RSS/Email Message Lifecycle in Brief — 20050622, BOF, Düsseldorf MAAWG. mengwong@pobox.com

Alice composes the message in her client. She clicks "send". Her client submits the message to a Message Submission Agent (MSA). This upload could occur over traditional SMTP on port 25, over SMTP on port 587, or using an HTTP POST using RSS conventions.

The MSA stores the message in the user's permanently-connected Outbox. The MSA observes that Bob is a receiver of that message. It establishes a directed RSS feed for Bob and only Bob to read that message.

The MSA sends a UDP notification to Bob's permanently-connected receiver ISP. That ISP records the notification in a database: "Alice has sent mail to Bob, and the feed URL is such-and-such."

Bob's messaging client (MUA) queries his ISP, and obtains the list of new mail as an RSS feed. Bob's MUA then polls each sender in the list, and pulls down the new messages. Alice is one of those senders, so Bob's MUA pulls down her message too.

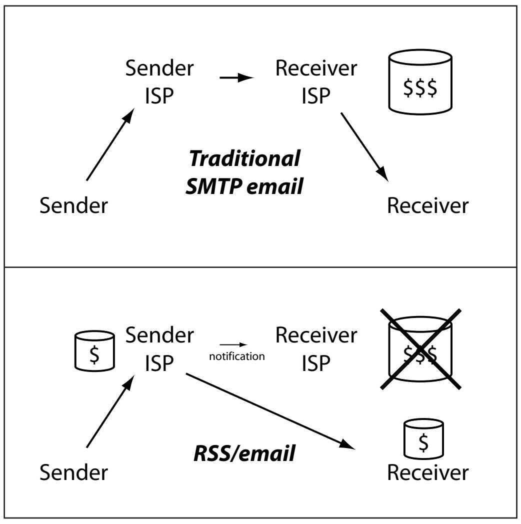
Bob's MUA sorts the messages by arrival time or whatever. To Bob, the message inbox is indistinguishable from a traditional email inbox. Bob's eyeballs read the message from Alice.

Return Receipt. As a sender, you know exactly when the receiver's MUA pulled a message you sent. If the message was never pulled, you definitely know that the message didn't hit the eyeballs.

Retraction. If you go "oops, I shouldn't have sent that" you have a chance to pull the message quickly.

Dynamic Content. It's a lot easier to program dynamically generated content into an RSS feed.

The Burden Shifts to Senders. A common refrain in the antispam community is "if only we could shift the cost of spam to the people who send it." In fact, the entire history of the antispam movement can be read as a succession of efforts to do exactly that, from postage stamps to hashcash to sender authentication to challenge response. The RSS/Email model requires senders to store messages, so you get exactly this benefit.



The Burden Shifts to the End-User. In fact, receiver ISPs don't have to store entire messages anymore; instead, they just keep a database of new mail, basically just a dirty list; the heavy lifting moves to the end-user MUA client, where CPU and bandwidth are cheaper. So we conform to the "dumb network, smart edge" model.

Bandwidth Replaces Disk. In Legacy Email, in the ideal case, a single message to 100 recipients at the same ISP goes across the network once, but repeats itself 100 times on disk. In RSS/Email, one message to 100 recipients at the same ISP goes across the network 100 times, but doesn't appear on disk at all. In an era where bandwidth is cheaper than disk, this makes sense.

No More "Over Quota" Bounces. The quota problem is an aftefact of receiver-side storage and the cost of ISP disk. In a world where bandwidth and end-user disk are cheap, it makes more sense to skip the ISP disk step entirely. And suddenly end-users are limited only by how much disk they have on their local cache devices, which nowadays is a lot.

Webmail is a problem. RSS/Email promises to bypass receiver ISPs, at least to the extent that they don't have to store messages. But webmail means the receiver ISP has to store the entire mailbox after all. This negates many of the benefits of moving the intelligence to the end-user's home machine.