Network Working Group
Internet-Draft
Intended status: Informational
Expires: 5 August 2026

E. Fjeldstrom
Independent

1 February 2026

End-to-End Semantics in a World Without Ambient Reachability
draft-fjeldstrom-revisiting-end-to-end-02

Abstract

This document examines the Internet’s current connectivity model as an observed architectural equilibrium rather than as a failure to be corrected. It argues that the deployed Internet no longer provides ambient, unsolicited end-to-end transport reachability, not because of a single design error, but as the cumulative result of rational responses to scale, security exposure, administrative autonomy, and cost containment. Default-deny boundaries and policy enforcement have become structural features of the network.

Under these conditions, applications and services that require mutual visibility rely on a recurring set of mechanisms, including relays, overlays, tunnels, rendezvous services, and long-lived outbound connections, to enable interaction. These mechanisms are widely deployed, often effective, and serve multiple legitimate roles, including constrained initiation, resource discovery, and intentional mediation or fan-out. Their persistence reflects sustained demand for controlled initiation in the absence of an explicit transport-layer admission capability.

The document treats the convergence of these mechanisms into steady-state infrastructure as diagnostic evidence. When distinct interaction roles are collapsed into a common architectural form, local correctness is preserved, but predictable system-level effects emerge, including loss of locality, concentration of load, obscured failure semantics, and elevation of coordination and authorization semantics into higher layers.

This analysis is diagnostic rather than prescriptive. It does not propose new protocols, standards, or deployment requirements, nor does it advocate restoring ambient reachability. Instead, it argues that if transport-visible end-to-end semantics (properties observable and enforceable without application payload inspection) are to remain meaningful under contemporary conditions, the architecture must provide some form of explicit, policy-bounded admission at boundaries. Any architectural responses discussed are illustrative only, intended to make these constraints concrete rather than to mandate specific mechanisms.

The intent of this document is to make the present connectivity equilibrium visible as an architectural condition in its own right, and to provide a clearer foundation for subsequent analysis, design, or standardization efforts that engage with end-to-end semantics under real operational constraints.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction, Framing and Scope

The deployed Internet is a global distributed system operating under durable security and scaling constraints. In achieving essential goals (most notably default-deny security and address reclamation), ambient inbound initiation was withdrawn. Outside-initiated access is now routinely available only to deliberately exposed services (e.g., DMZs) or within pre-established trust contexts (e.g., tunnels). These mechanisms are treated here as established features of the deployed Internet, not as failures to be corrected.

The system nevertheless continues to require mutual visibility among many classes of endpoints. In the absence of an explicit, policy-respecting mechanism for controlled initiation through boundaries, applications and services adopted compensatory techniques (relays, overlays, rendezvous, and tunneling over permitted substrates) to restore interaction. The persistence and convergence of these techniques are taken as evidence of architectural mismatch rather than misuse: when the intended entry path is unavailable, alternative paths form where interaction is needed.

This document starts from that settled reality and examines the architectural constraints it implies, rather than proposing mechanisms or revisiting historical causality.

A primary consequence of the loss of ambient inbound reachability is the widespread reliance on triangle routing. In the absence of an effective mechanism for short-term, policy-bounded admission at the transport layer, interactions that require mutual visibility are forced to route through intermediaries. These intermediaries become part of the steady-state path, introducing additional latency, load concentration, and additional failure modes.

This document characterizes explicit, policy-bounded admission as a missing architectural function. This does not imply that such a function can be universally or cleanly realized, nor that its absence reflects a design error to be corrected. Rather, it reflects the narrower claim that end-to-end transport semantics implicitly assume the ability to initiate interaction under policy, which is an assumption no longer satisfied by the deployed Internet. The continued reliance on compensatory mechanisms follows directly from this absence. Questions of feasibility, deployment, and authority are logically subsequent to identifying the missing function and are intentionally deferred.

The contribution of this analysis is not the observation that reachability is constrained, but the identification of persistent compensatory mechanisms as architectural evidence of a missing admission function, rather than as a successful or stable architectural adaptation [RFC1958]. While the models discussed are concrete enough to reason about placement, authority, and failure semantics, they are not intended to prescribe protocols, mechanisms, or deployment strategies. Any realization of the conditions described here would necessarily involve policy choices, trade-offs, and protocol design decisions that are explicitly out of scope.

This analysis is independent of the mechanism by which a boundary is implemented, the entity that operates it, or the form taken by any realization of admission. It concerns only the presence or absence of explicit, policy-bounded initiation at boundaries, and the architectural consequences of that absence in the deployed Internet.

The sections that follow analyze architectural properties and constraints implied by a default-deny Internet; they describe neither uniform current behavior nor proposed mechanisms.

2. Terminology

The following terms are used throughout this document with the meanings defined below. These definitions are intended to constrain interpretation and avoid ambiguity arising from informal or operational usage.

Admission
: The explicit act of authorizing the creation of a communication context across a boundary, subject to policy, scope, and lifetime constraints.

Admission Request
: An explicit act of requesting authorization to create a communication context across a boundary. An admission request is logically ordered before any authorization grant and may be carried by, or embedded within, other protocols or exchanges. Admission requests may be received at or mediated by the boundary, but do not themselves create data-plane reachability.

Ambient Reachability
: The condition in which endpoints are generally reachable without prior authorization or explicit admission, an assumption common in early Internet architectures but no longer typical in deployed networks.

Boundary
: A policy enforcement locus that mediates communication between administrative domains. This may be implemented by one or more devices or services and need not coincide with a single physical network hop.

Control Plane
: Mechanisms used to request, grant, modify, or revoke authorization and to establish communication contexts. Control-plane exchanges do not carry application payload data.

Data Plane
: Mechanisms used to carry application payload data once a communication context has been explicitly authorized and established.

Ingress
: The delivery of authorized application payload traffic from outside a boundary into the interior after a communication context has been explicitly established. Ingress, as used in this document, refers only to post-admission data-plane connectivity.

Interior
: The protected domain behind a boundary. Services and topology within the interior are not directly reachable without explicit authorization.

Triangle Routing
: A steady-state communication pattern in which application data is exchanged via one or more intermediaries that remain on the data path. Triangle routing may arise for multiple reasons, including constrained initiation across boundaries, the need for rendezvous or resource resolution, or intentional reflection or fan-out of traffic.

These definitions are intentionally scoped to this document and reflect architectural distinctions between authorization, control, and data delivery.

3. Characteristics of Controlled, Bounded Connections

This section states the architectural properties a controlled, bounded connection must exhibit in the deployed Internet. These properties are descriptive rather than prescriptive [RFC3234].

Some boundaries are architecturally capable of exercising explicit, coherent admission control because authority, policy definition, and enforcement are already aggregated at a single locus. Other boundaries are not, due to fragmentation of authority, delegation across administrative domains, or the absence of a coherent policy decision point. This distinction is architectural rather than technological: it concerns the alignment of responsibility and authority, not the mechanisms by which enforcement is implemented.

A controlled, bounded connection can be described in terms of the following properties:

Explicit initiation: Interaction begins with an explicit request for a specific service; authorization is not inferred from data-plane behavior.

Boundary-resident authority: Admission and authorization decisions occur at a boundary (or an authority designated by it); external parties do not unilaterally create forwarding state.

Bounded authorization: Permission to forward traffic is bounded by policy (e.g., time, scope, identity, or interaction completion) and is revocable.

Separation from transport incarnation: Authorization applies to an

interaction rather than to a particular transport binding; individual transport incarnations may fail or be replaced without implying semantic termination. This reflects existing application practice, where multiple transport connections are treated as interchangeable realizations of a single interaction, and does not introduce a distinct session layer.

Direct forwarding after authorization: Once authorization is granted, data flows using native transport forwarding; the boundary does not remain in the data path as a semantic intermediary. This does not imply that the boundary is absent from the forwarding path, only that it does not participate as a semantic intermediary or application-layer relay.

Payload opacity: Authorization does not depend on inspection of application payloads; enforcement may rely on transport-visible information but is grounded in explicit knowledge of which interactions are permitted.

A controlled-access architecture therefore requires one or more well-defined, always-available contact points used solely for admission and authorization exchange.

Because the purpose and semantics of admission requests are known in advance, non-conforming requests can be rejected early with minimal processing, and enforcement can focus on authorized interactions. For authorized interactions, inspection and shaping can focus on enforcement and resource management, operating with the assumption that the interaction itself is valid rather than attempting to infer intent from payload content.

3.1. Positive Control Versus Inferred Control

In this model, admission and enforcement operate under positive control: the boundary acts on explicit knowledge of which interactions are permitted. This contrasts with inferred control, where policy is derived indirectly from heuristics, protocol guessing, or payload interpretation. The notion of positive control is borrowed from safety-critical domains (e.g., air traffic control and industrial safety systems), where explicit clearance precedes use and ongoing activity proceeds without interpretation by the control authority. Positive control simplifies enforcement logic, narrows ambiguity, and aligns policy decisions with declared intent rather than observed artifacts.

3.2. Explicit Termination

Interactions conclude through explicit disconnect by either endpoint, or through authorization expiry as determined by policy. Transport failure alone does not define completion.

3.3. Interaction Without Explicit Admission

In the absence of an explicit, policy-respecting mechanism for initiating interaction at a boundary, systems that require mutual visibility do not cease to function. Instead, interaction is displaced into a small and recurring set of compensatory patterns. These patterns are widely deployed, often effective, and structurally necessary under current conditions.

One such pattern is Layer-7 mediation, including application proxies, relays, and protocol-specific gateways. In these cases, intermediaries assume responsibility for initiation, coordination, and authorization because no lower-layer mechanism exists to perform those functions explicitly. Where no formal admission capability is available, such relocation becomes the only means by which interaction can proceed.

When intermediaries persist in the steady-state data path rather than serving only transient setup roles, recurring structural effects are observed:

* tight coupling between intermediaries and application protocols;
* persistent participation of intermediaries in the data path;
* concentration of load and expansion of shared failure domains;
* opaque or delayed failure semantics; and
* distortion of transport-layer behavior such as congestion response and path selection.

The persistence and convergence of these effects across independent systems are diagnostic evidence of architectural mismatch rather than misuse [RFC5218]. They indicate not that intermediaries are incorrect, but that distinct interaction roles are being collapsed due to the absence of explicit admission at the transport boundary.

Other compensatory patterns arise under the same constraint. Interaction may be attempted through probing and heuristic inference, with permission inferred from partial success or failure. Where probing is unreliable or undesirable, systems often rely on inverted egress, maintaining long-lived outbound connections to preserve reachability. Where neither approach suffices, interaction is relocated to third-party mediation infrastructure.

These patterns arise for multiple legitimate reasons, including constrained initiation across boundaries, the need for resource discovery or rendezvous, and intentional reflection or fan-out of traffic. In the absence of explicit transport-layer admission, these distinct cases are frequently realized through a common architectural form, with intermediaries remaining in the steady-state path regardless of whether their role is discovery, authorization, or intentional mediation.

Tunnels and VPNs remain appropriate where the architectural goal is the extension or fusion of administrative domains under a persistent trust relationship. Their use as general-purpose reachability mechanisms, however, reflects compensatory adaptation to missing admission capability rather than alignment with native transport semantics.

Where boundaries are capable of authorizing direct transport-layer forwarding, the availability of an explicit admission capability reduces reliance on external mediation infrastructure while preserving existing compensatory mechanisms where authorization cannot be obtained. This aligns responsibility, cost, and failure semantics with the communicating endpoints rather than externalizing them onto shared intermediaries.

4. Security Scope and Placement

Security considerations in this architecture are intentionally scoped and asymmetric. Security is concentrated in the control plane, where permission is established, and deliberately minimized in the data plane, where traffic flows according to application semantics. This separation is an architectural property rather than an omission.

4.1. Where Security Exists: The Control Plane

The control plane determines whether an interaction may occur at all. It is therefore the locus for security-relevant functions, including:

* validation of requests for interaction initiation;
* authorization decisions according to local policy;
* issuance of bounded authorization artifacts and connection parameters;
* protection against abuse of the request mechanism; and
* explicit revocation and expiry of authorization.

Security mechanisms in the control plane often employ authentication, integrity protection, rate limiting, replay resistance, and auditing as appropriate to local policy. The control plane operates on explicit requests and grants; it does not infer permission from data-plane traffic.

4.2. Where Security Does Not Exist: The Data Plane

Once authorization has been granted, the data plane operates transparently. The control plane does not directly touch, proxy, or mediate the data-plane connection. The data-plane protocol, encryption, congestion control, and semantics remain those of the application or service and are outside the scope of this architecture.

While the control plane does not observe or interpret data-plane payloads, it may convey expectations to the boundary as part of the authorization response. These expectations are provided by the service and may include the anticipated transport protocol (e.g., TCP for HTTPS/2, UDP for HTTPS/3 over QUIC), port ranges, or other transport-visible characteristics. Such information enables more precise shaping and stateful packet inspection without requiring payload awareness.

This information is optional and advisory: it narrows enforcement and reduces unnecessary inspection, but it does not redefine application semantics or place the control plane in the data path. End-to-end security mechanisms remain end-to-end between the communicating parties, unchanged by this model.

4.3. Always-Available Request Contact

This section concerns an always-available control-plane contact used for admission requests. This contact does not constitute ingress and does not carry application data.

A controlled-access architecture necessarily includes one or more well-defined, always-available contact points used solely for admission and authorization exchange. These contact points exist to support controlled initiation and do not create data-plane reachability.

The presence of an always-available request contact does not introduce a new class of externally reachable surface. Contemporary networks already depend on externally reachable control and rendezvous surfaces, including HTTPS services, relay allocation endpoints, and traversal infrastructure. The architectural distinction made here is that this function is made explicit,

narrowly scoped, and semantically constrained, rather than being distributed implicitly across application-specific services and heuristics.

Because the expected behavior of admission requests is known in advance, filtering and inspection rules can be narrowly defined. Traffic that does not conform to the expected request pattern can be discarded early with minimal processing. The existence of this contact does not weaken data-plane security.

Compared to heuristic inference from data-plane traffic, such explicit request handling reduces ambiguity, limits state creation, and bounds attack surface by design rather than by interpretation.

5. Ordering Model (Non-State-Based)

This document describes ordering and validity constraints over interactions, not a protocol state machine. Ordering is an inherent property of networked interaction and distributed systems; it is not specific to the architecture discussed here.

The minimal ordering model uses the following artifacts, described only in terms of relative ordering:

Request (t(0)): A client expresses intent for a specific service.

Grant (t(1)): The service (or an authorized proxy) confirms the request and provides bounded authorization together with connection details for the next attachment.

Attachment (t(2), t(n)): One or more transport incarnations attach using details derived from a valid Grant. Failure of an incarnation does not, by itself, terminate the interaction.

Reattachment (t(n+1)): A client reattaches using a still-valid authorization and receives refreshed connection details.

Termination (t(k)): The interaction ends by explicit disconnect from the service or client, or by authorization expiry according to policy.

Here, t(x) denotes the abstract event ordering position of event x (happens-before), not wall-clock time or transport-level timeouts.

The following ordering relationships are observed:

* authorization precedes forwarding;

* transport attachments are replaceable incarnations of an interaction;
* reattachment is possible only while authorization remains valid; and
* interaction termination is explicit or policy-driven, not inferred solely from transport failure.

Transport-level events can be used by implementations to reclaim resources, but they do not define interaction lifetime or semantic completion.

This ordering model is descriptive and architectural; it is not intended to imply wire formats, message exchanges, protocol state machines, or sequencing requirements for any particular realization.

6. Privacy, Connectivity, and Tradeoffs (Non-Central Consideration)

This document necessarily touches on questions of privacy, identity exposure, and coordination, but these considerations are not its central focus. They are noted here to clarify architectural constraints rather than to engage in normative privacy debate.

On the open Internet, strong connectivity and strong privacy have never been simultaneously achievable in absolute terms. Enabling bidirectional interaction, accountability, abuse mitigation, and policy enforcement inevitably requires that some information be visible to some parties for some duration. The architectural degrees of freedom lie not in eliminating this tradeoff, but in deciding where it is paid, how explicitly it is acknowledged, how narrowly it is scoped, and how long any identifying signals persist.

Contemporary Internet architecture often resolves this tension indirectly, through inference, heuristics, and compensatory mechanisms (e.g., connection tracking, relays, long-lived outbound associations, or payload inspection). These approaches do not eliminate information exposure; instead, they redistribute it in less explicit and often less controllable ways.

Any discussion of explicit admission or authorization mechanisms therefore intersects with privacy considerations by necessity. However, this document does not propose specific mechanisms, nor does it attempt to resolve privacy tradeoffs. Its purpose is limited to observing that architectural instability frequently arises when these tradeoffs are left implicit and are instead rediscovered through layered compensation.

By making the existence of the tradeoff explicit, without prescribing how it should be resolved, this analysis aims to inform future architectural work while remaining neutral on policy, regulatory, or operational choices that lie outside its scope.

7. Conclusion

The Internet’s present connectivity equilibrium is an architectural condition shaped by necessary and rational responses to scale, security, and administrative autonomy. In withdrawing ambient inbound reachability, the deployed Internet preserved survivability but also removed the transport layer’s ability to accept short-term, explicitly-authorized ingress. The resulting reliance on relays and intermediaries is therefore not accidental, but a direct consequence of a missing admission capability at boundaries.

The central analysis here is narrow and diagnostic. Where boundaries are capable of exercising positive control, the absence of explicit, controlled admission needlessly forces steady-state triangle routing and associated costs. Introducing a bounded admission capability at such boundaries can be understood as a retrofit to the deployed architecture, not as an optimization or a universal mandate. It can be applied selectively, incrementally, and according to operator requirements, while remaining compatible with existing compensatory techniques where it is not deployed.

By separating admission from use, and by locating security decisively in the control plane rather than in data-plane inference, this architecture clarifies the roles of boundaries, endpoints, and applications. It does not prescribe mechanisms, replace protocols, or eliminate the need for intermediaries in all cases. Instead, it makes visible a missing architectural function whose absence has had observable and persistent effects on Internet connectivity.

Architectural necessity does not imply immediate deployability, universal applicability, or economic justification in all environments. Establishing necessity is logically prior to questions of feasibility. Architectural diagnosis precedes mechanism design: identifying a missing function and its consequences is logically prior to determining whether, where, or how that function should be realized under real-world constraints. Multiple realizations could satisfy the architectural constraints identified here, including ones that differ substantially from the illustrative structure used to make those constraints concrete.

The analysis establishes a clearer foundation for future work that seeks to reduce unnecessary indirection while respecting the constraints of the deployed Internet. Whether and how the described admission capability is realized remains a matter for operators, implementers, and subsequent design efforts.

8. IANA Considerations

This document has no IANA actions.

9. Security Considerations

This document does not define new security mechanisms, cryptographic algorithms, or protocol behaviors: it is architectural and diagnostic in nature.

Security-relevant aspects of the architecture (specifically the placement of authorization and enforcement at boundaries) are discussed earlier in Section 4. That discussion describes where security responsibilities are located and where they are deliberately absent; it does not impose requirements on existing security protocols or deployments.

No additional security considerations arise beyond those already present in the deployed Internet.

10. Notes on References

The references below are provided for architectural context. They are not normative and are cited to situate the discussion within existing Internet architecture literature.

11. Informative References

[MBONE] Diot, C., Levine, B., Lyles, B., Kassem, H., and D. Balensiefen, "Deployment Issues for the IP Multicast Service and Architecture", IEEE Network, Vol. 14, No. 1, January/February 2000, pp. 78-88., DOI 10.1109/65.81917, 2000, <https://doi.org/10.1109/65.81917>.

Floyd, S., Jacobson, V., Liu, C.-G., McCanne, S., and L. Zhang, "A Reliable Multicast Framework for Light-Weight Sessions and Application Level Framing", IEEE/ACM Transactions on Networking, Vol. 5, No. 6, December 1997, pp. 784-803., DOI 10.1109/90.650139, 1997, <https://doi.org/10.1109/90.650139>.

[OSI] ISO/IEC, "Information technology -- Open Systems Interconnection -- Basic Reference Model: The Basic Model", jointly developed with ITU-T, ISO/IEC 7498-1, 1994.

[SRC84] Saltzer, J. H., Reed, D. P., and D. D. Clark, "End-to-End Arguments in System Design", ACM Transactions on Computer Systems, Vol. 2, No. 4, November 1984, pp. 277-288. Revised version of a paper from the Second International Conference on Distributed Computing Systems, Paris, April 1981, DOI 10.1145/357401.357402, 1984, <https://web.mit.edu/saltzer/www/publications/endtoend/endtoend.pdf>.

[Stevens94] Stevens, W. R., "TCP/IP Illustrated, Volume 1: The Protocols", Boston: Addison-Wesley, 1st ed., ISBN 0-201-63346-9, 1994.

[TR37] ECMA International, "Framework for OSI Management", ECMA TR/37, January 1987, <https://ecma-international.org/publications-and-standards/technical-reports/ecma-tr-37/>.

[RFC1958] Carpenter, B., Ed., "Architectural Principles of the Internet", RFC 1958, DOI 10.17487/RFC1958, June 1996, <https://www.rfc-editor.org/info/rfc1958>.

[RFC3234] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and Issues", RFC 3234, DOI 10.17487/RFC3234, February 2002, <https://www.rfc-editor.org/info/rfc3234>.

[RFC5218] Thaler, D. and B. Aboba, "What Makes for a Successful Protocol?", RFC 5218, DOI 10.17487/RFC5218, July 2008, <https://www.rfc-editor.org/info/rfc5218>.

Appendix A. Clarifying Commonly Misunderstood Concepts

This appendix addresses several concepts that are frequently misunderstood or oversimplified in discussions of Internet architecture. Its purpose is to clarify why the architecture described in this document does not violate layering principles or the end-to-end argument, and why the proposed admission capability is consistent with established architectural reasoning.

A.1. End-to-End Argument as Function Placement

The end-to-end argument as articulated by Saltzer, Reed, and Clark [SRC84] concerns the appropriate placement of function, not the prohibition of intermediaries. It assumes certain environmental preconditions (most notably the availability of endpoints and the feasibility of direct interaction) and argues that functions requiring application semantics should reside at the endpoints whenever possible.

The argument does not state that all network-layer or transport-layer assistance is illegitimate. Rather, it observes that lower-layer mechanisms cannot, by themselves, guarantee correctness for functions whose meaning is defined at higher layers.

Endpoints were never fully ambient in practice. Even historically, access to an endpoint was mediated by operating system controls, service-level authorization, and gateway policy; unconditional inbound acceptance has always been an abstraction rather than an operational norm.

The admission capability discussed in this document does not implement application semantics, nor does it attempt to infer correctness. It restores a missing precondition (explicit permission to initiate interaction across boundaries) without relocating semantic function away from the endpoints.

A.1.1. Historical Context: Qualified Reachability as an Accepted Condition

By the early to mid-1990s, qualified reachability had already become an accepted operational reality. Contemporary texts and operational guidance no longer treated unqualified end-to-end reachability as a baseline assumption.

As discussed by [Stevens94] (p. 85, first edition), the successful exchange of ICMP messages does not imply transport-layer reachability, since packet filters may selectively permit or deny access based on protocol and port. This reflects the normalization of protocol- and port-qualified access control, with reachability understood to be conditional rather than ambient well before policy enforcement consolidated at modern boundaries.

Firewalls and access controls were sufficiently common that they did not require special justification, nor were they treated as exceptional deviations from Internet architecture. The end-to-end argument was therefore already being interpreted in an environment where explicit qualification of access was normal.

This historical context matters because it shows that the proposal described in this document does not introduce a new class of intermediary or violate long-standing architectural principles. Instead, it seeks to restore (at modern, consolidated boundaries) a form of qualified, transport-level admission that was still feasible in the mid-1990s when policy enforcement had not yet migrated to network transition points. In that earlier environment, direct L4-to-L4 connectivity could still be established under policy control; the architecture described here aims to re-enable that capability in a structured way that is compatible with today's boundary-centric deployment model.

A.2. The OSI Model as Methodology, Not Taxonomy

The OSI model [OSI] has often been treated as a rigid taxonomy of layers, or as a competing network protocol model. In this document, it is used instead as a theoretical reasoning framework for analyzing dependency, responsibility, and semantic containment. Its enduring value lies in its method: identifying which functions depend on which assumptions, and where responsibility for correctness must reside.

Under this interpretation, correct layering is not about placing mechanisms into enumerated slots, but about assigning responsibility at the lowest boundary capable of supporting it without semantic leakage. The controlled admission capability discussed in this document is therefore not proposed as a new "layer" or protocol, nor as a replacement for existing transport or application mechanisms. It is a restoration of a missing responsibility (explicit initiation under policy) at the boundary where that responsibility can be exercised coherently in the deployed Internet.

A.3. Compression of Upper Layers in Practice

In deployed systems, the distinction between the session, presentation, and application layers (OSI layers 5-7) is frequently compressed. Many implementations treat these layers as a single semantic domain responsible for interaction lifecycle, data representation, and application meaning.

This compression is an empirical observation, not a defect. It reflects the fact that these layers share common assumptions about endpoint semantics and interaction state. The architecture described in this document respects this practice by avoiding interference with application semantics and by locating admission outside the compressed semantic domain.

A.4. DNS as an Example of Correct Layering

The Domain Name System provides a useful example of correct layering under constraint. DNS operates above the transport layer and performs explicit signaling and delegation to restore higher-layer semantics (naming, service discovery, and reachability) without recreating lower-layer mechanics such as routing or forwarding.

In particular, the introduction of the MX record provided a deterministic mechanism for clients to locate mail servers even when direct host-to-host connectivity could not be assumed. By explicitly naming admissible service endpoints, MX records allowed clients to initiate interactions without requiring ambient reachability or probing at lower layers. This role was often compared contemporaneously to UUCP connectivity, in that both provided structured indirection when direct paths were unavailable.

Subsequent DNS extensions followed the same pattern. SRV records were introduced to provide explicit service location and port information, reducing reliance on fixed ports and trial-and-error connection attempts. TXT records have been used to convey additional service metadata and policy hints where no more specialized record type existed. These mechanisms reflect repeated attempts to supply deterministic signaling at higher layers in response to constrained or unreliable lower-layer reachability.

However, even when SRV is used to identify the correct connection point, these mechanisms still assume that the service is continuously available and willing to accept inbound interactions. They do not provide a way to obtain explicit, short-term permission to initiate transport-layer interaction across boundaries. As a result, DNS has accumulated multiple incremental extensions addressing discovery and preference, without resolving the underlying absence of controlled admission.

DNS does not violate layering by existing as an intermediary; instead, it compensates for missing global knowledge in a way that preserves separation of concerns. Similarly, controlled admission restores a missing capability (explicit permission to initiate) without entangling itself in transport or application semantics.

A.5. Partial-Layer Corrective Sufficiency

It is a common misconception that architectural repair requires completeness across all layers. In practice, restoring a single missing function at the correct boundary is often sufficient to alleviate broader systemic issues, while attempting to impose full-layer structure where it is not needed can introduce unnecessary complexity.

This observation is consistent with the guidance in [TR37], which notes that systems need not be fully specified or complete at every layer to be effective. What matters is that the correct function is restored at the appropriate point in the architecture.

In many deployed systems, lower layers are intentionally minimal. For example, adding a full network-layer abstraction on top of a simple Layer-2 bridge is neither necessary nor desirable when link-local connectivity already satisfies the operational requirements. Forcing additional layering in such cases would increase complexity without improving correctness.

The admission capability discussed here follows the same principle. It addresses a specific, missing function (explicit, controlled initiation) at the boundary, without attempting to redesign or replicate functionality that is already adequate elsewhere in the stack. Restoring the right function at the right boundary is sufficient; completeness for its own sake is not.

A.6. MBONE as Diagnostic Evidence of Structural Compensation

The Multicast Backbone (MBONE) provides a historically well-documented example of a compensatory mechanism that was architecturally correct as an experiment, yet unsustainable as steady-state infrastructure. Extensive contemporaneous analysis of MBONE deployment experience and its operational limitations is available in the multicast literature [MBONE]. MBONE demonstrates that compensatory overlay mechanisms can be correct and useful in exploratory phases, yet become structurally unstable when pressed into steady-state service.

The eventual decline of MBONE was not due to protocol defects or implementation failure. It reflected a mismatch between the layer at which compensation occurred and the layer at which authority and enforcement were required. By attempting to substitute persistent overlay coordination for a missing lower-layer control plane, MBONE reproduced the same class of layering violation identified by the end-to-end argument in other contexts.

While MBONE's decline had multiple contributing factors, including operational complexity, scaling challenges, and limited router support, its trajectory remains illustrative of a broader structural limitation: sustained overlay-based compensation becomes increasingly fragile when lower-layer authority and explicit admission are absent.

MBONE is therefore instructive not as a cautionary tale against overlays or experimentation, but as diagnostic evidence. It demonstrates that compensatory mechanisms can function correctly and usefully in the short term, yet become structurally unstable when pressed into continuous service.

Appendix B. Desire Paths and Broken Floors

This appendix introduces an informal illustrative analogy, referred to here as the "broken floor pattern", to help describe a recurring pattern observed in complex systems. The term is not intended to define a technical failure mode, architectural defect, or normative condition; rather, it serves as a framing device to aid intuition for how systems adapt when assumed structural support is partially unavailable or unreliable.

B.1. Broken Floor Pattern

B.1.1. Overview

"Broken floor pattern" is an architectural condition in layered systems in which unanticipated modifications or missing functions at a lower layer are repeatedly addressed by adding compensatory mechanisms at higher layers, rather than repairing or replacing the layer itself. Over time, these compensations accumulate, obscuring the original unanticipated modifications, and eventually make downward repair infeasible even if the underlying problem is later understood or solvable.

The name is intentionally physical: it reflects the familiar household pattern in which a modified structural floor is successively covered with linoleum, carpet, plywood, or hardwood overlays. Each new layer improves surface usability while further concealing the structural defect below. At some point, the original floor is no longer visible, reachable, or practically repairable without demolishing everything above it.

B.1.2. Defining Characteristics

A system exhibiting the broken floor pattern typically shows the following properties:

Upward-only repair path: When unanticipated modifications occur, the only viable fixes involve adding new abstractions or mechanisms above the affected layer.

Loss of downward accessibility: The original layer becomes unreachable to applications and operators, eliminating opportunities for direct repair or replacement.

Semantic displacement: Responsibilities that properly belong to the lower layer (e.g., reliability, admission, timing, routing, or flow control) are reimplemented at higher layers with altered semantics.

Accumulating dependency: Higher-layer components come to depend on the compensatory mechanisms, making their removal or bypass politically, operationally, or economically infeasible.

Diminishing reversibility: Even if consensus later emerges that the lower-layer design was flawed, restoring it would require rewriting or dismantling large portions of the system built above.

"Reversibility" here is practical rather than absolute. While direct repair of the underlying layer remains architecturally possible, accumulated dependencies, deployment scale, and continuity constraints make such repair increasingly disruptive. The pattern is intended to describe why adaptation tends to dominate in practice, not to assert that restoration is impossible in principle.

Broken floor pattern differs from ordinary technical debt. Technical debt implies that refactoring or repayment is possible with sufficient effort. Broken floor pattern indicates that the structural repair path has been sealed off by subsequent layers.

B.1.3. Relationship to Desire Paths

The broken floor pattern is the conceptual inverse of the "desire path" phenomenon:

* Desire paths emerge below an intended structure. They are created by repeated user behavior routing around misfit or inconvenience, thereby exposing misalignment between design assumptions and real use. Desire paths are diagnostic signals that repair or redesign is still possible.

* Broken floor pattern emerges above the structural modifications. Instead of working around the modifications, successive layers are added to conceal it. The misalignment is no longer visible through behavior; it is embedded into infrastructure. At this stage, repair is no longer driven by usage signals but constrained by dependency and inertia.

In architectural terms, desire paths indicate a system that is still capable of self-correction. The broken floor pattern indicates that the window for direct correction has closed.

B.1.4. Architectural Consequences

Once broken floor pattern becomes fixed, systems tend to exhibit several long-term consequences:

Persistent semantic elevation: Higher layers assume responsibility for functions that were originally intended to be provided below, reducing the expressive power and relevance of the lower layers.

Complexity accretion: Each compensatory layer introduces additional metadata, modes, heuristics, or profiles, none of which simplify or replace earlier mechanisms.

Opaque failure modes: Operational failures are masked, delayed, or reinterpreted by upper layers, making root causes difficult to observe or reason about.

Stagnation of lower layers: Innovation or improvement at the original layer yields diminishing returns because applications no longer interact with it directly.

Policy lock-in: Organizational, economic, or deployment constraints reinforce the continued use of the compensatory stack, even when superior architectural alternatives are known.

B.1.5. Diagnostic Use

Broken floor pattern is best understood as a diagnostic concept rather than a prescriptive rule. Its value lies in identifying when a system has transitioned from adaptive compensation to structural concealment.

Indicators that a system may be experiencing this include:

* Repeated proposals to "add a layer" as the primary response to unanticipated modifications.

* Increasing difficulty in explaining where authority or responsibility for a function resides.
* Claims that repairing a lower layer would be "too disruptive" despite widespread acknowledgment of its deficiencies.
* Protocols or abstractions that explicitly assume the lower layer is irreparable or irrelevant.

Recognizing broken floor pattern early is critical, because once the concealment becomes infrastructure, architectural repair requires demolition rather than refactoring.

B.1.6. Scope and Neutrality

The concept of broken floor pattern does not imply error, negligence, or irrationality on the part of system designers or operators. In many cases, the initial compensatory layers are locally rational responses to urgent constraints or missing capabilities. The pattern arises not from the original compensations, but from the long-term normalization and generalization of such compensations beyond their original scope. This framing allows broken floor pattern to be discussed without attributing blame, focusing instead on structural dynamics and their implications for future design and repair.

B.2. Summary

Broken floor pattern names a class of architectural conditions in which successive layers conceal, rather than repair, unanticipated modifications to the substrate. By the time the issues are widely recognized, the system has become structurally dependent on the concealment, rendering direct repair impractical.

As a diagnostic counterpart to desire paths, the broken floor pattern helps distinguish between systems that are still signaling misfit and those that have sealed it in. Recognizing the transition between these states is essential for understanding when architectural intervention remains possible, and when adaptation has become the dominant response and increasingly difficult under existing operational and deployment constraints.

The "broken floor pattern" is not an argument against restoration, nor a claim that earlier adaptations were mistakes. It is a diagnostic description of a system that has successfully stabilized by building around a missing capability. In such systems, change cannot proceed by rollback or exposure, but only through explicit, bounded interfaces that respect the structures now in place. The architectural response described in this document should be understood in that light: not as an attempt to remove the compensations on which the Internet now depends, but as a means of

restoring direct transport-layer interaction where policy permits, without destabilizing the equilibrium that those compensations made possible.

Author’s Address

Erik Fjeldstrom
Independent
Email: erik_fjeldstrom@yahoo.ca