Exploring the Tussle Space For Information-Centric Networking

Dirk Trossen
Computer Laboratory
University of Cambridge
Cambridge, UK
dirk.trossen@cl.cam.ac.uk

Alexandros Kostopoulos
Department of Informatics
Athens University of Economics and Business
Athens, Greece
alexkosto@aueb.gr

ABSTRACT
In a global communication system like the Internet, conflicts between different adversaries are inevitable. Such conflicts can be driven by economic as well as political interests but also by the desire of individuals to express themselves in the many forms that the Internet provides. It has long been recognized that the nature of these conflicts (or tussles) has a direct impact on the viability of various designs in general and many design decisions in particular. Such recognition plays an important part not only in today's Internet but even more so in any effort that aims at designing a future of the Internet. One such example effort is that of information-centric networking. In this paper, we look at particular aspects of such future from the viewpoint of conflicts between various parties that might unfold. We investigate how such information-centric Internet can improve on addressing such conflicts through an increased modularity of functions. We furthermore outline a first attempt for a methodology that helps us better understand certain design aspects that arise in such investigations. We present our work along a set of use cases, directly inspired by a tussle taxonomy that we lay out early on.

Keywords
Tussles, information-centric networking, system dynamics modeling, content delivery networks

1. INTRODUCTION
There has been an increasing interest to re-design the IP layer of the Internet. A particular branch of efforts, such as NDN [1], PURSUIT [2], and others, declares information a first class citizen at the networking level, building transient relationships between providers and consumers of information at any point in time. We group these efforts as *information-centric networking* in the following.

Core to these proposals is the recognition that the WHAT within a communication relation is more important than WHO is communicating. Supported by technological developments in computing and storage resources, these efforts recognize that the WHAT of a communication scenario is likely to exist in many more places than the originally addressed WHO. But we can go even further beyond this observation, by creating a link between information dissemination and realizing distributed computational tasks. We argue that available storage and computing resources within a distributed environment are utilized towards implementing such tasks. Hence, it becomes the role of an information-centric network to facilitate the dissemination of any information pertaining to the tasks, while optimizing the particular implementation of this facilitation within the realm in which the information is disseminated. This makes sub-architecture optimization a crucial aspect of information-centric networking, an issue that will be important in the work presented in this paper.

But any such radical change of the design, the provided abstractions and the resulting implementations at this core layer of the current Internet require a careful thinking as to what the potential benefits might possibly be. The authors in [3] formulate a set of desirable architectural claims that would motivate such fundamental change of today's Internet. It is not within the scope of this paper to revisit all of the presented claims made. Instead, we focus on one claim that stands out, namely that of *improving the delineation of tussles along well-defined boundaries within the resulting architecture*. This claim is based on observations made in [4], where *tussles* are defined as conflicts among stakeholders in their interests of implementing a particular function at hand. As suggested in [4], proper modularization along crucial lines of delineation within the overall architecture is essential in ensuring viability and adjustability of the architecture to varying socio-economic conditions. It is this improved ability to adjust to changes that is the essence of the tussle claim in information-centric networking.

However, the authors in [3] only provide a high-level view as to why such improvement is likely. They argue that the separation of functions for identifying information, finding it, and finally delivering it along a suitable delivery graph within an information-centric architecture is at the heart of this claim. Furthermore, the authors assert that the focus on information allows for establishing information boundaries, and therefore effectively information asymmetries, more flexibly given the exposure of information in a different, more consistent way throughout the architecture. But the lack of a deeper analysis weakens the overall message that is made here: *an information-centric networking architecture improves the ability to accommodate various constellations of stakeholder interests*.

Our work in this paper intends to provide some insight into this claim being made. For this, we utilize an approach that is driven by dedicated use cases. For each of these use cases, we outline the possible conflicts between major players as they exist in an IP-centric world and how these conflicts could play out in an information-centric alternative. We believe that this comparative approach that is based on concrete examples will aid the development of a general methodology for tussle space analysis in an architectural context. In this paper, we will sketch such methodology based on a system dynamics approach. We will also exemplify the methodology with a crucial function of our architecture, the finding of information.

Before delving into the use cases, we first provide the architectural backdrop of information-centric networking in Section 2. We then classify various conflicts in such system through a tussle taxonomy in Section 3. This taxonomy will help us better understand the specific use cases in Section 4 and 5, each of which has a specific focus within the architectural context of information-centric networking. This leads us to our attempt to formulate a methodology to better understand the tussles we encounter in Section 6, exemplified with another use case in Section 7. We end our discussion with general architectural lessons learned from our work in Session 8, before concluding our paper.

2. ARCHITECTURAL CONTEXT

Information-centric networking has been touted as a replacement to the traditional endpoint-centric IP networking approach of the current Internet. In order to enable an understanding of the tussle work that is introduced in the later sections, we first provide a brief introduction into this new architectural context that information-centric networking provides. Most of our presentation here is based on an overview given in [3], taking the liberty to omit many of the details necessary to understand the full workings of the various proposed approach but also extending in parts to better understand certain aspects that will follow in our tussle analysis.

The intuitive starting point is that all network operations shall be based on information being the primary named entity across all layers. We assert that this aids the consistency of concepts across the layers and enables efficiency gains in operating over a single concept, namely that of information, across all layers. We assume that each piece of information has a statistically unique name and that applications can request the network to deliver named information. Hence, the primary function of the network is to find an appropriate location for an information provider and deliver information rather than. In other words, the network emphasizes the WHAT of a communication scenario, while building transient relations between the WHO might have and want the information at hand. This is significantly different to IP, which places the emphasis on the exchange of opaque bits between specifically identified endpoints, i.e., it helps to locate hosts and arranges communications between them.

In order to make the vast amount of information manageable, we introduce a concept called *scope* as a way to group related data together. From the network's perspective, a scope denotes the party being responsible for locating a copy of the data. With that, it creates a point of control to implement, e.g., access control and usage policies. Each information item may reside in more than one scope. Treating a set of items as an information item itself, this allows for grouping scopes in other scopes as well. With this, the network directly operates on a (directed acyclic) graph of information with operations to manipulate these graphs. These operations follow a *publish-subscribe model*. In other words, information is published by any provider, while it is subscribed to by anybody who is interested in it. A dedicated matching process ensures that data exchange only occurs when a match in information item and scope has been made.

This intuitive introduction into information-centric networking highlights a very important aspect of changing to this paradigm of internetworking, namely the change of abstractions that are visible to applications and network nodes alike. These abstractions move from links, sockets and endpoints to information graphs with operations to manipulate these through a pub/sub model rather than a push-like send/receive model.

2.1 Conceptual High-level Architecture

This change in abstractions being exposed to application and network developers alike, the conceptual architecture changes in significant parts. In order to implement the abstractions outlined above, the architecture provides the required mapping of the underlying concepts onto concrete forwarding relations between endpoints, which are producing and consuming information. While this keeps the network architecture simple (and allows for separately optimizing the realization of parts of the network), it enables a growing complexity of application-level problems to be implemented on top of this simple model.

Figure 1 presents the main architectural components on a very high level. The *pub* and *sub* components at the application level implement applications based on basic publish-subscribe network services, enabling *publications* and *subscriptions* towards information items within particular scopes. Transactional services, operating in request-reply mode, can easily be supported through a publish-subscribe model, with the server subscribing to receive requests over identifiers being created for that purpose by the application. The relation of such new API with traditional middleware layers is that it conflates low-level information discovery as well as location determination of publishers and subscribers into a single network service. This is likely to have an impact on middleware developments, an issue left out of the discussions in this paper.

The network architecture itself consists of three main functions, *rendezvous*, *topology* and *forwarding*. Generally, the *rendezvous* function implements the matching between publishers and subscribers of information. The matching is realized for a particular part of the overall information graph that is constructed by the application. The matching is performed by at least one rendezvous point which is directly associated to the identifier of the scope that it performs the matching over. In other words, rendezvous points match the semantic-free information items within the scope they are serving. With more than one rendezvous point possible to exist for a scope, requests to information items within that scope can be routed either to all or to the 'best' rendezvous point, using anycast-like functionality. Furthermore, rendezvous points implement policies associated with the matching, such as access control.

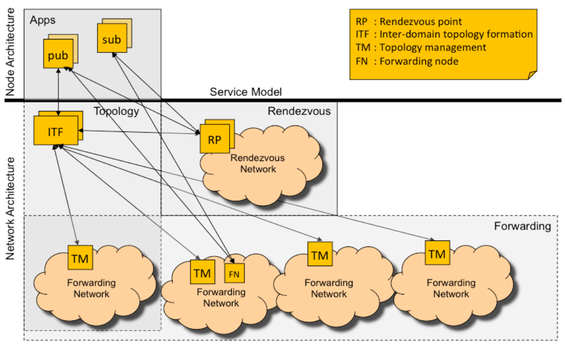


Figure 1: Conceptual Architecture

Upon having matched a publication and one or more subscriptions, an inter-domain forwarding graph is created in negotiation with the *inter-domain topology formation* (ITF) function. This is based on some form of location for the publisher and subscriber on the level of autonomous systems (ASes). Furthermore, any applicable policies as well as peering and transit relationships among ASes are included into the operation. This is similar to BGP, but the underlying networks forward information, not (opaque data) packets. Hence, there exists a rich set of policies attached to potentially every information item. Unlike BGP, this approach also allows for multiple ITF functions, each offering different sets of peering and transit opportunities that were exposed to them. This establishes the potential for peering markets in which the ITF providers serve as routing service providers. Choice can be achieved here by ASes publishing peering and transit relations to various ITF functions, usually constrained by policies governing these relations, while particular (sets of) ITF functions are chosen for topology formation. The desire to separate the tussle of (policy-based) inter-domain path selection and inter-domain forwarding requires that transit ASes cannot make additional policy-based decisions on traversing packets[1], e.g., changing the next peering hop after the path selection decision.

After constructing inter-domain paths between the forwarding networks to which publisher and subscribers are attached, intra-domain paths need to be constructed. This is done in collaboration with the AS-internal *topology management* function, which instructs its local *forwarding nodes* (FNs) to establish paths to local publishers and/or subscribers or to serve as transfer links between ASes. As in the current Internet, we do not prescribe any particular intra-domain forwarding mechanism, with the one constraint that the local mechanisms should support the traffic policies chosen by the ITF function.

2.2 Layering: A Different Internet Hourglass

Utilizing identifiers and the concept of scoping for structuring information goes further than attempting to provide application developers with a more natural way to access the network. Instead, it leads to a concept of layering that describes a new way to build up a layered architecture – defining a new Internet hourglass.

Referring to Figure 2, the concepts of (information) items within scopes are utilized above the waist to implement *scopes of discourse* through the composition of scopes. These composed scopes can be used as constraints in the pub/sub operations that act upon an information item. With this, we assert, concepts of context, scope of information reachability and other social constructs can be implemented through recursively applying a scoping operation.

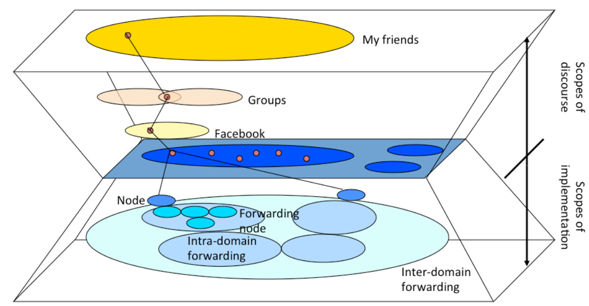


Figure 2: A New Hourglass

For instance, a high-level service such as Facebook might constitute a very large scope, exposed in the special global scope for universal reachability towards the members of Facebook. This larger scope can be further constrained as group or friend scopes, eventually limiting the reachability of the information items residing in these scopes of discourse. The reachability of the information items to given sets of users, e.g., your friends on Facebook, can be limited through realizing access control mechanisms for particular scopes. Hence, with this set of constraining scopes, various communication patterns within social networking applications can be implemented.

In another example, one can represent an organizational structure, in which a corporation is reflected in the highest scope (within the organization) with further scopes being used to constrain information to, e.g., business units, departments, groups, or individuals. It is worth noting that there is likely to exist a resolution mechanism for resolving human-readable concepts onto the *scopes of discourse* and the labeling within each of these scopes.

At the level of the waist, a new API is exposed to the application developer, which provides a higher level of abstraction where individual information items are requested through a pub/sub-like service model.

[1] We omit the details of how such separation can be enforced through new forwarding solutions that have been developed.

While we utilize the scoping concept above the waist to implement social structures through composing scopes of discourse, scoping is utilized below the waist, too, as *scopes of implementation*. Here, the discourse is that of realizing the delivery of information across actual transport networks. Scopes define the boundaries for a functional model of network functions that determine the dissemination strategy for the information items residing within a particular scope. These major functions relate to finding information, forming an appropriate delivery graph and finally delivering information along the formed graph.

As indicated in Figure 2, such boundaries can be thought of as node-internal strategies, link-local strategies, strategies within single domains, or across domains. The authors in [5], for instance, describe a node-internal implementation for an information-centric protocol stack that provides its own node-local scope for inter-process communication while providing scopes for intra- and inter-domain network functions, utilized for local forwarding, topology management, rendezvous or alike. The techniques in [6] outline an intra-domain forwarding solution, which effectively implements a series of overlapping link-local scopes within a single intra-domain scope. The information that is being disseminated is a series of packets being transmitted from a publisher (or domain ingress) node. This level of implementation is possibly several 'layers' under that of the application developer's original publication. This effectively leads to extending a high-level API that is exposed towards the application developer – we omit the details of the API and refer to more detailed technical descriptions such as in [5].

3. A TUSSLE TAXONOMY

We now turn to the various conflicts that can occur in the architectural context we outlined. We start with examples for conflicts, some of which we will deepen in our later use cases. From these examples, we then formulate a tussle taxonomy that can guide our work on exploring the tussle space for information-centric networking.

3.1 Some Examples of Potential Tussles

Tussles about what content we want and what we get: a common problem in today's Internet is the delivery of content that users do not actually want. In today's Internet, spamming has no sufficient cost and still remains a common marketing tool for most advertisers. There is a tussle between end-users and content providers that send spam e-mails, bulk messages or additional web pages that appear in users' browsers. Not only does this conflict with the users' interests, but it furthermore results in increasing congestion within networks and therefore increased costs for the delivery of the desired content. Although the information-centric architecture of Section 2 addresses this conflict by introducing a publish/subscribe service notion[2], new tussles may occur in our architecture. For instance, malicious users could send fake requests to the rendezvous system of our architecture, influencing the ranking system that is possibly implemented in the rendezvous point for the particular information. Such attacks are commonly known in today's Internet within ranking systems such as online shopping and alike. Hence, solutions to this problem need to be similar than in today's systems.

Tussles about what we need to expose in order to get what we want: Related to the issue of receiving wanted content, there is a recognized conflict that occurs when being required to reveal certain information in order to receive other. End users have become accustomed to gaining access to seemingly free content, albeit at the cost of revealing a plethora of information in the process of doing so. This is largely a conflict between end users and content providers, the latter gathering information about consumption at large-scale. Although data protection directives exist from various legislations, large-scale profiling is still considered as being in its infancy and the problems and impacts are still to be investigated. We can recognize, however, that our architecture introduces a clear control point for this conflict in the form of the rendezvous point for a particular information exchange.

Tussles about ownership of experience: who 'owns' the experience that is delivered to the end user is a conflict that widely exist in today's Internet already. Users have their delivery contract with their ISPs while often having additional agreements with content providers as well. Who 'owns' the experience here? The work in [7] has already pointed to the problems arising in such constellation of relationships and the problems that result from the separation of opaque bit transfer at IP level and the information exchange that is largely the WWW today. Our architecture in Section 2 introduces the rendezvous functionality as an intermediary between transport and end users. However, there is still a remaining tussle between the owners of actual delivery topology (represented through the topology manager function) and the owners of content (represented through the rendezvous function). Similar to today's bundled service offerings, ISPs might decide to offer rendezvous services, entering the game of brokering information in addition to delivering it. This could be countered, however, through regulatory enforcement of choice in selecting rendezvous services (similar to choosing your DNS service today). In conclusion, the conflicts are not much different but the modular boundaries, defined through the introduction of new architectural roles, could be different and therefore allow for different outcomes; something we elaborate on in our case in Section 4.

Tussles about optimizing delivery networks: related to the conflict of who owns the end user experience is that of optimizing the utilization of delivery networks. One aspect

[2] It is important to understand that content is only delivered if a receiver indicated the interest in receiving it. Hence, it is the rendezvous point that becomes the place for mediation and therefore a crucial control point in the conflict of spamming!

of this conflict is that of the role of *content delivery networks* (CDNs). CDNs are widely used in the current Internet to optimize the delivery of content. In most deployments, large content providers pay CDNs to deliver their content more efficiently and with guaranteed latencies. ISPs collaborate with CDNs in order to perform such optimized delivery. But recent initiatives such as the UK-based YouView [8] platform demonstrate the desire of ISPs to directly compete with CDN providers, such as Akamai, by replacing this overlay function with a natively supported function at ISP level. The prospect of offering lower prices for content distribution by directly exploiting the available infrastructure knowledge is what drives these efforts, albeit without a clear architectural basis for realization. Within our architecture, such ability to directly offer a service equivalent to today's CDNs is given through the exposure of a dedicated topology manager function (see Section 2.1). The boundary here lies in the interface between the rendezvous provider (representing experience requirements from the end user and content provider side) and the topology function (representing operational requirements from the ISP side).

Tussles about interconnecting networks: As part of the aforementioned optimization tussle, there is a set of particular conflicts related to interconnecting individual transport networks. One set of conflicts is that around the problem of *optimizing across administrative boundaries*, similar to proposals for inter-domain routing providers [9]. Such optimization often requires revealing operational data, such as topology information, link and router loads etc., which is seen as highly confidential by the individual ISPs. Although collaborating ISPs have an incentive to be truthful about their topology in order to have win-win situations, there could be situations in which untruthful operation is seen as beneficial, e.g., resulting in information exchange that unilaterally influences the choice of paths that are created (e.g., to shift load towards a particular ISP). We believe, however, that our information-centric aspect enables the possibility to expose the exchanged information similar to end user level content and apply similar ranking mechanisms that are used for content itself.

Another set of conflicts comes into play in the *incentive to interconnect transit and access ISPs*. With the transit ISP's business being based on the transport of content across its network, there is an obvious conflict between the desire to locally cache popular content at access ISP level (not only for cost reduction towards the transit ISP but also to maximize the user's experience in terms of reduced latency). Hence, transit ISPs lack an incentive to participate in an architectural change that is driven by an information-centric viewpoint as outlined in Section 2. This has also been recognized in [10]. However, such conflict could be decided decisively different when moving towards a transaction-based cost model, as suggested in [7], which could be enabled by the information-centric nature of the architecture as proposed in Section 2.1. Our use case in Section 5 addresses some of these conflicts.

Another interconnection tussle arises at the level of *interconnecting individual rendezvous solutions* within our architecture. The outcome of this tussle inherently influences aspects like reachability in the global information space and eventually the fragmentation of markets due to competing offerings. While interconnection incentives can be driven by economic as well as regulatory forces, desires to isolate counter these forces in areas where such isolation is required (e.g., for security reasons) or desired (e.g., for regionalization reasons). Our use case in Section 7 addresses some of these aspects.

Other examples, being left out for reasons of space, address issues of who defines identifiers as well as the structure for information (i.e., the structure of scopes in Section 2.2) as well as who ensures a trustworthy execution of various functions. While our following taxonomy lists some of these particular conflicts, it is clear that only deeper elaboration and study in future work can shed more light on these important issues.

3.2 A First Estimation for a Tussle Taxonomy

Before elaborating on some of our examples in more detail throughout the following section, we first formulate a first estimation for a taxonomy of tussles that can be utilized for a systematic investigation of the larger tussle space.

Figure 3 presents the various categories of tussles that we identified. It can be seen that the categories are not mutually exclusive, e.g., security tussles related to information overlap with tussles in the information category with the latter being more concerned economic aspects of our information-centric perspective. We can see that the inner categories are all encompassed by the larger socio-economic tussle category that is concerned with the establishment as well as intervention of markets (the intervention driven by various socio-economic players).

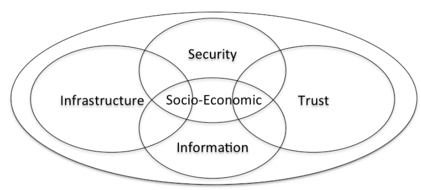


Figure 3: Tussle Categories

Table I elaborates on our tussle taxonomy. It outlines the likely involved actors as well as the architectural functions affected in some form or another. What is missing from this taxonomy is the particular remedy that our architectural approach provides in accommodating the tussles in each particular category. This is left for our tussle space exploration. We furthermore return to architectural lessons learned from our work in Section 8.

4. USE CASE: ACCESS PROVISIONING

We now turn to use cases for conflicts, investigating how our information-centric vision of an Internet architecture could affect the business models of existing actors. Our first example is that of current ISPs and their core business of providing Internet access to end users and business alike.

Category	Aspects of Conflicts	Actors Involved	Architectural Functions Affected
Security	*Infrastructure security*: who makes routing decisions? Who can define requirements affecting infrastructure security (such as path choices, load, …)? *Information security*: Payload encryption and key management (e.g., self-certified vs centralized), Governance of identifier space (e.g., long-lived vs short-lived identifiers), Governance of information structures (e.g., changes in structure to avoid profiling) *Accountability*: conflict between accountability and privacy of actions as well as content	ISPs, content providers, rendezvous providers, key providers, end users, regulators	Rendezvous, topology management, key management for identifier space, network attachment at end nodes
Trust	*Trust in functions*: policy-compliant execution, isolation of functions possibly misbehaving *Trust in information*: provenance, confidentiality	ISPs, content providers, rendezvous providers, end users, regulators	All architectural functions
Information	*Information governance*: Governance of identifier space as well as ownership of the defined information space *Brokering information*: policies for matching interests and availability, aspects of profiling usage and consumption for the benefit of, e.g., advertisement	ISPs, content providers, rendezvous providers, end users, regulators	Rendezvous, key management for identifier space
Infrastructure	*Brokering topological capabilities*: exposure of infrastructure information for optimized resource usage within and across networks *Delivering bits*: delivery of individual information items that is compliant to some agreed policy during route selection	Tier1 ISPs, access ISPs, end users, regulators	Topology management, forwarding
Socio-Economic	*Establishing flexible information asymmetries*: flexible exposure of stakeholder requirements, such as QoS or path selection, and association of pricing regimes with each with the ultimate goal to establish an information asymmetry that results in a market structure. *Defining functional boundaries*: Definition of modular boundaries along which to execute functions, including the enforcement of such boundaries through technological, market and regulatory means	All actors in the ecosystem	All architectural functions

Table I: Tussle Taxonomy

In today's Internet, there is a business relationship between end-users and Internet Service Providers (ISPs). End-users usually pay a fixed price for connectivity in ISPs' access networks. Alternative Service Level Agreements (SLAs) have also been proposed, such as volume or congestion-based charging [11]. However, fixed-pricing for connectivity seems to be the most common pricing scheme that ISPs employ. Often, ISPs offer end-users a bundled service of connectivity as well as DNS services. Although, alternative DNS services are offered (e.g., Google Public DNS [12], OpensDNS [13]), the majority of the end-users choose the default DNS service of their own ISP.

In an information-centric architecture, we can identify two new stakeholders: the *Rendezvous Networks* (RENE) and their individual rendezvous points, as well as the *Topology Managers* / *Internet Topology Formation* (ITFs). Moreover, there are two different types of end users: *Publishers*[3] and *Subscribers*. The Rendezvous Network serves as a platform where end users subscribe for as well as publish an information item. The RENE is a federation of *brokers* who owns the necessary information about the 'demand' and the 'supply' of the information items[4].

[3] Caches can be seen as alternative publishers for the same content.

[4] We observe that today's DNS is similar to the rendezvous functionality within our architecture.

This market structure seems very similar to a two-sided market. An economic network is a two-sided market if (a) there are two distinct groups of customers; (b) the value obtained by one kind of customers increases with the number of the other kind of customers; and (c) an intermediary is necessary for internalizing the externalities created by one group for the other group.

In two-sided markets, there are traditionally three types of stakeholders: two end users groups and an intermediate platform. In the current Internet's market, we identify the individual end users and content providers (the two end users group), as well as the ISP (the platform).

It should be noted here that the role of network externalities is also of high importance. In particular, we can distinguish between two main sets of externalities in a two-sided market: The *usage externality* results from the interaction between two different user groups, whereas the *membership externality* refers to the installed base [14].

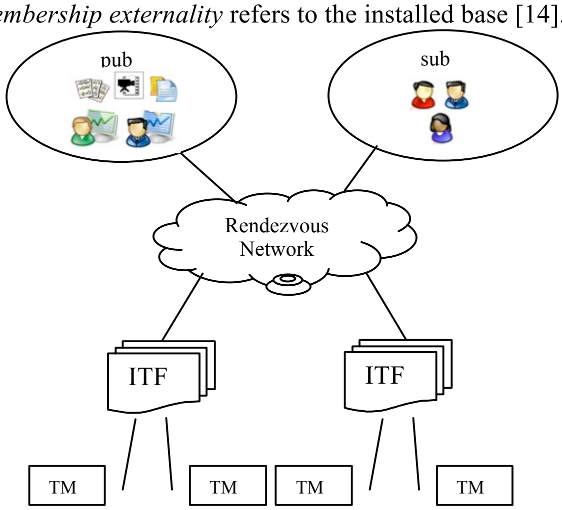


Figure 4: A Three-Sided Market

The presence of externalities and the existence of two different prices raise the issue of price allocation. Since there are two different user groups, ISPs face two distinct types of demand. So, the price structure will reflect the demand elasticity and externalities (in order to get both sides on board), as well as platform competition. Thus, the final end price is composed of a price paid by the web site and a price paid by Internet users.

However, we observe that the above example within today's Internet is quite different from our information-centric approach. In the previous example, the platform (namely the ISP) owns the network and charges based on the operating cost. The main difference lies in the new stakeholder; the Topology Manager who owns the network. The role of the Rendezvous Network could be seen as a broker between end users and topology managers. Thus, we can see the above case as a *three-sided market* (Figure 4). The Rendezvous Network could offer a fixed-fee SLA to publishers and subscribers for the connectivity service. Additionally, end-users could pay extra usage-based fees and these fees (or a proportion of them) could 'pass through' the ITFs via the RENEs.

We argue that our information-centric approach might possibly enable new market mechanisms, based on the observation made in [7]. The authors in this paper assert that the current Internet does not provide sufficient economic mechanisms for stakeholders to express preferences. This is largely due to the fundamental separation within the IP architecture between opaque bit transfer and information exchange on upper layers[5]. It is this separation that makes the establishment of pricing regimes an expensive solution, requiring out-of-band signaling solutions, which are largely limited to few services only. Instead, the authors propose to flexibly establish information asymmetries through utilizing implicit knowledge about the transferred information structures (without a need for expensive and error-prone deep packet inspection solutions). This could lead to an accountability framework for resource utilization that spans applications and networks.

In such framework, subscribers could express their preference about the QoS for receiving a specific information item. Potential preferences could be giving higher or lower priority for the subscribed information item, receiving a file from a specific publisher, paying the lower price for an information item, etc. Respectively, publishers could also express their preferences about the QoS of their delivered information item, such as publishing the same content to different rendezvous points and have different SLAs with them - classes of services, etc.

With this approach, the various functions in the network could have all the necessary information about the 'demand' and 'supply' of the available information items without the need for explicit signaling framework as in today's IP networks. Such demand/supply information could play a role in the establishment of final delivery graphs throughout the network. For instance, the final matching decision within the rendezvous point could be based on the feedback that inter-domain topology formation functions will provide to the rendezvous point about potential paths, their utilization, their metric of resilience, etc. Auction mechanisms could be applied, where bids will determine which ITF will be chosen by the rendezvous point for a specific data transmission. This could lead in a new different type of competition games between the providers for inter-domain connectivity.

5. USE CASE: CONTENT DELIVERY

In this section, we take a closer look at the most common business models in Content Distribution Networks (CDN) markets. In particular, we provide a brief analysis of the status in current Internet. Furthermore, we investigate how

---

[5] The authors compare the resulting pricing regimes to that of paying the truck driver in the supermarket rather than being billed for a complete product at the checkout.

such business model could be affected or changed in an information-centric architecture.

Generally, CDNs perform content replication in order to efficiently distribute it close to end-users. To achieve this, CDNs globally locate surrogate servers with cached content to improve accessibility and lessen the load of the origin servers [15]. Additionally, CDN solutions result in reducing congestion within a network, as well as the need for capacity expansion investments. From a technical angle, CDN overlays usually perform DNS dispatching, URL dynamic re-writing and HTTP redirection. Such request-routing mechanisms are used to forward the requests to cached servers that are close to the end-users [16].

Let us investigate the most common classes of CDNs. The first category is the *content-centric* model. Here, the CDNs' operation is driven by the needs of the content owners. In particular, content providers pay CDNs to host and serve their content under the promise of lower latencies and generally increased QoS.

Another category of CDNs is the *access-centric* model. This type of CDN is driven by the needs of Internet access service providers. Here, the money flow is different from the content-centic model, since an ISP pays CDN to serve popular content from caches close to its customers. Access centric CDNs can be distinguished in two separate subgroups. The *Internal Service Provider CDNs* aim at reducing the internal bandwidth load within an ISP's network by performing caching. They are mainly used to enhance service differentiation and premium content distribution (e.g., for IPTV, VoIP, etc). On the other hand, *External Service Provider CDNs* perform caching in order to reduce the external bandwidth cost for ISPs.

An obvious problem in both models is the usual network agnostic nature of CDNs since they do not own network infrastructure. They have limited control over the network in which they operate, and the only available information they have is the IP address of a specific request. Hence, based on the proximity of the source address, CDNs redirect the request to the 'best' server. Such criteria could be different in each CDN, e.g., latency, locality, etc.

However, utilizing network proximity is not always the right approach. First, this could result in a higher DNS request rate. Moreover, DNS requests by a CDN do not always provide the right information about a user's network location [17], e.g., in the case of a cache miss.

Due to this recognition, new trends appear in CDN markets [18]. One such trend is the establishment of CDNs directly through network owners, offering bundled services that take advantage of combining access provisioning and content hosting. Such advantage arise from having the necessary knowledge about traffic bottlenecks and not being dependent on bandwidth from third-party carriers. A similar approach for peer-assisted content distribution is the insertion of entities equipped with high resources (in terms of bandwidth and storage) that are controlled and managed by the ISP, called ISP-owned Peers (IoPs) [19].

Another trend in CDN markets is the cooperation of different stakeholders, such as hosting providers, access service providers and backbone carriers (e.g., Inktomi's Content Bridge Alliance [20]). The role of access network providers is of high importance in such federation solutions, since the access network is a critical quality bottleneck for end users.

On the other hand, the Content Delivery Network Interconnection (CDNI) efforts [21] propose to interconnect separate CDNs, supporting end-to-end content delivery while dynamically expanding the delivery footprint. However, this approach has similar locality disadvantages like current CDNs[6].

Let us now investigate how these business models could be affected or changed in an information-centric architecture. Firstly, the notion of content delivery is central to an information-centric architecture so that the role of a CDN can be assumed by anybody in the network who might have the requested information available. Hence, caching becomes a natural part of the network's operation.

This can be supported in ISP-driven models in which the topology information within the ISP's topology manager is utilized to determine 'nearby' caches as potential publishers for information requested by subscribers. This equates the access-centric model in today's Internet (albeit with the possibility to utilize more of the ubiquitous resources that are available today on standard, even user-managed equipment). The role of today's CDN providers would be downgraded towards a mere (storage) resource provider; a role that could easily be assumed by a massive storage (cloud) provider.

But there is also the equivalent to the aforementioned content-centric model, where the content provider aims at increasing end-users experience. This approach is enabled by the rendezvous function taking the main role in the decision making over publisher selection, albeit in collaboration with the transport network providers. It is clear that the selection considerations here are mainly driven by QoE (quality of experience) criteria, while the access-centric model is QoS-driven. But similar to the access-centric model, the role of traditional CDN providers would be that of a mere storage provider.

From this analysis, we can derive a very important, potential impact on today's CDN market:

> *The non-storage related functionality of traditional CDNs will be inherently implemented by the functions of topology management and rendezvous in our*

[6] Locality issues arise when HTTP re-directs end up in parts of the CDN that are more associated with the business association of the end user (e.g., his home ISP) than his actual physical location. This could occur, for instance, when travelling or utilizing corporate VPN solutions.

architecture[7]*, while the remaining storage provisioning can either be realized by managed storage providers (such as Amazon or other cloud storage providers) or through user-managed resources.*

5.1 Scenario 1: Minimize Congestion

In our first scenario, we assume that there are two ISPs that have a peering Service Level Agreement (SLA). Both ISPs are connected to a higher Tier ISP. Moreover, we assume that both ISP's topology managers (TMs) interact.

In the current Internet, ISPs have no information about what 'kind' of traffic traverses their network. Techniques like DPI (deep packet inspection) are utilized to gain insight about the characteristics of their traffic (e.g., peer-to-peer or real-time traffic). However, DPI boxes cannot capture the entirety of the traffic due to, for instance, encrypted packets.

On the other hand, in an information-centric architecture, such information is available within the network and this could be a useful input for the topology managers. Let us illustrate this issue. In Figure 5, we can see a publisher (P_1) in ISP_1, who has available a very popular information item. In ISP_2, there are three subscribers (S_1, S_2, and S_3 respectively) that have subscribed to this information item.

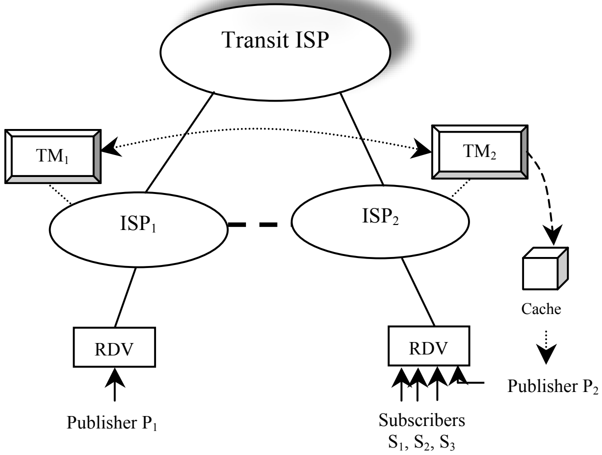


Figure 5: Using ICN to Minimize Congestion

Since, both ISPs have the available information about the publishers and subscribers for a specific information item, the topology managers can interact for the purpose of caching and hence reduce the exchanged traffic between their networks. In particular, we observe that there are many subscribers in ISP_2 for an information item located in ISP_1. Since, both ISPs have a peering relationship (so ISP_2 will not be charged by ISP_1 for the received traffic), TM_1 could inform TM_2 to cache the specific information item. Thus, the cache server of ISP_2 could now act as a new publisher (P_2) in ISP_2. As a result, the new subscribers of this information item in ISP_2 will receive it with increased QoS. Moreover, both ISPs will reduce their traffic within their network (no traffic for ISP_1 and less traffic for ISP_2). Hence, both ISPs have an incentive to cooperate.

5.2 Scenario 2: Minimize Transit Costs

In this scenario, we again assume two ISPs with a peering relationship. There is also another ISP (ISP_3), which has no peering agreement with ISP_1 and ISP_2. All three are connected to a higher tier ISP.

In Figure 6, we assume publisher P_1 having a very popular information item. In ISP_1 and ISP_2, there are a number of subscribers that are interested in this publication. However, neither ISP_1 nor ISP_2 has a peering relationship with ISP_3. Consequently, the traffic for the specific information item is passed through the transit ISP, resulting in increased transit costs for both ISPs.

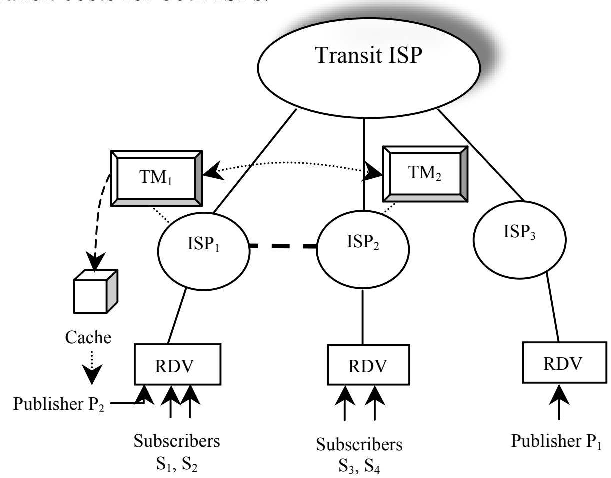


Figure 6: Minimizing ISP's transit costs

In the information-centric architecture case, ISP_1 and ISP_2 can implement their incentive to cooperate, in order to avoid additional charging by the transit ISP, by having either ISP_1 or ISP_2 cache this information item in its server. This cache could act again as a new publisher (P_2) for the same information item. In this case, both ISPs will avoid additional charges by the transit ISPs. Moreover, ISP_3 will lower potential congestion within its network.

6. TOWARDS A METHODOLOGY

In our goal to explore the tussle space in a large-scale design, we can recognize a few foci that appear across our use case. Firstly, it is the notion of an *actor* within the system. Secondly, this actor acts as a selfish adversary, driven by its own *incentives* and *strategies* to optimize the outcome of what could be outlined as a game. Thirdly, the strategies are influenced by the combined *causalities* that drive the overall behavior of the system and therefore the validity of the individual strategies. A successful participation of an actor in the overall system can be measured by the system's ability to accommodate the individual actors ability to implement their strategies towards their individual goals of participation. Formulated from a system design perspective, *a viable system design*

[7] In other words, the complexity of CDNs to implement the plethora of re-directions necessary to end up in the appropriate cache is largely implemented by core functions of the network architecture – a fundamental difference to today's IP world!

can be defined as one that is able to accommodate the implementation of competing (and changing) strategies, and therefore to accommodate the various tussles that arise from this implementation of strategies.

In the following section, we want to outline our early work on a methodology that emphasizes the understanding of these various causalities and their impact on crucial system design characteristics. For this, it is crucial to identify and describe the causalities that underlie the various tussles that arise from competing strategies of the actors within the system. Our goal within this methodology is to enable that formulation and quantification of the likelihood of possible outcomes under the influence of various parameters for the identified causalities.

6.1 A Primer in System Dynamics Modeling

A core tool in our quest for a methodology is that of *systems dynamics (SD) modeling* [22][8]. SD modeling captures the various dynamics that occur within a (large-scale) system. Its furthermore graphically describes the causalities driving these dynamics, more intuitively leading to an understanding as to what causalities drive the possible outcomes of the system. In addition to graphical representations, these causal structures can be turned into an analytical model in the form of time-varying differential equations. Hence, SD modeling provides the right starting point for our methodology in terms of capturing intuition as well as providing rigor for analysis.

Let us provide a brief introduction into some basics for SD modeling. At the heart of SD modeling is the notion of *feedback processes* that describe the dynamics within the system. It is the interactions between the various feedback processes that influence the overall system behavior over time. Within a defined *problem*, the *stock* defines the state of the system that best describes the underlying problem. The rate of change is defined as the *flow*, usually existing as an inflow and outflow. The causalities that influence these key variables of the system are captured as negative or positive feedback loops. Not only the causalities towards the stock and flow, but also the influences towards *auxiliary variables* in the system, are captured. Such notion of feedback loops lends itself to graphical depiction and therefore represents the qualitative part of the SD method.

An important step in creating the causal structure itself is that of developing the so-called *reference mode* of the system – that is the expected time-dependent behavior in an idealized form. Typical expected behaviors are linear or exponential growth (or decay) as well as hyperbolic curves [22] or combinations of them. It is an important step to outline the reference mode since it captures the modeler's understanding as to how the system is expected to behave, based on the currently captured feedback that the various actors provided into developing the model. Hence, a drastic divergence from this expectation is an important input into refining the model. It is an apparent signal that the seemingly understood system behavior is not reflected in the actual, simulated, one. Another important step is the development of *scenarios*, defining the various parameter sets that are used in the simulations. Putting all these pieces together results in a parameterized analytical model for a given problem, in which the varying outcomes depend on scenario-based input.

[8] We realize that tools for dynamical systems and feedback systems generally fall in the category of methods that capture large-scale system behavior.

6.2 Methodology

The applicability of SD modeling to system-level problems motivates the integration of this method into a methodology for understanding tussles that unfold in large-scale systems.

We can differentiate two different foci that are important for the viability of a design. The first one is that of a *market focus* in which we aim at understanding the various market outcomes enabled or inhibited by a particular design under a set of possible development scenarios. The second one is that of a *design focus*, where the strategies are in the forefront that would make particular design choices successful or fail in the presence of various socio-economic influences. In this paper, we focus on the former focus, i.e., the market outcomes, while the latter is omitted here, nevertheless being an important tool for a system designer.

Figure 7 presents the steps of our methodology. We first identify important *design characteristics* within the design space that is being considered. While such design characteristics may emerge from architectural discussions, characterizing their time-dependent behavior is where SD modeling will be of help.

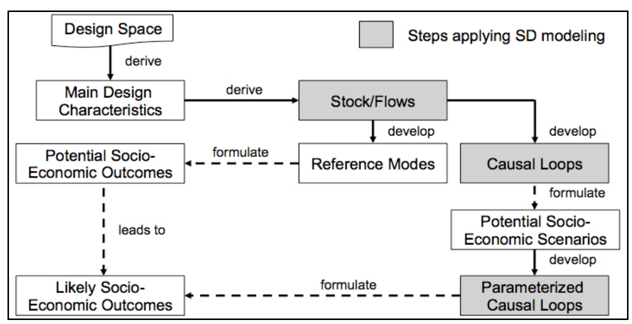


Figure 7: Methodology

For instance, design characteristics may be reflected as the number of players for a particular function, the degree of collaboration between particular players, or anything else that represents a particular characteristic of interest within the design space. Selecting the design characteristics is an important but also very subjective step that depends on the angle of the evaluation. It is therefore crucial to record the reasoning behind the choices.

This first step is connected to SD modeling by formulating the design characteristics as system dynamics problems, i.e., representing them as *stocks* and *flows*. For each model, a *reference* mode is developed, outlining the possible behavior of the modeled system. This allows for outlining potential *socio-economic outcomes* that are enabled (or prohibited) by particular design choices. In order to understand the likelihood of these outcomes, various influences are captured along multiple socio-economic dimensions, ranging from user behavior over business strategies to regulation.

Capturing these influences is usually done through desk research or interviews with various parties, such as regulators, incumbents, investors, or end users. The uncovered causalities are modeled as *causal loop* diagrams. This results in a SD model for each problem in which each auxiliary variable is defined through an analytical expression of the causality that it represents.

After having captured the causality models, we formulate relevant *socio-economic scenarios* under which the design choices are to be evaluated. These scenarios allow for parameterizing the variables, providing the parameter sets for running simulations of the developed models by solving the equations that underlie our individual SD models. These simulations lead us to the *set of likely socio-economic outcomes* as a subset of the possible outcomes under this given set of scenarios. This subset leads to making statements on the various markets being created.

Viability of system designs is clearly not an issue that can only be evaluated (and decided) at design time. The limited insight that a single one-shot capturing of conflicts can provide as well as the entrance of actors into the system after its original deployment require that the understanding of the system's viability must evolve. For that to happen, however, the understanding must be *codified* in a manner that allows re-visiting the thought processes (and models) of the past whenever necessary. Only such codification allows for capturing evolving trends, understanding the entrance of new players, the development of new business models, the introduction of new usage models, changes in regulatory landscape and other issues that require a constantly evolving perspective onto the viability of the system under these changing constellations.

The methodology in Figure 7, together with the resulting SD models, provides such codification of the wider socio-economic causalities as the basis for a continuous evaluation. In other words, when moving from design to runtime, we see the role of the methodology as that of *capturing evolving tussles*, including the ones that were not identified at design time. This leads to a repeatedly refined model, which could result in changes to the original design (which is in turn incorporated into a refined model etc).

6.3 Codification Through Tool Support

Codification cannot happen through a mere methodology; it requires tools for capturing the knowledge gathered along the way. This is even more important due to our ambition to support a constant evolution through a refined understanding of the design space. Hence, a strong focus in our work lies on developing design tools that capture the various steps in our methodology.

Developing an SD model requires the model developer to understand the problem space from many angles, usually through a series of interviews with various stakeholders and experts. With these interviews, the relevant causalities and influences need to be uncovered. Since setting up interviews with various experts is difficult and time-consuming, it is crucial for the modeler to capture the essential information. Hence, it is imperative to provide a tool for the modeler that supports the recording of findings, interviews, anecdotes and desk research in a coherent and guided way. In addition, such recordings might also help in the actual engagement with the various experts through graphical visualization of already captured information that can be utilized in future dialogues.

The result of these considerations is our toolkit that captures the findings of the various steps in our methodology. In order to aid engagement with the various parties, we utilize *mind mapping* techniques, complemented with the ability to add comprehensive notes, including multimedia annotations (e.g., recordings of interviews). The toolkit is implemented with the open source software XMind [23]. As part of our efforts, we plan to release the toolkit under an open source license.

Figure 8 outlines the steps of our toolkit. Behind each leaf in the mind map, we implement a separate sheet (indicated by the small "**c**" symbols in Figure 8), with more specific representations for the particular step[9]. Each mind map can be extended according to the instructions.

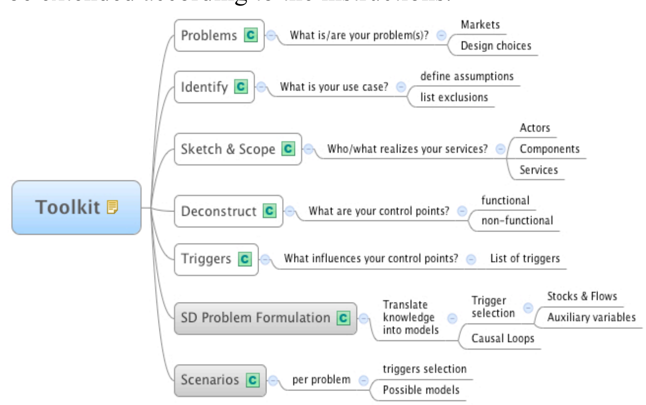


Figure 8: Steps of Our Toolkit

The steps in the toolkit start with specifying the particular focus of the *problem*, formulated as specific questions that directly relate to the design in question. The *identification* of the use case aids the designer in focusing on a particular part of the architecture. The use case also captures the assumptions being made, often for simplification of the problem space. The *Sketch & Scope* step captures the various actors, components, and services that are required to implement the desired design. This step helps identifying the *functional control points* of the socio-economic environment, i.e., the ones directly implementing the

[9] For space reasons, we omit these specific sheets.

technical design. A *control point* here is defined as a point in the environment where control of some sort can be applied, e.g., through centralizing a particular component or enforcing a particular regulatory requirement. This step is extended by the *Deconstruct* step, which lists all control points within the socio-economic environment, extending the functional ones from the previous step. The next step captures the influences on these control points. These *triggers*, like control points, are divided into socio-economic categories that range from user behavior over regulation and business strategy to technology.

With the help of the toolkit, the case developer can now translate the gathered knowledge into SD models. For that, a return to the *Problem* step helps selecting the appropriate triggers to formulate an initial set of models that represents the formulated design problems (following our methodology in Figure 7). This step also identifies the auxiliary variables for an initial set of causal loop diagrams. The development of the causal loop diagrams is done separately, based on the information captured via the toolkit. For our purposes, we use a commercial tool, resulting in an interaction between the SD modeling and the recordings captured by the toolkit. The SD tool allows for a graphical representation of stock and flow models with causal loops, each of which has an underlying set of equations for each set of auxiliary variables. The tool further allows for manipulation of exogenous factors based on text file input or in real-time.

It is this combination of codified knowledge within our toolkit and the development of SD models that not only enables our methodology at design time, but also allow for repeatedly evolving the understanding of the system's viability over time. This is achieved through re-visiting the recorded knowledge, allowing for refinements in secondary modeling steps, e.g., after the initial desk research has been performed or parts of the socio-economic environment have evolved beyond the initial setting.

7. Exemplifying our Methodology

Let us now return to another use case within our architectural context. This use case is to exemplify our methodology that we introduced in the previous sections.

For our example, we focus on the function of global rendezvous (see Section 2.1), i.e., the discovery of rendezvous brokers for information that needs to be globally available. We first outline the main characteristics for an architectural solution before presenting our modeling work in this space.

7.1 Structure of An Architectural Solution

Large-scale rendezvous comes with many faces and for many purposes. The main structure, however, is similar throughout most if not all solutions. Firstly, rendezvous is usually implemented in a tiered manner. In other words, a request is sent to a well-known local entity for resolution (tier 3), the local *rendezvous point* (RP). If the request cannot be resolved, it is forwarded to a local federation, the local *rendezvous network* (tier 2), which usually represents some form of administrative boundary, such as a corporate environment, an administrative network, or similar. If the request still cannot be resolved, it is sent to other local federations via some interconnection structure, the so-called *interconnection overlay* (IO) provider, representing tier 1 in the process. Secondly, tier 3 and 2 entities might choose several next tier entities in the process of resolution. This includes forwarding to different parents for different requests (e.g., based on some local policy). Although such choice does not always exist, it might be important to consider for certain deployments. Requests that cannot be resolved within the interconnection structure are forwarded to other interconnection providers for resolution, i.e., there exists an assumption of interconnecting at tier 1 so as to finally be able to resolve any request.

7.2 Main Design Characteristics

Looking more closely at the structural properties outlined above, we can recognize two major characteristics that will influence the design of most solutions, namely the *existence of distinct players* as well as the *degree of interconnection* between these players[10]. For instance, a higher number of IO providers favors designs with manageable (or low) cost for providing the overlay, while solutions with higher costs for overlay provisioning might still be viable in scenarios with a low number of IO providers. On the other hand, the number of rendezvous networks could provide guidance on required scalability and forwarding efficiency. In addition, the degree of interconnection between these players determines the fragmentation of regions and therefore markets. In the following, we focus on these design characteristics. Within our methodology, we can formulate these characteristics as the following *problems*:

1. How many IO providers will there be?
2. How many rendezvous networks will there be?
3. What is the incentive to interconnect?

These problems are now translated into a set of stock and flow models from which we derive our possible as well as likely socio-economic outcomes. Hence, we can define stocks for three individual models, namely the *number of interconnection overlay providers* as well as the *number of rendezvous network providers*, together with the *incentive to interconnect* (normalized between 0 and 1). In this paper, we focus on the presentation of our results for the first characteristic, i.e., the number of interconnection providers.

7.3 Model for Interconnection Providers

We start our modeling by outlining the reference mode for the expected model. As a result, Figure 9 outlines the possible behavior for the interconnection provider within our architectural context. The timeline as well as the total

---

[10] Other possible characteristics could be the required degree of deployment and the collaboration between systems having deployed the system and the ones that have not.

number on the y-axis are only indicative with the final annotation to be found in the actual scenario-based evaluation cases.

After the initial deployment, we expect a phase of market interest that is reflected by a growing number of providers. At some tipping point, a phase of commercialization occurs, leading to a competitive market with entrants and exits stabilizing. A phase of consolidation leads to either (a) a single dominant player (lower, dotted curve) (b) a stable but limited number of market players (dashed curve) or (c) commoditization of the interconnection provider function over time (e.g., due to technological advances) as shown with the lower, solid curve. The latter commoditization can also occur without commercialization phase, reflected in the uppermost solid curve.

The outcomes in Figure 9 can be identified as various types of markets that are eventually enabled (or prohibited). Monopoly and oligarchy markets are defined by the lower outcomes, while commoditization of the market is captured by the upper outcomes[11].

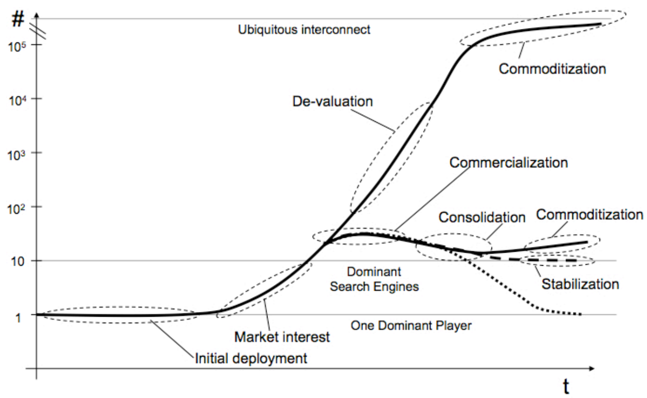


Figure 9: Reference Mode for IO Providers

The underlying system dynamics model utilized in our evaluation is shown in Figure 10. The model is divided into two distinct parts. The lower part models the various factors that influence the barrier of entry for players in the (IO provider) market. As exogenous factors, memory and processing performance as well as reliability of technology utilized for implementation are influencing the barrier to entry with a variable weight. Furthermore, a set of four major concerns influences the propensity to change the barrier of entry through regulatory action. These are information visibility (e.g., through insufficient interconnection), public pressure, (sufficient) rendezvous interconnection and competition.

All these concerns are modeled as being linearly dependent on the number of IO providers up to a given level of IO providers (after which the concern remains constant at a low level). The propensity to regulatory action is influenced by an exogenous factor that represents the lack of clarity in the legislative situation (e.g., through introducing new concerns, change in procedures). A delayed action is modeled through a smoothened delay, determined by the exogenous factor that determines the time to react to regulatory demands. The resulting barrier to entry multiplicatively influences the entrant and exit rate.

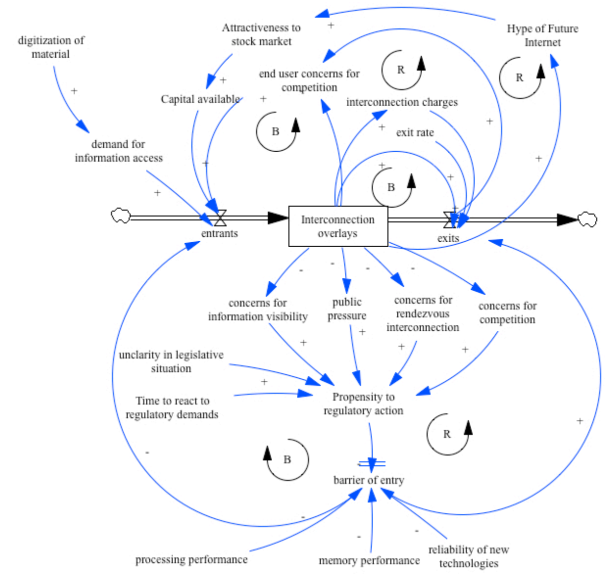


Figure 10: SD Model for Interconnection Providers

The upper part of the model captures several causal loops. The outer loop represents the availability of investment, influenced by the hype for Future Internet technologies. Furthermore, the entrants are influenced by end user concerns for competition, which in itself depends on the number of IO providers and rate of exit from the market. Such dependence is not directly linear but indirectly defined through a utility function that captures perceived quality through the choice of providers. This utility function is linearized in our equations for the model.

As an exogenous factor, the demand for information access influences the entrants in a weighted manner (while such demand is driven by the digitization of material, an exogenous factor in our model). Last but not least, we model a causal relation of interconnection charges and exits from the market, while an exogenous factor determining the general exit rate (e.g., due to capital burnout or other factors) is factored into the exit flow in a weighted manner. We also represent the desire to not interconnect as an exogenous factor into our model. This factor is driven by our model for the interconnection incentive, representing a fully interconnected market through a value of 1.

With this model, we capture regulatory, user-centric, and market causalities as well as certain exogenous drivers that influence the number of IO providers. All variables of the model are parameterized with equations that describe the various dependencies. The equations can be made available as Vensim simulation files upon contacting the authors.

[11] Fragmentation is captured by our work on interconnection incentives, which we omit here for space reasons.

7.4 Example Scenario and Likely Outcomes

Given the possible socio-economic outcomes for our main design characteristics, possible interesting scenarios for testing our models address variations in the size and nature of markets (in the case of IO providers), the regionalization of markets (in the case of RENE networks) and the extent of collaboration in terms of interconnection.

Given our focus on the IO providers, we present a scenario that addresses the size of the markets in this space. We tie our results back to the reference modes, allowing us to reflect on the likely socio-economic outcomes for each scenario. Given that our evaluation focuses on the system-level behavior under a range of parameter changes, it is the shape and final outcome of the curves that is important rather than the exact numerical result of each simulation.

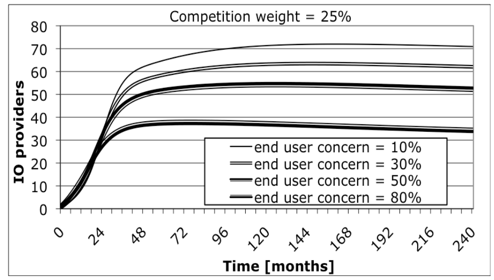
For our scenario, we assume the following configuration. The lifetime of our simulations is 20 years. Assuming a non-discriminating deployment of the solution, an initial delay similar to the current Internet (i.e., the various phases of transfer from a research network over a pre-commercial towards a fully commercial environment) is left out of our consideration[12]. For the capital investment available, we assume \$20 million per month. This represents a venture capital market driven model of entrants. Left for future work is the division of entrants into incumbents and new venture-driven entrants. As the upper limit of entrants in the IO provider market, we assume 20 per month, a number taken from the competitive VPN (virtual private networks) market. We assume a ten-times higher number for RENE providers, given the locality of investment. The reliability of technology factor is assumed to be linearly increasing from 0.1 towards 0.8 over the lifetime of the simulation. This represents the assumed venture-driven entrance while simulating increased maturity of technology over time. For the exit rate of players from the market due to capital burnout issues, we assume 10%.

Scenario: Anti-Monopoly Movement

This scenario assumes an increasing movement away from various monopolies. This might result, for instance, in increasing number of grass root movements of various forms, as for instance seen in the wireless access space with a number of community schemes [24][25]. The parties driving this trend are end users (through public campaigns), legislators (through increased public pressure and the need to address international monopolies), regional powers (not accepting monopolies imposed by other regional powers) and corporations not being successful in establishing themselves as monopolies. The IO provider market is a clear target for such a scenario since possible monopoly or even oligarchy market constellations are not unlikely to occur (see Figure 9 for the possible socio-economic outcomes of this market). In addition, the RENE provider market is also evaluated against this scenario.

[12] Hence, our simulation lifetime represents the lifetime of a commercial market under full deployment capability.

a)



b)

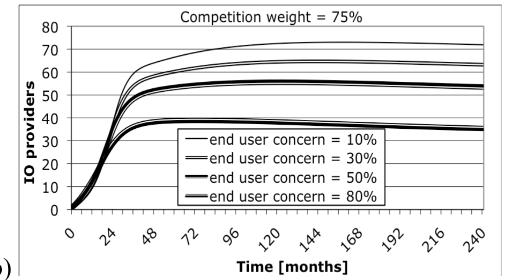


Figure 11: IO Provider Simulations

Figure 11 shows the outcome of our simulations. In a) and b), we vary the weight of end user concerns into the investment decisions for new entrants from 10% to 80% while changing the regulatory concern for competition from 25% to 75%. We can see that in all runs, an increasing end user concern clearly influences the number of IO providers in their final outcome. For the highest concern weight of 80%, we can even see a phase of consolidation towards a slightly lower number. This is not surprising since the end user concerns influence the investment decisions (although not linearly). However, we can still see a substantial investment into the market despite a significant increase of the weight from 10% to 80%, for instance. Hence, a public anti-monopoly movement does have an impact on the market size in terms of players although it does not change its overall outcome, namely that of a significant and stable number of market players.

It is striking that the number of players decreases with an increasing end user concern for competition. This is probably in contrast to the expectation of seeing an increase here (and therefore competition). The reason for this is the equation used for the causality that describes the end user concern for competition. As can be seen in Figure 10, both the number of IO providers and the exit rate of players play a role in the concern. In our equation for the end user concern, we weighted the number of exits more strongly than the impact that lower numbers of IO providers would have, since we saw the rate of exits as being stronger in influence (due to its immediate impact on public opinion compared to a long term trend of change in numbers). This explains the behavior observed in Figure 11.

Our scenario above can only provide a glimpse on the potential that our methodology provides. Our extended work in this space includes a full modeling of all three characteristics with a set of four scenarios, ranging from privacy backlash scenarios over regionalization scenarios to technology breakdowns.

8. ARCHITECTURE LESSONS LEARNED

There are several key points to be taken away from our initial tussle space work.

Design for choice: Various incentives on the regulatory and user side as well as on the side of ISPs emphasize the need for choice in revealing information and assembling functions acting on this information. This is driven by the stronger emphasis in our architecture on the exchange of information, which inherently carries value for actors in the ecosystem. This is somewhat different from, e.g., interconnection in the Internet today, which focuses on resource pooling and therefore cost minimization. Differentiation on service or content level is hardly provided. Hence, any solution should consider mechanisms for choice. Our architectural approach provides such hooks for choice in several functions, e.g., the ITF function, the topology manager, and the rendezvous network. Expression of choice can be implemented by accompanying the information structures (on which the network directly operates) with metadata information that defines the preferences being taken into account for the particular structure to be delivered. We return to this aspect later.

Design for isolation: One expression of choice is a desired isolation of information spaces, each of which might be interconnected by its own provider or to each of which end users might connect through different ISPs (e.g., using specific financial network providers compared to regular ISPs). But also the enforcement of digital rights influences the incentive to widely interconnect within an isolated island of policy enforcement. Such regional power struggles already exist today and are likely to exist in the future. Any design must accommodate these influences. In our architecture, such as aspects of isolation can be accommodated by the clear identification of the various roles that are responsible for functions of finding information, building a delivery graph and eventually delivering the bits (of information). In particular the second function today only exists in the many technological extensions to the Internet, realized in various middlebox solutions, all of which are only badly (if at all) exposed to the various actors in the system[13].

Design for flexible deployment: The need for evolvability of solutions has long been recognized. Hence, any design should consider various deployment scenarios. One such scenario is one being driven by vertical industries being the drivers for initial adoption, significantly contrasting a full adoption model in which every player will need to adopt the technology. Considering various deployment options should be accompanied by a proper understanding of their market impacts through, for instance, utilizing our methodology in this paper.

[13] We recognize that various efforts, e.g., in the IETF, exist to expose such middleboxes. These efforts are driven by similar motivations than ours when it comes to designing for choice.

Decouple business models: Another aspect is that of de-coupling business models, such as interconnection models on the bit and information level. In our rendezvous example of Section 7, such coupling would occur by routing discovery requests along the same upgraph connections that are being established through bit-level interconnection (such as proposed in [26]). With that, one creates a strong alignment of the business models underlying both interconnections, namely that at bit transport and that at discovery level. However, such alignment is not necessarily upheld in reality, such as in the search space today.

Taking into consideration the aforementioned points, there are two major architectural findings that stand out:

Clearly defined modular boundaries are crucial: Clark et al. have elaborated in [4] on the role that modular boundaries play when needing to accommodate tussles at system runtime. Also the work in [27] outlines the importance of clearly separating functions along recognized boundaries in order to minimize the impact of that function (and the tussles surrounding it) on other functions, reducing the overall dependence of players on other stakeholders.

But while these works focus on the general importance of this architectural design principle, our work within an information-centric architecture context provides examples for such modular boundaries, namely that of the three crucial functions of finding information, building appropriate delivery graphs and delivery the bits of information. These main functions are well exposed and defined through the architecture. This aspect is important, for instance, in separating the information brokering from the transport of the final bits. This separation effectively creates a tussle space, and therefore market, boundary between bit- and information-oriented markets.

But also ancillary functions, such as identifier governance, key management, and network attachment, have their clear role in the architecture. Hence, we assert that our particular architectural context provides a positive example for the flexibility that Clark et al assert for any 'good' design along well-defined modular boundaries. Significant future work, however, is required to shed more light on the aspects of what defines 'good' and 'flexibility' in terms of metrics; metrics that could potentially be generalized for other architectural use cases.

Flexibly creating information asymmetries is key: Any formation of markets is based on creating as well as re-shaping appropriate information asymmetries between the various actors within the markets. Any architecture needs to accommodate such fundamental mechanism in order to enable (and sustain) a flourishing ecosystem, i.e., to enable todays but also any future business models that the actors within the system strive to establish.

Within a communication system, this economic observation boils down to enabling a well-defined information exposure between various parties in the system. This has also been recognized by Ford et al. [28] within the example of congestion notification in today's Internet. Generally, such exposure of information (such as preferences, end user interests, topological information being used for routing or observed congestion in the network) needs to be inherently supported by a communication system[14]. We assert that a communication system that itself operates on information (within well-defined and exposed structures for this information) provides an improved ability for supporting such information exposure. We recognize, however, that such assertion needs a larger pool of anecdotal evidence through an extended investigation of use cases; an effort being left for our future work.

9. CONCLUSIONS

Information-centric networking has attracted an increasing attention in the networks community, both on the academic stage but also within corporate research organizations. Given the attention that this particular range of future Internet proposals has been receiving, it seems only natural to study the socio-economic playground and its tussles that such architectural context would bring about. This is not only important for understanding the socio-economic impact of such possible technological change. It is also a crucial exercise in understanding the viability of technological solution proposals within the wider socio-economic environment that is our society.

We clearly recognize that this paper can only be the starting point for exploring the tussle space for a global scale communication system such as the (future) Internet. But we believe that the initial tussle taxonomy in this paper as well as our methodology for codifying and evaluating the viability of large-scale design choices provide a useful first insight. Only a continued future development and application of this methodology will provide the necessary anecdotal evidence for its suitability in exploring and better understanding the tussle spaces of future Internet proposals.

ACKNOWLEDGEMENTS

We would like to thank Costas Courcoubetis, George Parisis, Chintan Vaishnav, Xenofon Vasilakos and Christos Tsilopoulos for their useful comments to our research.

The research of Dirk Trossen is supported by the EU FP7 project PURSUIT under grant FP7-INFSO-ICT 257217.

The research of Alexandros Kostopoulos has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Heracleitus II. Investing in knowledge society through the European Social Fund.

[14] The authors in [7] discuss some of the problems in todays Internet to appropriately express such preferences as well as implications for not being generally able to do so.

REFERENCES

[1] Named-Data Networking at http://www.named-data.net/, 2011

[2] PURSUIT at http://www.fp7-pursuit.eu/, 2011

[3] D. Trossen, M. Sarela, K. Sollins, "Arguments for an Information-Centric Internetworking Architecture", ACM Computer Communication Review, April 2010

[4] D. Clark, J. Wroclawski, K. R. Sollins, R. Braden, "Tussle in Cyberspace: Defining Tomorrow's Internet," in IEEE/ACM Transactions on Networking, Vol. 13, No. 3, 2005

[5] D. Trossen (ed), "Conceptual Architecture: Principles, Patterns and Sub-Component Descriptions", Deliverable D2.2 of the PURSUIT project, at http://www.fp7-pursuit.eu/PursuitWeb/?page_id=158, 2011

[6] P. Jokela, A. Zahemszky, S. Arianfar, P. Nikander, C. Esteve, "LIPSIN: Line speed Publish/Subscribe Inter-Networking", ACM SIGCOMM, August 2009

[7] D. Trossen, G. Biczok, "Not Paying the Truck Driver: Differentiated Pricing for the Future Internet", ReArch 2010 workshop at ACM Context, December 2010

[8] YouView initiative, at http://www.youview.com/, 2011

[9] K. Lakshminarayanan, I. Stoica, S. Shenker, "Routing as a Service", UCB/CSD-04-1327, UC Berkeley, 2004

[10] J. Rajahalme, M. Särela, P. Nikander, S. Tarkoma, "Incentive-Compatible Caching and Peering in Data-Oriented Networks. Re-Arch'08 workshop at ACM CoNext, Dec 2008

[11] B. Briscoe, "Internet: Fairer is Faster", BT White Paper TR-CXR9-2009-001, May 2009

[12] Google Public DNS, at http://code.google.com/speed/public-dns/, 2011

[13] OpenDNS, at http://www.opendns.com, 2011

[14] J. Rochet, J. Tirole, "Two-sided markets: a progress report", The RAND Journal of Economics, vol. 37, pp. 645–667, September 2006

[15] G. Pallis, A. Vakali, "Insight and Perspectives for Content Delivery Networks", Communications of the ACM (CACM), vol. 49, No. 1, pp. 101-106, 2006

[16] D. Clark, B. Lehr, S. Bauer, P. Faratin, R. Sami, J. Wroclawski, "Overlay Networks and the Future of the Internet", Communications & Strategies, no. 63, 3rd quarter 2006

[17] P. Vixie, "What DNS is Not", Communications of the ACM Vol. 52 No. 12, pp. 43-47, December 2009

[18] J. Wulf, R. Zarnekow, T. Hau, W. Brenner, "Carrier Activities in the CDN Market – An Exploratory Analysis and Strategic Implications", 14th Intl. Conference on Intelligence in Next Generation Networks (ICIN), October 2010

[19] I. Papafili, S. Soursos, G.D. Stamoulis, "Improvement of BitTorrent Performance and Inter-Domain Traffic by Inserting ISP-owned Peers", 6th Intl. Workshop on Internet Charging and QoS Technologies (ICQT'09), Aachen, Germany, May 2009

[20] R. Wetzel, "CDN Business Models – Not All Cast from the Same Mold", at http://wetzelconsultingllc.com/CDNArticle.pdf

[21] G. Bertrand, F. Le Faucheur, L. Peterson, “Content Delivery Network Interconnection (CDNI) Experiments”, IETF draft-bertrand-cdni-experiments-00, February 2011

[22] J. D. Sterman, “Business Dynamics: Systems Thinking and Modeling for a Complex World”, McGraw-Hill Higher Education, Boston, 2000

[23] XMind software at http://www.xmind.net, 2011

[24] I. Brown (ed), “Towards a Future Internet – State of the Art report”, Study for European Commission, 2009

[25] Rand Corporation, “Trends in connectivity technologies and their socioeconomic impacts”, Technical Report 776, 2009

[26] J. Rajahalme, M. Särelä, K. Visala, J. Riihijarvi, “Inter-Domain Rendezvous Scalability in Content Networking Architectures”, TR09-003 at http://www.psirp.org, 2009

[27] C. Kalogiros, A. Kostopoulos, A. Ford, “On Designing for Tussle: Future Internet in Retrospect”. EUNICE 2009 - The Internet of the Future, LNCS 5733, pp. 98–107, September 2009

[28] A. Ford, P. Eardley, B. van Schewick, “New Design Principles for the Internet”, International Workshop on the Network of the Future, 2009