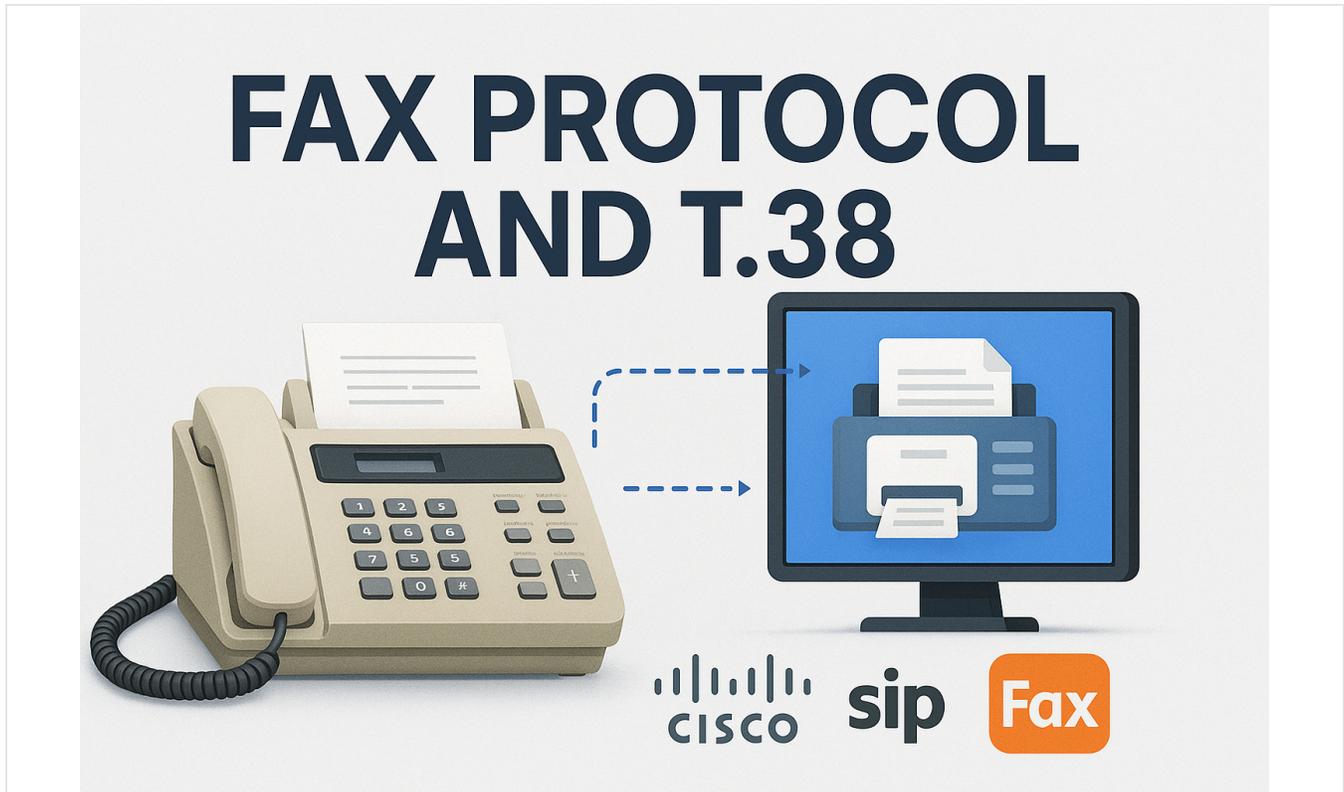


# The History and Technical Evolution of Fax Protocols

By ClearlyIP Published April 8, 2025 30 min read



## The History and Technical Evolution of Fax Protocols

### Origins and Evolution of Fax Technology

The concept of facsimile (fax) transmission dates back to the 19th century (e.g. Alexander Bain's 1843 invention of an image telegraph). However, modern fax technology truly emerged in the mid-20th century with the standardization of fax groups by the [International Telecommunication Union \(ITU-T\)](#). **Group 1** fax was introduced in 1968 (ITU-T Recommendation T.2) as the first worldwide fax

standard [dialogic.com](http://dialogic.com)[en.wikipedia.org](http://en.wikipedia.org). Group 1 devices were analog and very slow – on the order of **six minutes per page** at a vertical resolution of 96 lines per inch [en.wikipedia.org](http://en.wikipedia.org). **Group 2** fax followed in 1976 (Recommendations T.3 and T.30) with improved analog modulation techniques, cutting transmission time to about **three minutes per page** (still 96 lpi vertical resolution) [en.wikipedia.org](http://en.wikipedia.org). Both Group 1 and 2 transmit a page line-by-line as a continuous analog signal (much like a TV scanline), and these analog fax machines are now obsolete [en.wikipedia.org](http://en.wikipedia.org).

The real breakthrough came with **Group 3** fax in 1980, enabled by [digital technologies](#) that **digitized and compressed** the scanned page data [en.wikipedia.org](http://en.wikipedia.org). Group 3 fax became by far the most widely adopted standard. It introduced **digital image compression** (per ITU-T T.4) and faster **modems**, slashing fax transmission to roughly **6–15 seconds per page** [en.wikipedia.org](http://en.wikipedia.org)[en.wikipedia.org](http://en.wikipedia.org). Group 3 faxes operate over regular analog phone lines but use digital data modulations – they handshake and agree on parameters, then send a digital image bitstream modulated as tones. Typical Group 3 resolution is 203×98 dpi (standard) or 203×196 dpi (fine), with optional higher resolutions (e.g. 391×391 dpi “superfine”) allowed [en.wikipedia.org](http://en.wikipedia.org)[en.wikipedia.org](http://en.wikipedia.org). Group 3 fax machines also introduced the need for a little onboard memory (to buffer scan lines) [en.wikipedia.org](http://en.wikipedia.org). In practice, Group 3 fax standardized the **ITU-T T.30 protocol** (for session control) and the **T.4/T.6 image encoding**, and used a set of ITU-T V-series **modems** for data transmission [dialogic.com](http://dialogic.com)[dialogic.com](http://dialogic.com). We will discuss these technical underpinnings in the next section.

**Group 4** fax, finalized in the late 1980s, was designed for all-digital networks (e.g. the 64 kb/s ISDN service) [en.wikipedia.org](http://en.wikipedia.org). Group 4 machines could transmit high-resolution pages at even faster speeds, taking advantage of error-free digital circuits. The ITU-T T.6 recommendation introduced a more advanced **two-dimensional compression** called **Modified Modified READ (MMR)** for Group 4, achieving higher compression ratios than the one-dimensional or one-and-a-half-dimensional methods (**Modified Huffman (MH)** and **Modified READ (MR)**) used in Group 3 [networkworld.com](http://networkworld.com)[networkworld.com](http://networkworld.com). However, Group 4 fax saw limited deployment (mostly in enterprise or inter-bank networks) since analog phone lines and Group 3 remained dominant for decades. In summary, by the 1990s fax technology had evolved from slow analog scanning (Group 1/2) to efficient digital transmission (Group 3/4) using a stack of ITU-T standards: T.30 for control, T.4/T.6 for image coding, and various V.xx modems for physical signaling.

## Traditional Fax Protocols over Analog Phone Lines (PSTN)

Traditional Group 3 fax transmission over the [Public Switched Telephone Network \(PSTN\)](#) is built on a **handshake and session protocol (T.30)** and analog **modem communications**. When one fax machine calls another over the phone network, the two devices engage in a sequenced **T.30 session** that negotiates their capabilities and manages the transfer. The calling fax initially emits a special **calling tone** (CNG, usually 1100 Hz blips) to indicate a fax call, and the answering fax (if it's fax-equipped) responds with an **answer tone** (CED, 2100 Hz) for about 2–3 seconds [soft-switch.org](#). Following this, the real T.30 handshake begins: the devices exchange digital identification frames over the analog line. These control frames are sent as **HDLC data packets modulated at 300 bit/s** (using ITU-T V.21 FSK modulation) [networkworld.com](#). In essence, one can think of the fax handshake as a low-speed data link negotiation embedded within the phone call.

[ITU-T T.30](#) defines the entire fax **session protocol**, including handshaking, capability exchange, page transmission control, and call termination [networkworld.comnetworkworld.com](#). T.30 frames coordinate everything about the fax call: for example, they negotiate **page resolution, page size,** and **image encoding method** (MH/MR/MMR) to use, they decide the **data modem speed** for the image, and whether to use **error correction mode (ECM)** [networkworld.comnetworkworld.com](#). (ECM is an optional T.30 feature that breaks each page into blocks with checksums so that lost or corrupted data can be retransmitted, improving reliability.) T.30 control messages are sent in **half-duplex** fashion – only one side sends at a time while the other listens. Because these messages are small, the 300 bps rate is sufficient and not a major delay factor [networkworld.com](#). In fact, much of the fax speed bottleneck historically was in scanning/printing and in the image data transmission, not the T.30 signaling itself.

Once the faxes agree on parameters, they proceed to **transmit the page image data**. Here, the fax machines switch to a higher-speed **data modem** as negotiated. A variety of ITU-T **V-series modems** are used in fax: initially V.27ter (2400 or 4800 bps) and V.29 (7200/9600 bps) for early Group 3, later V.33 (12000/14400 bps) and the most common **V.17** (7200–14400 bps) for “14.4k” fax [dialogic.com](#). (In the mid-1990s, the **V.34** modem was introduced for “Super G3” fax at up to 33.6 kbps [en.wikipedia.org](#), but this only became usable when ECM and later protocol enhancements were in place to handle its fast speeds.) The sender's fax modem modulates the compressed image bits (per T.4/T.6 coding) into analog tones, and the receiver's modem demodulates them back into a bitstream to reconstruct the image. Importantly, **fax modems are half-duplex** – only one side transmits image data at a time (fax is essentially a send-and-wait

protocol, page by page). After a page is sent, the fax machines switch back to the 300 bps control channel to send end-of-page signals, check for errors (if ECM was used), and either proceed with the next page or terminate the call.

**Figure: Fax Transmission Phases.** In summary, a Group 3 fax call over analog lines goes through distinct phases (as defined in T.30): Phase A – call setup (telephone dialing, answer); Phase B – capabilities handshake (exchange DIS/DCS frames, etc. at 300 bps); Phase C – image data transmission (at high speed using modems); Phase D – post-message signaling (end-of-page, confirmations, repeat if errors, etc. at 300 bps); and Phase E – call release. These procedures were engineered to cope with analog line impairments – for example, fax protocols allow for certain noise and timing tolerances, and modems perform training sequences (like V.17's training check) to adapt to line quality. **T.30 was designed with the analog PSTN in mind**, including generous timeouts and retrains to handle noise, echo, or other issues on phone lines [soft-switch.org](http://soft-switch.org). By the end of the 1980s, this analog fax protocol stack (T.30/T.4 with V.17 or slower modems) was ubiquitous, enabling any two fax machines to communicate reliably over the telephone network.

## Challenges of Fax Transmission over IP Networks

The advent of Voice over IP (VoIP) in the late 1990s introduced new challenges for fax communication. The traditional fax protocol (T.30) assumes a **smooth, continuous analog circuit** between the endpoints [scansource.com](http://scansource.com). In a pure PSTN call, once a connection is established, data flows in real time with minimal jitter or loss. By contrast, an IP network is **packet-based** and can introduce variable delays (jitter) and packet loss. Furthermore, VoIP systems typically use **lossy compression codecs optimized for voice**, such as G.729 or GSM codecs, which **severely distort fax modem tones** and make demodulation nearly impossible [en.wikipedia.org](http://en.wikipedia.org). Even using an uncompressed codec like G.711 (64 kbps PCM) over IP is not a perfect solution – while G.711 can carry fax audio signals, the fax call then becomes vulnerable to IP problems like jitter, delay, and packet loss. A 20 ms network jitter or one lost IP packet can scramble a few dozen bits of a fax's modem data, potentially causing the fax page to fail decoding.

In essence, **T.30 was built for the analog world**, with error handling tailored to occasional analog noise, not the bursty errors and dropouts of packet networks. IP networks can also trigger timeouts; e.g., if packets are delayed, the fax machines might not receive expected signals in time and abort the call. Additionally, VoIP systems often employ **echo cancellers, silence suppression (VAD), and jitter buffers** – all of which can interfere with fax signals if not disabled. (For example, an echo

canceller might mistake a fax training tone for a residual echo and try to suppress it.) Early attempts to send fax over VoIP (“fax pass-through”) therefore required using G.711 (no compression) and turning off echo cancellation and VAD on that call. Even then, reliable faxing over a congested or long-haul IP network proved hit-or-miss. Studies in the late 90s found that **standard VoIP had a high fax failure rate** due to network impairments [en.wikipedia.org](https://en.wikipedia.org). The fax user experience suffered – many calls would drop partway or produce garbled pages.

These challenges drove the need for a fax-specific solution for IP networks. The goal was to make fax over IP **as reliable as over PSTN**, despite the packet network’s imperfections. In other words, the solution had to **mask the IP network** so that legacy fax machines could communicate without realizing they weren’t on a direct phone line [en.wikipedia.org](https://en.wikipedia.org). This need led to the development of the **T.38 Fax over IP protocol** in the late 1990s.

## Fax over IP Solutions: Store-and-Forward vs. Real-Time (T.37 vs T.38)

Two main approaches emerged for fax over IP. The first is **store-and-forward faxing**, exemplified by ITU-T **T.37**, where a fax is essentially sent as an email attachment (e.g. a TIFF image) rather than in real time [en.wikipedia.org](https://en.wikipedia.org). In T.37 mode (often called “Internet Fax” or iFax), a fax device will scan and encode the document into an image file (typically TIFF for fax), then **email** it via SMTP to a mail server which eventually delivers it to the destination (either another fax server or a fax machine that retrieves email). T.37 is **asynchronous** – not an immediate phone call – but it allows faxes to travel over the Internet without any real-time constraints. This is analogous to sending a message and letting it queue and deliver like email. Store-and-forward is useful for integrating fax with corporate email systems and getting around real-time issues, but it doesn’t satisfy scenarios where a real-time, interactive fax connection is needed (for example, many traditional fax workflows or machine-to-machine fax systems expected an immediate fax handshake).

The second approach, and the focus of this report, is **real-time fax over IP via T.38**. ITU-T **T.38**, approved in 1998, defines a method to carry fax sessions **in real time over IP networks** [en.wikipedia.org](https://en.wikipedia.org). T.38 is often called a “fax relay” protocol: it relies on gateways or devices that **intercept the T.30 fax communication and encapsulate it into IP packets** for transport to a peer device, which then presents it back to T.30 on the other side [en.wikipedia.org](https://en.wikipedia.org). In simpler terms, T.38 creates a real-time bridge for fax from one phone line, through an IP network, to

another phone line (or to an IP fax endpoint). The endpoints still run the normal T.30 protocol – they think they are talking directly via a phone line – but in the middle, the fax data is packetized and transported over the network in a way that is resilient to packet impairments.

It's useful to contrast T.38 **fax relay** with the simpler **fax pass-through** method. In fax pass-through, the fax audio (the analog tones) are merely placed into an IP voice stream (like a regular VoIP call) using G.711; essentially treating the fax like a voice call. This uses 64 kbps each direction and is sensitive to any packet loss. In **fax relay (T.38)**, by contrast, the gateways do **not** transmit the raw fax audio. Instead, they **demodulate the fax tones to recover the digital data (T.30 frames and image bits)** and send that data over IP in a structured form [networkworld.com](http://networkworld.com). On the far end, another gateway (or IP fax device) remodulates or otherwise processes the data back into the analog domain for the receiving fax machine [networkworld.com](http://networkworld.com). Because the fax is transmitted as data, T.38 can **significantly reduce the required bandwidth** and also avoid the distortions of voice codecs. For example, a fax image might be 14.4 kbps of data rather than using 64 kbps of continuous audio [en.wikipedia.org](http://en.wikipedia.org). Moreover, the T.38 protocol can add redundancy or forward error correction to recover from packet loss, something not possible with a raw audio stream. The result is a much higher success rate for faxes over IP. In fact, T.38 fax relay became the **de facto standard** for FoIP and is recommended in virtually all modern VoIP deployments when fax is needed [networkworld.com](http://networkworld.com). (Fax pass-through is still used as a fallback in some cases, but it offers no bandwidth savings and can fail more easily under adverse network conditions [networkworld.com](http://networkworld.com).)

To summarize, by the early 2000s, **T.38 Fax over IP** was widely implemented in VoIP gateways, ATAs (analog telephone adapters), and fax server software to enable reliable faxing across IP networks. We will now dive deeper into *how T.38 works technically*, including its protocol architecture, packet structure, and integration into telephony systems.

## In-Depth Technical Overview of the T.38 Protocol

### T.38 Architecture and Fax Relay Operation

At its core, T.38 defines how to transport the fax session data (T.30 signaling and fax page content) over an IP network in real time. A typical T.38 deployment involves one or more **fax gateways** (also called **Fax over IP gateways** or **T.38 gateways**) that interface between the analog phone line and the IP network. Each gateway has two sides: a **PSTN-facing side** that speaks standard T.30 over a phone line to a fax machine, and an **IP-facing side** that communicates with a peer over the packet

network using T.38. Conceptually, inside a T.38 gateway are virtual fax modems and a packetization engine. When a fax call comes in, the gateway's modem will **demodulate the incoming fax signals to extract the T.30 frames and image bits**, rather than passing them through as audio [en.wikipedia.org](http://en.wikipedia.org). The gateway then wraps those digital fax data units into T.38 packets and sends them over the IP network. On the receiving side, the process is reversed: the gateway takes T.38 packets, reconstructs the T.30 signaling and image data, and **remodulates them into analog fax tones** as if it were the sending fax machine, delivering the fax to the local device [networkworld.com](http://networkworld.com).

! [image](#)

*Figure: Architecture of a fax-over-IP system using T.38 fax relay. Two standard G3 fax machines (left and right) are connected via PSTN to **emitting** and **receiving** gateways, which translate the T.30 fax protocol into packets across an IP network. The fax endpoints are unaware of the IP network in between – the gateways make the IP transport **transparent** to them [en.wikipedia.org](http://en.wikipedia.org). Also shown (bottom right) is an “Internet-Aware Fax” (IAF) device, a fax terminal that natively supports T.38 and can communicate directly over IP without a gateway.*

In this architecture, T.38 is not a call setup or signaling protocol by itself – it needs a call control protocol like SIP, H.323, MGCP/H.248, or analog call setup to initiate the fax session [en.wikipedia.org](http://en.wikipedia.org). A common scenario is that a VoIP voice call is established first (between two gateways or between a fax machine and a fax server), usually as a voice call using SIP/H.323. The gateways monitor the call for fax tones (e.g. the calling tone CNG or the answer tone). When a fax is detected, the endpoints **switch to T.38 mode**, often via a SIP re-INVITE or H.323 message, to begin the fax relay transfer. In SIP, this negotiation is indicated in the SDP (Session Description Protocol) by offering a media stream of type “image” with T.38 transport [vocal.com](http://vocal.com). For example, the SDP might contain an `m=image` line with protocol “udptl” and a series of `a=T38Fax...` attributes describing T.38 capabilities [vocal.com](http://vocal.com). Likewise, in H.323, during H.245 capability exchange, the endpoints advertise T.38 support, and then invoke a T.38 logical channel for the fax when needed [community.cisco.com](http://community.cisco.com). The **key point is that both sides must agree to use T.38**, and this usually happens dynamically when a fax is detected mid-call, or a call can be initiated as T.38 from the start if both sides know it's a fax call (such as two fax servers communicating).

Once in T.38 mode, the gateways act as fax protocol translators. They still run the T.30 procedure with their local fax machines (end-to-end T.30 is effectively terminated at the gateways [networkworld.com](http://networkworld.com)), but between themselves they exchange fax data using T.38's frames. Importantly, T.38 transmits the fax information in a form that is **much more efficient and robust**

than raw audio. The ITU T.38 standard defines an **Internet Facsimile Protocol (IFP)**, which is the core content carried in T.38 packets. The IFP messages carry things like "fax control frame X with these bits" or "fax image data for line Y". There are generally two types of data sent:

- **Fax signaling frames:** These correspond to the T.30 HDLC frames (the 300 bps handshake messages). Instead of an actual 2100 Hz tone and a fuzzy analog HDLC waveform, the gateway decodes the HDLC and sends the exact bit sequence (and perhaps some metadata) as a T.38 packet. For example, when the calling fax sends a DIS (Digital Identification Signal) frame listing its capabilities, the emitting gateway will receive that over V.21 and then send an IFP packet containing the DIS frame bits to the other side. The remote gateway will then *reconstruct* an analog V.21 waveform of DIS to play to its fax machine. In T.38 nomenclature, these low-speed (V.21) fax protocol frames are often called **"control" or "LS" (low-speed) packets**.
- **Fax image data:** This is the bulk of the fax – the compressed image bits (per T.4/T.6) that were modulated by (say) V.17 at 14.4k. The sending gateway's modem demodulates the V.17 carrier and recovers the digital image data. These bits are then packaged into T.38 IFP data packets and sent across. The receiving gateway, upon getting these, will modulate them back into a V.17 carrier to send to its fax machine. These are often referred to as **"HS" (high-speed) packets**, since they carry the high-speed fax data.

Additionally, T.38 has provisions for sending indications of certain **tones or events** that are not easily represented as HDLC frames. For instance, the calling CNG tone or the answer CED tone can be indicated via a special message so that the far end knows those tones occurred [t38faxvoip.com](http://t38faxvoip.com). This way, even the presence of tones can be conveyed without risking them being lost over a voice path.

## Packet Format, Transport, and Redundancy

By default, T.38 uses a transport protocol called **UDPTL (UDP Transport Layer)**, which is a UDP-based protocol optimized for T.38 fax. (T.38 can also optionally operate over TCP, and newer standards even allow it over RTP in some cases, but UDPTL is the most common.) UDPTL is essentially a lightweight packet framing with a sequence number and optionally some error correction data. Each UDPTL packet carries one primary IFP data unit (the current fax data or control frame) and it can also carry one or more **redundant copies of previous data units** [t38faxvoip.com](http://t38faxvoip.com). This redundancy feature is key to overcoming UDP packet loss. T.38 allows sending, say, the last N packets' worth of data in each new packet as backup. If one packet gets lost, hopefully the next packet still has a copy of that lost data. The T.38 specification doesn't mandate a fixed number of redundant packets – it's configurable (and negotiable via SDP as

T38FaxUdpEc ) [vocal.com](http://vocal.com). Typical settings might be to send 2–3 redundant copies for critical data like control frames [t38faxvoip.com](http://t38faxvoip.com) [t38faxvoip.com](http://t38faxvoip.com). For example, a gateway might include the last two HDLC control frames in each new packet to ensure the receiver gets them even if a packet drops. The T.38 protocol distinguishes between three categories for redundancy settings: **“Indication” packets** (the tone indicators), **Low-Speed (LS) control frames** (the 300 bps T.30 frames), and **High-Speed (HS) data** (fax image data) [t38faxvoip.com](http://t38faxvoip.com) [t38faxvoip.com](http://t38faxvoip.com). Network administrators often choose to protect the control frames most heavily (since a lost handshake frame can abort the fax), whereas losing a few bits of image data might be recoverable or only cause a small image glitch (especially if ECM is on). It’s common to see, for instance, 0–1 redundancy for image data, but 2–3 redundancy for control frames, based on quality of the network.

*T.38 Example:* Suppose the fax is sending its image at 14.4 kbps. Rather than stream 64 kbps of audio, the gateway will generate T.38 packets containing the 14.4k data. These might be sent, for example, every 20 ms (the packetization interval can vary). If each packet carries 288 bytes of fax data (20 ms worth at 14400 bps) and we choose to repeat the last packet’s data once, then each UDPTL packet will contain ~288 bytes of new data + 288 bytes of the prior packet, so ~576 bytes payload. The IP overhead is minor in comparison. If a packet is lost, the receiver still gets the duplicate of that data in the next packet. In effect, T.38 trades a bit of bandwidth overhead for resilience. **Without T.38’s redundancy, even a 1% packet loss could be devastating to fax** (since a single missing packet could correspond to a chunk of a page image or a critical signal). With redundancy, T.38 can tolerate a moderate level of packet loss. It’s often said that T.38 “prefers delay to loss” – the fax can withstand some delay (since the protocol and gateways can buffer data and still satisfy T.30 timing), but it cannot easily recover from missing data without redundancy. (Some implementations also support an alternate error-correction using FEC parity packets as defined in T.38 Annex, but this is less commonly used than straight redundancy.)

**Bandwidth Considerations:** T.38 is also much more efficient than audio transport. Recall that a G3 fax modem call at 14.4k each way consumes  $64k \times 2 = 128k$  bps on a voice call [en.wikipedia.org](http://en.wikipedia.org). With T.38, only the actual fax data is sent, e.g. 14.4k in one direction (and virtually nothing in the other direction at that time, since fax is half-duplex) [en.wikipedia.org](http://en.wikipedia.org) [en.wikipedia.org](http://en.wikipedia.org). Even with overhead and redundancy, T.38 might use on the order of 20–30 kbps, which is significantly less than a G.711 pass-through call [en.wikipedia.org](http://en.wikipedia.org) [en.wikipedia.org](http://en.wikipedia.org). This can be important on bandwidth-constrained links.

## Integration with SIP, H.323, and Telephony Systems

As noted, T.38 is not a standalone calling protocol – it's typically embedded within a call setup provided by SIP, H.323, or MGCP in VoIP environments [en.wikipedia.org](http://en.wikipedia.org). In a SIP-based system, a fax call might go like this: A user dials a fax number; a normal SIP INVITE sets up an audio call (often with G.711). The fax tone is detected by the gateway or ATA, which then sends a SIP **re-INVITE** to switch the media to T.38. The re-INVITE's SDP will have a media line like `m=image 5000 udptl t38` indicating the UDP port and protocol for T.38, along with attributes such as `a=T38MaxBitrate`, `a=T38FaxVersion`, etc., to negotiate features [vocal.com/vocal.com](http://vocal.com/vocal.com). The other side (if capable, e.g. another gateway or fax server) answers with its T.38 SDP and the RTP voice stream is then closed, replaced by the UDPTL stream. From that point, the fax modems are cut through to the T.38 engines. In H.323, a similar process happens: H.323 has a mode called "H.323 Annex D – Fax" where the terminals can open a T.38 channel when fax is detected. H.323 uses an H.245 message to open a new logical channel for T.38 and close the audio channel. MGCP (used in some carrier gateways) handles fax by a call agent commanding the gateway to switch the mode. For example, there is an MGCP Fax package (as per IETF RFC 5347) that allows the call agent to request a transition to T.38 with desired parameters [rfc-editor.org](http://rfc-editor.org).

Interoperability between different vendors' T.38 equipment is generally good, since T.38 is an open standard. However, there are some **version differences** and optional features. T.38 has a concept of protocol "version" (indicated by T38FaxVersion in SDP). Version 0 was the original 1998 spec; later versions added enhancements like V.34 support. For instance, **T.38 version 3 (from 2005)** introduced support for V.34 (33.6 kbps) fax relay, enabling "Super G3" fax machines to achieve full speed over IP [qualitylogic.com/qualitylogic.com](http://qualitylogic.com/qualitylogic.com). If one gateway only supports version 0 (up to V.17 speeds) and the other supports version 3, they will usually fall back to V.17 (14.4k) operation, meaning a V.34-capable fax will downshift to 14.4k on that call [cisco.com](http://cisco.com). Most modern devices now support T.38 V3, but this is a consideration in some interop scenarios. Other parameters negotiated include maximum buffer and datagram sizes (to ensure neither side overruns the other with large packets) [vocal.com](http://vocal.com), and the method of handling the modem training (T.38 can either relay the training phase or have the gateway locally generate a training tone – this is the `T38FaxRateManagement` parameter, e.g. "transferredTCF" vs "localTCF" [vocal.com](http://vocal.com)).

A special case of integration is when a fax **server** or "Fax-over-IP endpoint" is used. For example, many enterprise fax servers or cloud fax providers have a software stack that can directly send and receive T.38, without an actual analog modem. These are often effectively "Internet-Aware Fax (IAF) devices" (as the ITU calls them) [en.wikipedia.org](http://en.wikipedia.org). An IAF can originate or terminate a fax entirely over IP, using T.38 from the get-go. In that case, no PSTN is involved on that side – the fax server's

T.38 stack takes the place of a fax machine+gateway combo. The other side could be another fax server (making it pure end-to-end IP fax), or a traditional fax reached via a gateway. T.38 is flexible enough to handle all these scenarios.

## Real-World Implementation Issues and Adaptations

While T.38 greatly improves fax reliability over IP, it is not without quirks. Real-world implementations have had to address several issues:

- **Timing and Spoofing:** Fax machines have tight timing requirements in T.30 (certain responses are expected within a few seconds). When sending over IP, especially long distances, network latency can introduce delays. Gateways sometimes implement **T.30 “timer spoofing”** – for instance, acknowledging a frame to the sender fax before the remote side has actually received it – to keep the fax from timing out. The gateway then ensures the remote eventually gets the data or handles a retry. This requires careful handling to not violate the protocol; most gateways today have optimized these mechanisms to hide IP latency up to a point. A well-known guideline is that T.38 can typically **tolerate up to about 500 ms or more of network delay** without failing, but **cannot tolerate significant packet loss** (hence the need for redundancy) [linkedin.com](#).
- **Packet Loss and Jitter:** As discussed, if redundancy is not set high enough, packet loss will kill faxes. Administrators often configure at least 2–3 redundant frames for the V.21 handshake frames [t38faxvoip.com](#). Jitter can be managed by having jitter buffers on the T.38 receiver, but too much jitter (out-of-order packets, etc.) can also cause slip-ups in the re-modulation timing. Proper QoS for fax over IP is often recommended to minimize loss and jitter.
- **Vendor Proprietary Modes:** Before T.38 was finalized, some vendors (like Cisco) had proprietary fax relay protocols. Cisco’s older **“Cisco Fax Relay”** used a method with NSE (Named Signal Events) within RTP to switch to a proprietary fax mode [community.cisco.com](#). While modern Cisco gear supports standard T.38, one might still encounter configurations or backward compatibility issues where proprietary mode is used if both sides are Cisco. Fortunately, most systems today converge on T.38 for interoperability [networkworld.com](#).
- **Fax Machines Variability:** Fax devices from different eras have slight differences in how they implement T.30 (timing tolerances, how they handle certain signals). Gateways have to be robust in handling these. For example, some fax machines send non-standard tones or have longer pauses than specified [soft-switch.orgsoft-switch.org](#). The gateway’s modem/equipment needs to accommodate such deviations (many fax DSP stacks include numerous workarounds

for various fax machine behaviors). This is analogous to how VoIP gateways had to adapt to different telephone signaling nuances in voice; in fax, it's about handling slightly off-spec implementations gracefully.

- **Error Correction and ECM:** A nuance in fax relay is how ECM (error correction mode) is handled. If both fax machines support ECM, they will use it to detect and request retransmission of flawed image data (in frames of 256 bytes). Over T.38, if the IP layer delivered everything perfectly, the fax should have no errors. But if some image data was lost and not recovered by redundancy, the receiving fax's ECM will catch the error (via CRC) and request a retransmit of that frame. T.38 gateways need to handle these retransmissions properly (storing the last image data blocks in memory until acknowledged). This works well, but if a network is very lossy, too many ECM retransmissions might exceed T.30 limits and still fail the page. In practice, a combination of T.38 redundancy and ECM makes fax extremely robust – it's rare to fail a page unless the network is really poor.
- **Transition at Call Start:** Detecting fax reliably is important. Many fax calls start as a normal ring and answer – the sending fax might play a CNG tone, or might even have a human dial and then hit "Start". If the gateways fail to detect the fax tone, the call might remain in voice mode and the fax will certainly fail. Thus, gateways often use multiple cues: the presence of a CNG tone from caller, a CED or V.21 flag pattern from answer, or even the detection of an HDLC frame in the audio (even if no tone was detected) to trigger fax mode [t38faxvoip.com](http://t38faxvoip.com). Some tricky scenarios include voice calls that switch to fax mid-call (someone on a call hears fax tones and connects a fax machine – rare but possible); good gateways will dynamically switch to T.38 in mid-call if needed.
- **Firewall/NAT Issues:** Because T.38 uses UDPTL on dynamically negotiated ports, it has similar NAT traversal issues as RTP. Ensuring that firewalls allow the UDPTL port and possibly using SIP ALG or ICE/STUN in newer systems might be necessary so that T.38 packets can flow between endpoints behind NAT. Many Session Border Controllers (SBCs) explicitly support T.38, and will anchor the T.38 stream similar to how they handle RTP, rewriting IPs/ports as needed.

Despite these challenges, real-world experience has shown that a properly configured T.38 deployment can achieve fax success rates comparable to traditional PSTN fax. Vendors have continuously improved their fax relay implementations over the last two decades, making FoIP a stable technology.

## The Current State of Fax in Modern Telecom

Even in the 2020s, fax continues to be used in many industries – but the way it is implemented is evolving. A significant trend is the move toward **cloud-based and digital fax solutions** (often branded “**eFax**” or “cloud faxing”). These services often combine the approaches discussed: they might receive faxes via T.38 or traditional lines on their servers, then deliver them to users via email or web portal (store-and-forward); or allow users to send a document via an app/email which the service then sends out as a fax to the PSTN. This hybrid approach leverages the Internet for convenience while still reaching legacy fax machines. Many organizations are retiring physical fax machines in favor of **fax servers or hosted fax services**, to integrate fax with IT systems and to eliminate the need for analog phone lines.

However, fax is *far from dead*. In sectors like healthcare, legal, finance, and government, fax is often mandated or habitually used for transmitting signed documents, prescriptions, sensitive data, etc., because it is considered a secure and legally recognized medium. Recent market research indicates that fax usage is **still growing** in certain areas – for example, the global volume of fax pages is increasing due to regulatory requirements, and fax services market is projected to grow annually in the coming years [cacm.acm.org](http://cacm.acm.org). Regulations like HIPAA in healthcare encourage fax for patient data since it can be more secure point-to-point than email (if the fax is delivered directly). Modern fax solutions therefore focus on **security and integration**, offering features like encryption (for digital storage), audit trails, and API integration (Fax-over-API).

From a technical perspective, **VoIP providers today routinely support T.38** on their analog adapters and SIP trunk offerings to ensure customers can fax reliably over IP. Many enterprise phone systems (IP PBXs) include a fax server or T.38 gateway capability, or at least allow plugging in an analog fax device via a media gateway. In the carrier space, some interconnections between telecom providers even carry fax calls as T.38 across networks to avoid transcoding issues. Where T.38 is unavailable or fails, fallback to G.711 pass-through is the backup, but with the recognition that it's less reliable.

**FoIP (Fax over IP)** is now a mature technology, and together with email-based fax and other innovations, it has helped fax adapt to the all-IP world. In summary, the history of fax protocols – from the analog Group 1/2, to the digital Group 3/4, to the IP-era T.38 – showcases an evolution aimed at making fax faster, more efficient, and compatible with modern networks. Today's telecom engineers dealing with fax must be conversant in both the legacy analog protocols (to interface with

old machines) and the IP fax protocols like T.38 (to traverse packet networks). With these tools, fax continues to fill its niche as a reliable document transmission method well into the Internet age, coexisting with newer technologies rather than being completely replaced.

## References:

- ITU-T Recommendations for fax: T.2, T.3 (Group1/2 analog fax standards – *withdrawn*), **T.4** (Group 3 image format), **T.6** (Group 4 image format), **T.30** (Group 3 session protocol), **T.37** (store-forward Internet Fax), **T.38** (real-time Fax over IP), etc.
- Technical literature and guides on fax protocols and FoIP: Dialogic and Cisco white papers, Network World Cisco Subnet series on fax [networkworld.comnetworkworld.com](http://networkworld.comnetworkworld.com), and ITU-T official standards.
- Real-world fax implementation notes from vendors (e.g. AudioCodes, Cisco) and FoIP service providers, as cited throughout this report.

---

Tags: fax, protocols, history, evolution, itu-t, group 1, group 2, group 3, telecommunications, data compression

---

## About ClearlyIP

### ClearlyIP Inc. — Company Profile (June 2025)

---

#### 1. Who they are

ClearlyIP is a privately-held unified-communications (UC) vendor headquartered in Appleton, Wisconsin, with additional offices in Canada and a globally distributed workforce. Founded in 2019 by veteran FreePBX/Asterisk contributors, the firm follows a "build-and-buy" growth strategy, combining in-house R&D with targeted acquisitions (e.g., the 2023 purchase of Voneto's EPlatform UCaaS). Its mission is to "design and develop the world's most respected VoIP brand" by delivering secure, modern, cloud-first communications that reduce cost and boost collaboration, while its vision focuses on unlocking the full potential of open-source VoIP for organisations of every size. The leadership team collectively brings more than 300 years of telecom experience.

---

#### 2. Product portfolio

- **Cloud Solutions** – Including *Clearly Cloud* (flagship UCaaS), **SIP Trunking**, **SendFax.to** cloud fax, **ClusterPBX OEM**, **Business Connect** managed cloud PBX, and **EPlatform** multitenant UCaaS. These provide fully hosted voice, video, chat and collaboration with 100+ features, per-seat licensing, geo-redundant PoPs, built-in call-recording and mobile/desktop apps.
- **On-Site Phone Systems** – Including CIP PBX appliances (FreePBX pre-installed), ClusterPBX Enterprise, and Business Connect (on-prem variant). These offer local survivability for compliance-sensitive sites; appliances start at 25 extensions and scale into HA clusters.
- **IP Phones & Softphones** – Including CIP SIP Desk-phone Series (CIP-25x/27x/28x), fully white-label branding kit, and *Clearly Anywhere* softphone (iOS, Android, desktop). Features zero-touch provisioning via Cloud Device Manager or FreePBX "Clearly Devices" module; Opus, HD-voice, BLF-rich colour LCDs.
- **VoIP Gateways** – Including Analog FXS/FXO models, VoIP Fail-Over Gateway, POTS Replacement (for copper sun-set), and 2-port T1/E1 digital gateway. These bridge legacy endpoints or PSTN circuits to SIP; fail-over models keep 911 active during WAN outages.
- **Emergency Alert Systems** – Including **CodeX** room-status dashboard, **Panic Button**, and **Silent Intercom**. This K-12-focused mass-notification suite integrates with CIP PBX or third-party FreePBX for Alyssa's-Law compliance.
- **Hospitality** – Including **ComXchange** PBX plus PMS integrations, hardware & software assurance plans. Replaces aging Mitel/NEC hotel PBXs; supports guest-room phones, 911 localisation, check-in/out APIs.
- **Device & System Management** – Including **Cloud Device Manager** and **Update Control (Mirror)**. Provides multi-vendor auto-provisioning, firmware management, and secure FreePBX mirror updates.
- **XCast Suite** – Including Hosted PBX, SIP trunking, carrier/call-centre solutions, SOHO plans, and XCL mobile app. Delivers value-oriented, high-volume VoIP from ClearlyIP's carrier network.

---

### 3. Services

- **Telecom Consulting & Custom Development** – FreePBX/Asterisk architecture reviews, mergers & acquisitions diligence, bespoke application builds and Tier-3 support.
- **Regulatory Compliance** – E911 planning plus **Kari's Law**, **Ray Baum's Act** and **Alyssa's Law** solutions; automated dispatchable location tagging.
- **STIR/SHAKEN Certificate Management** – Signing services for Originating Service Providers, helping customers combat robocalling and maintain full attestation.
- **Attestation Lookup Tool** – Free web utility to identify a telephone number's service-provider code and SHAKEN attestation rating.
- **FreePBX® Training** – Three-day administrator boot camps (remote or on-site) covering installation, security hardening and troubleshooting.

- **Partner & OEM Programs** – Wholesale SIP trunk bundles, white-label device programs, and ClusterPBX OEM licensing.
- 

#### 4. Executive management (June 2025)

- **CEO & Co-Founder: Tony Lewis** – Former CEO of Schmooze Com (FreePBX sponsor); drives vision, acquisitions and channel network.
  - **CFO & Co-Founder: Luke Duquaine** – Ex-Sangoma software engineer; oversees finance, international operations and supply-chain.
  - **CTO & Co-Founder: Bryan Walters** – Long-time Asterisk contributor; leads product security and cloud architecture.
  - **Chief Revenue Officer: Preston McNair** – 25+ years in channel development at Sangoma & Hargray; owns sales, marketing and partner success.
  - **Chief Hospitality Strategist: Doug Schwartz** – Former 360 Networks CEO; guides hotel vertical strategy and PMS integrations.
  - **Chief Business Development Officer: Bob Webb** – 30+ years telco experience (Nsight/Cellcom); cultivates ILEC/CLEC alliances for Clearly Cloud.
  - **Chief Product Officer: Corey McFadden** – Founder of Voneto; architect of EPlatform UCaaS, now shapes ClearlyIP product roadmap.
  - **VP Support Services: Lorne Gaetz** (appointed Jul 2024) – Former Sangoma FreePBX lead; builds 24x7 global support organisation.
  - **VP Channel Sales: Tracy Liu** (appointed Jun 2024) – Channel-program veteran; expands MSP/VAR ecosystem worldwide.
- 

#### 5. Differentiators

- **Open-Source DNA:** Deep roots in the FreePBX/Asterisk community allow rapid feature releases and robust interoperability.
  - **White-Label Flexibility:** Brandable phones and ClusterPBX OEM let carriers and MSPs present a fully bespoke UCaaS stack.
  - **End-to-End Stack:** From hardware endpoints to cloud, gateways and compliance services, ClearlyIP owns every layer, simplifying procurement and support.
  - **Education & Safety Focus:** Panic Button, CodeX and e911 tool-sets position the firm strongly in K-12 and public-sector markets.
- 

#### In summary

ClearlyIP delivers a comprehensive, modular UC ecosystem—cloud, on-prem and hybrid—backed by a management team with decades of open-source telephony pedigree. Its blend of carrier-grade infrastructure, white-label flexibility and vertical-specific solutions (hospitality, education, emergency-compliance) makes it a compelling option for ITSPs, MSPs and multi-site enterprises seeking modern, secure and cost-effective communications.

---

## **DISCLAIMER**

This document is provided for informational purposes only. No representations or warranties are made regarding the accuracy, completeness, or reliability of its contents. Any use of this information is at your own risk. ClearlyIP shall not be liable for any damages arising from the use of this document. This content may include material generated with assistance from artificial intelligence tools, which may contain errors or inaccuracies. Readers should verify critical information independently. All product names, trademarks, and registered trademarks mentioned are property of their respective owners and are used for identification purposes only. Use of these names does not imply endorsement. This document does not constitute professional or legal advice. For specific guidance related to your needs, please consult qualified professionals.