DINRG - Workshop on Centralization in the Internet

Centralization is about Control, not Protocols

Henning Schulzrinne - Columbia University

Many common “consumer” applications, i.e., applications widely used by non-technical users, are now provided by a very small number of companies, even if that set of companies differ across geographic regions, or rely on a very small number of implementations even if the applications are largely standards-based. I believe that a common set of reasons other than protocol design explain this, and thus protocol design may not be the most effective way to counter this development -- unless the protocol design anticipates the problems and includes them in the design.

Here, I want to emphasize that what matters is decentralization of control, not protocol design. To take an old example, if AWS were to run S3 as a DHT, that may yield advantages of resiliency, but does not increase decentralization of cloud-based storage services, i.e., the ability of users to materially affect the design and operation of that service or to choose the lowest-price or highest-performance option among many competing offerings.

Even Systems Designed for Decentralized Operations May Centralize Operationally

Most internet applications and their protocols were designed to allow or foster decentralized operation and interconnection, with email and web being early, and blockchains a more recent example at the application layer. This also holds for consumer internet access networks at the lower layers.

However, in all cases, we are observing an increasing market concentration. At the network layer and below, core internet protocols certainly support having a dozen network alternatives for each home or business, but, at least in the United States, most households have at best two options (cable and fiber or DSL) for fixed access, and three facilities-based providers for wireless services. Indeed, existing protocols would likely support having stub autonomous systems for each of the 3,100 counties in the United States, yet the vast majority of consumers and businesses are served by about ten national providers. At the application level, most consumers experience email through Gmail, Yahoo Mail, Office 360 and similar services, even if branded as a university or corporate service. While less pronounced, web pages are increasingly likely to be hosted by SquareSpace or Google Pages, not a personal or even hosted Apache or nginx server.

Before embarking on new protocol designs, it seems helpful to understand why protocols that do not inherently favor centralization still seem to tend towards having fewer and fewer

operators. I would argue that we care about operational centralization, including the loss of user control, not the number of nodes. I state a few hypotheses below.

Users Prefer Services over Systems

The classical model of (IETF) protocol design assumed a version of the waterfall model: Standards organizations provide standards that are sufficiently detailed to allow a multitude of competing open-source and closed-source implementations to interoperate. These implementations are then acquired by operating entities that might (modestly) customize and integrate these solutions into their organizational workflows. This model obviously directly derived from the earlier telecommunications industry structure, and can even be found in physical transportation industries, with variations. (For example, standards define rail dimensions, couplers and signaling systems, with rolling stock provided to small and large railroads.)

But this model does not work well for most users. In most cases, users do not care about the systems themselves, and have neither the skills nor interest to install, host or operate these services. Thus, they rely on operators, with the usual economies of scale and scope, leading to the operational centralization mentioned earlier.

Bad Actors Encourage Centralization

An increasingly large fraction of operating systems is spent dealing with bad actors, whether that's DDOS attacks, data exfiltration, unwanted communication, undesirable content (from illegal to sufficient to drive users away) or insider attacks. Dealing with these threats is unpleasant and requires building specialized expertise and management tools. For example, not surprisingly, once managing email systems became largely synonymous with managing spam and phishing attacks, the notion of having individuals or smaller organizations manage their own email services became unattractive and many organizations delegated that task to a handful of large operators. See also "Users prefer Services".

Also, decentralized systems increase the total cost of trust. For example, most of the later-stage papers in peer-to-peer networks, offering effectively a simple keyed storage system, tried to deal with malicious nodes that either want to disrupt the system or freeload, but none of the solutions were terribly effective in practice. The design made it harder to isolate different users, as users relied on other, untrustworthy, users to perform the service. Inherently, even if data was encrypted, it provided users with the ability to observe what other users were doing. Cloud service providers offering storage solutions have a much easier time getting rid of unwanted customers - and charging deters many resource-based attacks. For these systems, a user does not depend on another user to perform tasks - they simply depend on the operator to isolate each user from the resource usage of other users, and hide their actions from other users. All such systems also consumed, in total, more energy than more centralized systems, even if the computational resource was perceived to be "free".

Bitcoin is an extreme example. Except for the purveyors of ransomware as a service, Bitcoin offers no functional advantages over credit cards or wire transfers, yet the lack of trust (and trust enforcement) requires both significant complexity and several orders of magnitude more energy for each transaction.

To prevent undesirable outcomes, from fraud to privacy breaches and robocall laws, most systems now operate in an environment that includes laws and regulations - something that early internet design efforts could largely ignore. Legal compliance can't be wished away, and often has a high fixed cost, again favoring centralization.

Almost all users expect systems to shield them from "bad" content. Federated systems make that significantly harder, as spam and robocalls amply illustrate. But almost all systems now also need to deal with more subtle content threats, whether that is "Zoom bombing" in video conferencing, harassment and child pornography in almost all human-to-human communication systems or disinformation. Thus, decentralized systems need to demonstrate that they can offer users protection against unwanted, harmful or illegal content, at reasonable cost.

Standards-Based Systems are too Complex

Interoperability has costs - in particular, interfaces need to be specified, often at great complexity. These complex interfaces have a history of not working well. While, for example, an operator could construct a multi-vendor VoIP system, most operators pick a single vendor.

The complexity has also led to the effective monoculture of implementations. Core "distributed" internet protocols like DNS, SMTP, TLS and HTTP seem to converge on two or so open-source implementations that capture 90% of traffic, plus maybe a proprietary implementation by the hyperscale providers.

Standards-Based Systems are too Simple

Yet, perversely, even as these systems gain complexity, they are often inferior in user experience to closed, non-standards-based systems. For example, there are thousands of pages of VoIP RFCs, yet they still do not offer all the control features that common commercial video conferencing systems such as Zoom or WebEx provide. Not surprisingly, most users experience video conferencing through these proprietary systems, even if components of the system rely on interoperable standards, such as media codecs and HTTP APIs.

Proprietary applications also simplify capturing value. It is easy for Zoom to restrict users to 40 minutes on their free plan and thus encourage upgrading or to have students participate in a class where only the university is paying for a license; it is much harder for a separate vendor of communications applications to get a user to try their product and then pay for both the software and the service or to allow the one-pays, lots-participate model. (For smartphones, this largely leaves only two vendors making profits on hardware and only two vendors developing operating systems.)

Summary

In short, unless the effort to decentralize protocols takes the non-technical and operational issues mentioned above into account, they are likely to lead to pseudo-solutions that are only nominally distributed, but increase complexity and the cost of operations and, if not done carefully, primarily benefit bad actors. The history of decentralized applications can offer valuable lessons as to why they have, largely, failed to achieve true decentralization of control or had side effects that makes centralization seem like the lesser evil. In many cases, it is the lack of antitrust enforcement, such as mandatory interconnection, that has led to centralization. Protocols can help here to make interoperability and data portability feasible. This may turn out to be a more realistic approach to combating the concentration of power in the internet.