed out of US defense research (ARPANET) and adopted heavily by academia (NSFNET).	Designed by official international standards bodies (ISO, CCITT) and backed by European PTTs (telecom monopolies).
Philosophy	"Rough consensus and running code." If it worked in practice, it was adopted.	"Perfection on paper." Exhaustive, highly detailed theoretical frameworks designed before implementation.
Network Smartness	Dumb network, smart endpoints. The network just passes packets; the computers at the edges handle errors.	Smart network, dumb endpoints. The network checks for errors and manages connections; terminals just send data.
Technology	Connectionless packet switching. Packets find their own way to the destination.	Connection-oriented virtual circuits (X.25). Mimicked traditional phone calls by setting up a dedicated path.

Why Europe (and JANET) Clung to OSI and X.25

In the 1980s, European governments and telecom authorities were deeply uncomfortable with TCP/IP for several reasons:

- **American Dominance:** TCP/IP was seen as a proprietary, military-backed American standard controlled by the US Department of Defense.
- **Telecom Monopolies:** European Post, Telegraph, and Telecommunications (PTT) authorities wanted a system where they could charge users by the packet or by the minute, much like international long-distance phone calls. X.25 allowed for this strict accounting; TCP/IP did not.
- **The Mandate:** The European Commission and national governments legally mandated that public academic and government networks adopt **Open Systems Interconnection (OSI)** standards. This is why JANET developed its "Coloured Book" protocols—they were meant to be a stepping stone toward full, official OSI compliance.

The Turning Point: Why TCP/IP Won

By the late 1980s, the OSI faction had a fatal flaw: **it was taking too long**. The standards committees were bogged down in endless bureaucracy trying to design the perfect, comprehensive networking stack.

Meanwhile, TCP/IP was free, widely deployed, and built directly into [UC Berkeley's Unix operating system \(BSD\)](#), which ran on thousands of university workstations worldwide.

The war ended not because OSI was technically inferior, but because **TCP/IP actually existed and worked.**

- **The Academic Mutiny:** University researchers needed to share data immediately. They grew tired of waiting for official OSI software and quietly started running TCP/IP over their lines.
- **The JANET Pivot:** This tension is exactly why JANET launched **JIPS** in 1991. The official policy was still to wait for OSI, but the demand for native Internet access (TCP/IP) to collaborate with US researchers became a tidal wave that the UK academic community could no longer ignore.

By the mid-1990s, the Protocol Wars were over. OSI became a theoretical model used primarily for teaching networking layers, while TCP/IP became the undisputed fabric of the global Internet.

**** End of Page**