Deployment and Adoption of Future Internet Protocols

Philip Eardley[1], Michalis Kanakakis[2], Alexandros Kostopoulos[2], Tapio Levä[3], Ken Richardson[4], and Henna Warma[3]

[1] BT Innovate & Design, UK
philip.eardley@bt.com
[2] Athens University of Economics and Business, Greece
{kanakakis, alexkosto}@aueb.gr
[3] Aalto University, School of Electrical Engineering, Finland.
{tapio.leva, henna.warma@aalto.fi}
[4] Roke Manor Research, UK
ken.richardson@roke.co.uk

Abstract. Many, if not most, well-designed Future Internet protocols fail, and some badly-designed protocols are very successful. This somewhat depressing statement illustrates starkly the critical importance of a protocol's deployability. We present a framework for considering deployment and adoption issues, and apply it to two protocols, Multipath TCP and Congestion Exposure, which we are developing in the Trilogy project. Careful consideration of such issues can increase the chances that a future Internet protocol is widely adopted.

Keywords: Protocol Deployment, Adoption Framework, Multipath TCP, Congestion Exposure.

1 Introduction

New protocols and systems are often designed in near isolation from existing protocols and systems. The aim is to optimise the technical solution, in effect for a greenfield deployment. The approach can be very successful, a good example being GSM but there are many more examples of protocols that are well-designed technically but where deployment has failed or been very difficult, for example interdomain IP multicast and QoS protocols. On the other hand there are several examples of protocols that have been successfully deployed despite a weak technical design (by general consent), such as WEP (Wired Equivalent Privacy).

Several attempts have been made at studying the adoption of consumer products [1] and new Internet protocols, including [2], [3], [4], [5], [6] and [7], which we build on. The adoption of Internet protocols is tricky because the Internet is a complex system with diverse end-systems, routers and other network elements, not all of whose aspects are under the direct control of the respective end users or service providers.

In this Chapter we propose a new framework for a successful adoption process (Section 2), and apply it to two emerging protocols, Multipath TCP (Section 3) and Congestion Exposure (Section 4).

J. Domingue et al. (Eds.): Future Internet Assembly, LNCS 6656, pp. 133–144, 2011.

The framework is not a "black box" where candidate protocols are the inputs and the output is the protocol that is certain to be adopted. Rather, it is a structured way of thinking, useful at the design stage to improve the chances that the new protocol will be widely deployed and adopted.

2 A Framework for the Deployment and Adoption of Future Internet Protocols

We propose a new framework (Figure 1) for a successful adoption process, with several key features:

- It asks two key questions at each stage: what are the benefits and costs? And is it an incremental process?
- It distinguishes an initial scenario from one where adoption is widespread
- It distinguishes implementation, deployment and adoption

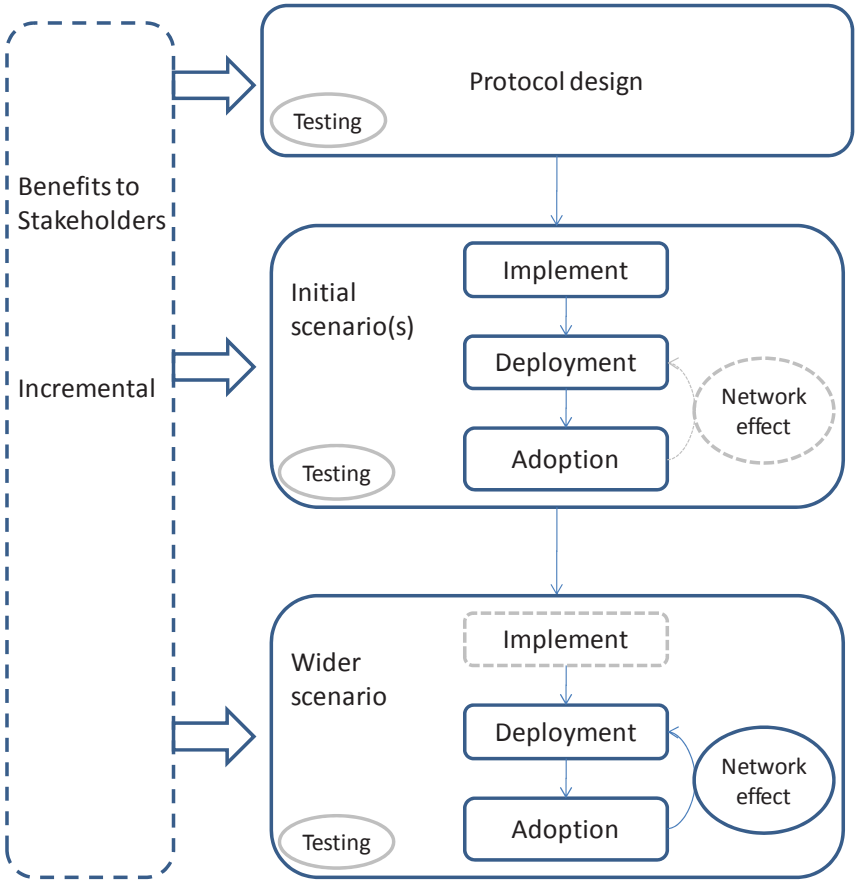

Fig. 1. An adoption framework

A version of the framework has been applied in two papers, [8] and [9]. The framework is intended to be generally applicable to Internet protocols.

The first key question is: *what are the benefits (and costs) of the protocol?* There must be a "positive net value (meet a real need)" [2]. Further, the benefits and costs should be considered for each specific party that needs to take action, as "the benefit of migration should be obvious to (at least) the party migrating" [3]. For example, browsers and the underlying http/html protocols give a significant benefit to both the end users (a nice user interface for easy access to the web) and to the content providers (their content is accessed more; new opportunities through forms etc). As another example, a NAT (Network Address Translator) allows an operator to support more users with a limited supply of addresses, and has some security benefit. As a counter-example, IPv6 deployment has a cost to the end host to support the dual stack, but the benefit is quite tenuous ('end-to-end transparency') and long-term. Deploying a new protocol may have knock-on costs, for example a new application protocol may require changes to the accounting and OSS (Operations Support Systems).

The second key question is: *can the changes required be adopted incrementally?* This is similar to the guideline "Contain coordination and Constrain contamination" [3], meaning that the scope of changes should be restricted in terms of (respectively) the number of parties involved and the changes required within one party. Backwards compatibility is also important. Successful examples include: https, which doesn't require deployment of an infrastructure to distribute public keys; and NATs, which can be deployed by an ISP without coordinating with anyone else. As a counter-example, IPv6 requires at least both ends and preferably the network to change.

Combining these two key factors leads to the idea of an incremental process, where the aim at each step is to bring a net benefit for the party(s) migrating. So commercial, not technical, considerations should determine what the right step size is – it adds sufficient functionality to meet a specific business opportunity. If each step is the same, this is equivalent to saying that there should be a benefit for earlier adopters. However, often the steps will be different, as typically a protocol gets deployed and adopted in a specific use case, later widening out if the protocol proves its utility. Then each step may involve different stakeholders, for example BitTorrent was initially adopted by application developers (and their end-users) to transfer large files, later widening out to some Content Providers, such as Tribler, to distribute TV online. Hence the framework distinguishes initial scenarios from a widespread one.

At each step of the framework careful consideration is needed of benefits and incrementalism. But such consideration should not wait until the initial scenario is about to start. Instead, during the design a mental experiment should be performed to think about a narrow use case and about the final step of widespread deployment and adoption. The more specific thinking may reveal new factors for the design to handle.

Finally, at each step the framework makes a distinction between the concepts of implementation, deployment and adoption:

- *Implementation* refers to the coding of the new protocol by developers
- *Deployment* refers to the protocol being put in the required network equipment and/or end-hosts
- *Adoption* is dependent upon deployment, with the additional step that the protocol is actually used.

For network equipment the distinction between implementation and deployment is particularly important because different stakeholders are involved – equipment vendors implement, whilst network operators deploy; their motivations are not the same.

No further implementation may be needed at the "wider scenario" stage, since the software has already been developed for the initial scenario and it is simply a matter of deploying and adopting it on a wider scale. Perhaps an enhanced version of the protocol can include lessons learnt from the initial use case. But for some protocols the wider scenario requires extra critical functionality – for example, security features, if the initial scenario is within a trusted domain. Also, at the wider scenario stage, "network externalities" are likely to be important: the benefit to an adopter is bigger the greater the numbers who have already adopted it [5].

Testing of the new protocol is included within each of these stages. For example, vendors will validate their implementation of the new protocol, operators will check that it works successfully if deployed on their network, and users will complain if they adopt it and it breaks something.

Note that it is not possible to prove that the framework is necessary or sufficient to guarantee the adoption of a protocol – the framework is not a "black box" with an input of a candidate protocol and an output of yes/no as to whether it will be adopted. The framework also ignores factors such as risks (deployment is harder if the associated risk is higher), regulatory requirements and the role of hype and "group think".

When there are competing proposals (which should be selected for deployment?) it is important to think through the issues in the framework, otherwise an apparently superior protocol may be selected that proves to be not readily deployable. It may be better instead to incorporate some of its ideas, perhaps in a second release.

The main message of this Chapter is that implementation, deployment and adoption need to be thought about carefully during the design of the protocol - for example, mental experiments performed for narrow and widespread scenarios.

3 Multipath TCP

Multipath TCP (MPTCP) enables a TCP connection to use multiple paths simultaneously. The current Internet's routing system only exposes a single path between a source-address pair, and TCP restricts communications to a single path per transport connection. But hosts are often connected by multiple paths, for example mobile devices have multiple interfaces.

MPTCP supports the use of multiple paths between source and destination. When multiple paths are used, MPTCP will react to failures by diverting the traffic through paths that are still working and have available capacity. Old and new paths can be used simultaneously. Since MPTCP is aware of the congestion in each path, the traffic distribution can adapt to the available rate of each path – the congestion controllers for the paths are coupled. This brings the benefits of resilience, higher throughput and handles more efficiently sudden increases in demand for bandwidth.

MPTCP is currently under development at the IETF [10]. The MPTCP design [11] provides multipath TCP capability when both endpoints understand the necessary

extensions to support MPTCP. This allows endpoints to negotiate additional features between themselves, and initiate new connections between pairs of addresses on multi-homed endpoints.

The basic idea of building multipath capability into TCP has been re-invented multiple times [12] [13] [14] [15] [16] [17] [18]. However, none of these proposals made it into the mainstream. The detailed design of MPTCP strives to learn the lessons from these proposals, for instance:

- It builds on the breakthrough of [19] [20], who showed theoretically that the right coupled congestion controller balances congestion across the sub-flows in a stable manner. Stability is required for the benefits (of resilience and throughput) to be worthwhile.
- It seeks to be equitable with standard TCP, essentially meaning that at a bottleneck link MPTCP consumes the same bandwidth as TCP would do; again, this helps persuade the IETF that it is safe to deploy on the internet. Also an operator might otherwise be tempted to block MPTCP to prevent the degradation of the throughput of its "legacy" users.
- It is designed to be application-friendly: it just uses TCP's API so it looks the same to applications. This, plus the following two bullets, help MPTCP be incrementally deployable.
- It automatically falls back to TCP if the MPTCP signalling fails, hence an MPTCP user can still communicate with legacy TCP users and can still communicate if the signalling is corrupted by a middlebox.
- It is designed to be middlebox-friendly (be it a NAT, firewall, proxy or whatever), in order to increase the chances that MPTCP works when there are middleboxes en route:
  - MPTCP appears "on the wire" to be TCP
  - The signalling message that adds a new sub-flow includes an Address ID field, which allows the sender and receiver to identify the sub-flow even if there is a NAT
  - Either end-host can signal to add a new path (in case one end-host's signalling is blocked by a middlebox).
  - MPTCP's signalling is in TCP-Options, because signalling in the payload is more likely to get traumatised by some middleboxes.
  - There is a separate connection-level sequence number, in addition to the standard TCP sequence number on each sub-flow; if there was only a connection-level sequence number, on one sub-flow there would be gaps in the sequence space, which might upset some proxies and intrusion detection systems.
- NAT behaviour is unspecified and so, despite our care in designing MPTCP with NATs in mind, their behaviour may be quite surprising. So we are now working on a NAT survey to probe random paths across the Internet to test how operational NATs impact MPTCP's signalling messages [21].

There are also various proposals for including multipath capability in other transport protocols, such as SCTP [22], RTP [23] and HTTP [24].

For MPTCP, our current belief is that a data centre is the most promising initial scenario (Figure 2). Within a data centre, one issue today is how to choose what path to use between two servers amongst the many possibilities - MPTCP naturally spreads traffic over the available paths.

- Benefits: Simulations show there are significant gains in typical data centre topologies [25], perhaps increasing the throughput from 40% to 80% of the theoretical maximum. However, the protocol implementation should not impact hardware offloading of segmentation and check-summing. One reason that MPTCP uses TCP-Options for signalling (rather than the payload) is that it should simplify offloading by network cards that support MPTCP, due to the separate handling of MPTCP's signalling and data.
- Incremental: the story is good, as only one stakeholder is involved viz the data centre operator.

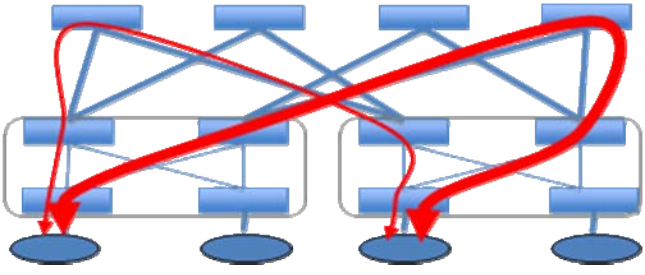

Fig. 2. Potential MPTCP deployment scenario, in a data centre. In this example, traffic between the two servers (at the bottom) travels over two paths through the switching fabric of the data centre (there are four possible paths).

Another potential initial scenario would be a mobile user using MPTCP over multiple interfaces. The scenario reveals a potential distinction between deployment (which involves the OS vendor updating their stack) and adoption (which means that MPTCP is actually being used and requires the consumer to have multiple links) – so in theory it would be possible for MPTCP to be fully deployed but zero adopted. (Note there's little distinction between implementation and deployment, since it is only in end-hosts and deployment is mainly decided by the OS (Operating System) vendor and not the end user.)

Therefore we believe that a more promising initial scenario is an end user that accesses content, via wireless LAN and 3G, from a provider that controls both end user devices and content servers [26] – for example, Nokia or Apple controls both the device and the content server, Nokia Ovi or Apple App Store.

- Benefits: MPTCP improves resilience - if one link fails on a multi-homed terminal, the connection still works over the other interface. But it is a prerequisite, and cost, that devices are multihomed.
- Incremental: Both the devices and servers are under the control of one stakeholder, so the end user 'unconsciously' adopts MPTCP. However, there may be NATs on the data path, and MPTCP's signalling messages must get through them.

The wider scenario of widespread deployment and adoption is again worth thinking about this even during the design of the protocol.

- Benefits: Several stakeholders may now be involved. For instance, it is necessary to think about the benefits and costs for OS vendors, end users, applications and ISPs (Internet Service Providers). Here also we see the importance of network effects. For instance, as soon as a major content provider, such as Google, deploys MPTCP – perhaps as part of a new application with better QoS - then there is a much stronger incentive for OSs to deploy it as well, as the network externality has suddenly increased.
- Incremental: Existing applications can use MPTCP as though it was TCP, ie the API is unaltered (although there will also be an enhanced API for MPTCP-aware applications). MPTCP is an extension for end-hosts – it doesn't require an upgrade to the routing system; if both ends of the connection have deployed MPTCP, then it "just works" (NATs permitting).

4 Congestion Exposure

The main intention of Congestion Exposure (Conex) is to make users and network nodes accountable for any congestion that is caused by the traffic they send or forward. This gives the right incentives to promote cooperative traffic management, so that (for example) the network's resources are efficiently allocated, senders are not unnecessarily rate restricted, and an operator has a better incentive to invest in new capacity.

Conex introduces a mechanism so that any node in the network has visibility of the whole-path congestion – and thus also rest-of-path congestion (since it can measure congestion-so-far, by counting congestion markings or inferring lost packets), Figure 3. A Conex-enabled sender adds signalling, in-band at the IP layer, about the congestion encountered by packets earlier in the same flow, typically 1 round trip time earlier.

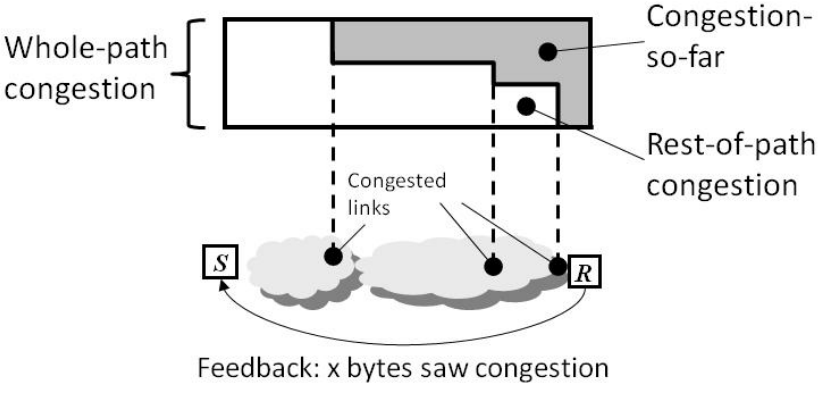

Fig. 3. Conex gives all nodes visibility of the whole path congestion, and thus also rest-of-path congestion.

(In today's internet this information is only visible at the transport layer, and hence not available inside the network without packet sniffing.)

Conex is currently under development at the IETF [27].

Today, without Conex, a receiver reports to the sender information about whether packets have been received or whether they have been lost (or received ECN-marked). The former causes a Conex-sender to flag packets it sends as "Conex-Not-Marked", and the latter to flag packets as "Conex-Re-Echo". By counting "Conex-Re-Echoes", any node has visibility of the whole-path congestion.

Conex also requires, by default, two types of functionality in the network. Firstly, an auditor to catch those trying to cheat by under-declaring the congestion that their traffic causes; the auditor checks (within a flow or aggregate) that the amount of traffic tagged with Conex-Re-Echo is at least equal to the amount of traffic that is lost (or ECN-marked). Secondly, a policer to enforce policy specifically related to the user being served. A user pays, as part of its contract, to be allowed to cause a certain amount of congestion. The policer checks the user is within its allowance by counting the Conex-Re-Echo signals. Similarly, a policer at a network's border gateway checks that a neighbouring ISP is within its contractual allowance.

Conex's default requirement for a policer and auditor, as well as a Conex-enabled sender, is problematic as it requires several stakeholders to coordinate their deployment [9]. Since this is likely to be difficult, we seek an initial scenario that is more incrementally deployable.

We believe that the most promising initial scenario for Conex is as part of a "premium service" by an operator who runs both an ISP and a CDN (Content Distribution Network); network operators are increasingly seeking to run their own CDN, to reduce their interconnect charges and to decrease the latency of data delivery. The CDN server sends "premium" packets (perhaps for IPTV) as Conex-Not-Marked or Conex-Re-Echo. Conex traffic is prioritised by the operator ("premium service"). To a first order of approximation, the only point of contention is the backhaul – where the operator already has a traffic management box, typically doing per end user (consumer) volume caps and maybe deep packet inspection, to provide all users with a "fair share". The operator upgrades its traffic management box so that it drops Conex traffic with a lower probability. However, the operator does not need to deploy a policer or auditor, since it is also running the CDN and therefore trusts it.

In the initial scenario the CDN offers a range of deals to Content Providers, with the more expensive deals allowing a Content Provider to send more Conex traffic and more Conex-Re-Echoes (ie to cause more congestion) – effectively the CDN offers different QoS classes. In turn, the content provider (presumably) charges consumers for premium content.

- Benefits: The CDN offers a premium service to its Content Providers. Also, the Conex (premium) traffic is not subject to per end user caps or rate limits by the ISP.
- Incremental: Only one party has to upgrade, ie the combined CDN-ISP. The Content providers and consumers don't know about Conex. Note that the receiver doesn't need to be Conex-enabled, and the network doesn't need to support ECN-marking.

One way this scenario could widen out is that the content provider is now informed about the Conex-Re-Echoes and upgraded to understand them. The benefit is that, at a time of congestion, the content provider can manage its premium service as it wants - effectively it can choose different QoS classes for different users.

Another way this scenario could develop is that the operator offers the service to all CDNs, again as a premium service. However, the ISP can no longer trust that the CDN is well-behaved – it might never set packets as Conex-Re-Echo in order to try to lower its bill. Therefore the ISP needs to upgrade two things. Firstly its traffic management box: it needs to do occasional auditing spot-checks, to make sure that after it drops a packet then it hears (a round trip time later) a Conex-Re-Echo packet from the CDN sender. Secondly, its gateway with the new CDN needs to count the amount of Conex traffic and the amount of Conex-Re-Echo, to make sure the CDN stays within its contractual allowance.

Eventually the scenario could widen out to end hosts (consumers) so that the ISP also offers them the premium service. Most likely the regular consumer contracts would include some allowance and then the host's software would automatically send the user's premium traffic (VoIP say) as Conex-enabled. In this case the ISP needs to upgrade its traffic management box to check the consumer stays within their Conex allowance.

- Benefits: The premium service is offered to other CDNs, ISPs and consumers - effectively QoS is controlled by the CDN or end user, so that they choose which of their traffic is within which class of QoS, always bearing in mind that they must stay within the limits that they have paid for.
- Incremental: Conex capability is added a CDN or end user at a time.

5 Enhancing the Framework

One important development in telecoms is virtualisation. Although the basic idea is long-standing, it has recently come to much greater practical importance with the rise of cloud networking. Normally the advantages are explained in terms of storage and processing "in the cloud" at lower cost, due to efficiency gains through better aggregation. However, there is also an interesting advantage from the perspective of deployment. A new application can be deployed "on the cloud" – effectively the end users use a virtualised instance of the new application. Although our adoption framework is still valid, there are now differences in emphasis:

- Roll out of the software should be cheaper, therefore the expected benefits of the deployment can be less.
- There is no need to coordinate end users all having to upgrade. Every user can immediately use the new (virtualised) software, so effectively a large number of users can be enabled simultaneously.
- These factors reduce the deployment risk, especially as it should also be easier to "roll back" if there is some problem with the new software.

Virtualisation is not suitable for all types of software, for instance new transport layer functionality, such as MPTCP and CONEX, needs to be on the actual devices.

There is an analogy with the digitalisation of content, which has greatly lowered the costs of distribution. Virtualisation should similarly lower the cost of distribution – in other words, it eases deployment.

Another aspect is the interaction of a new protocol with existing protocols. It is important that the design minimises negative interactions, and to test for this. For instance the MPTCP is designed to cope if a middlebox removes MPTCP-options.

There will also be cases of positive interactions, where a new protocol suddenly enables an existing protocol to work better. One set of examples is the various IPv4-IPv6 transition mechanisms that try to release the (currently hidden) benefits of IPv6. Another example is a protocol "bundle", for instance telepresence offerings now wrap together several services that separately had less market traction.

6 Conclusions

The main message of this Chapter is that implementation, deployment and adoption need to be thought about carefully during the design of the protocol, as even the best technically designed protocol can fail to get deployed. Initial narrow and subsequent widespread scenarios should be identified and mental experiments performed concerning these scenarios in order to improve the protocol's design. We have presented a framework; by using it we believe a designer improves the chances that their protocol will be deployed and adopted.

We have applied the framework to two emerging protocols which we are developing in the Trilogy project [28]. Multipath TCP (MPTCP) is designed to be incrementally deployable by being compatible with existing applications and existing networks, whilst bringing benefits to MPTCP-enabled end users. For Congestion Exposure (Conex), a reasonable initial deployment scenario is a combined CDN-ISP that offers a premium service using Conex, as it requires only one party to deploy Conex functionality.

Acknowledgments. This research was supported by Trilogy (http://www.trilogy-project.org), a research project (ICT-216372) partially funded by the European Community under its Seventh Framework Programme. The views expressed here are those of the authors only. The European Commission is not liable for any use that may be made of the information in this document. Alexandros Kostopoulos is co-financed by the European Social Fund and National Resources (Greek Ministry of Education – HERAKLEITOS II Programme).

References

1. Rogers, E.: Diffusion of Innovations. Free Press, New York (1983)
2. Thaler, D., Aboba, B.: What Makes for a Successful Protocol? RFC 5218 (2008)

3. Burness, L., Eardley, P., Akhtar, N., Callejo, M.A., Colas, J.A.: Making migration easy: a key requirement for systems beyond 3G. In: VTC 2005-Spring, IEEE 61st Vehicular Technology Conference (2005)
4. Hovav, A., Patnayakuni, R., Schuff, D.: A model of Internet Standards Adoption: the Case of IPv6. Information Systems Journal 14(3), 265–294 (2004)
5. Katz, M., Shapiro, C.: Technology Adoption in the Presence of Network Externalities. Journal of Political Economics 94, 822–841 (1986)
6. Joseph, D., Shetty, N., Chuang, J., Stoica, I.: Modeling the Adoption of New Network Architectures. In: International Conference on Emerging Networking Experiments and Technologies (2007)
7. Dovrolis, C., Streelman, T.: Evolvable network architectures: What can we learn from biology? ACM SIGCOMM Computer Communications Review 40(2) (2010)
8. Kostopoulos, A., Warma, H., Leva, T., Heinrich, B., Ford, A., Eggert, L.: Towards Multipath TCP Adoption: Challenges and Perspectives. In: NGI 2010 - 6th EuroNF Conference on Next Generation Internet, Paris (2010)
9. Kostopoulos, A., Richardson, K., Kanakakis, M.: Investigating the Deployment and Adoption of re-ECN. In: ACM CoNEXT ReArch'10, Philadelphia, USA (2010)
10. Multipath TCP Working Group, IETF. Latest status at http://tools.ietf.org/wg/mptcp (2010)
11. Ford, A., Raiciu, C., Handley, M.: TCP Extensions for Multipath Operation with Multiple Addresses, draft-ford-mptcp-multiaddressed-02.txt, work in progress (2010)
12. Huitema, C.: Multi-homed TCP, draft-huitema-multi-homed-01.txt, work in progress (2005)
13. Hsieh, H.-Y., Sivakumar, R.: pTCP: An End-to-End Transport Layer Protocol for Striped Connections. In: IEEE International Conference on Network Protocols, ICNP (2002), http://www.ece.gatech.edu/research/GNAN/work/ptcp/ptcp.html
14. Rojviboonchai, K., Aida, H.: An Evaluation of Multi-path Transmission Control Protocol (M/TCP) with Robust Acknowledgement Schemes. Internet Conference IC (2002)
15. Zhang, M., Lai, J., Krishnamurthy, A., Peterson, L., Wang, R.: A Transport Layer Approach for Improving End-to-End Performance and Robustness Using Redundant Paths. In: Proc. of the USENIX 2004 Annual Technical Conference (2004)
16. Dong, Y.: Adding concurrent data transfer to transport layer, ProQuest ETD Collection for FIU, Paper AAI3279221 (2007), http://digitalcommons.fiu.edu/dissertations/AAI3279221
17. Sarkar, D.: A Concurrent Multipath TCP and Its Markov Model. In: IEEE International Conference on Communications, ICC (2006)
18. Hasegawa, Y., Yamaguchi, I., Hama, T., Shimonishi, H., Murase, T.: Improved data distribution for multipath TCP communication. In: IEEE GLOBECOM (2005)
19. Kelly, F., Voice, T.: Stability of end-to-end algorithms for joint routing and rate control. Computer Communication Review 35, 2 (2005)
20. Key, P., Massoulie, P., Towsley, D.: Combined Multipath Routing and Congestion Control: a Robust Internet Architecture, no. MSR-TR-2005-111 (2005), http://research.microsoft.com/pubs/70208/tr-2005-111.pdf
21. Honda, M.: Call for contribution to middlebox survey (2010), http://www.ietf.org/mail-archive/web/multipathtcp/current/msg01150.html
22. Becke, M., Dreibholz, T., Iyengar, J., Natarajan, P., Tuexen, M.: Load Sharing for the Stream Control Transmission Protocol (SCTP), draft-tuexen-tsvwg-sctp-multipath-00.txt, work in progress (2010)

23. Singh, V., Karkkainen, T., Ott, J., Ahsan, S., Eggert, L.: Multipath RTP (MPRTP), draft-singh-avt-mprtp, work in progress (2010)
24. Ford, A., Handley, M.: HTTP Extensions for Simultaneous Download from Multiple Mirrors, draft-ford-http-multi-server, work in progress (2009)
25. Raiciu, C., Plunkte, C., Barre, S., Greenhalgh, A., Wishcik, D., Handley, M.: Data center Networking with Multipath TCP. ACM Sigcomm Hotnets (2010)
26. Warma, H., Levä, T., Eggert, L., Hämmäinen, H., Manner, J.: Mobile Internet In Stereo: an End-to-End Scenario. In: 3rd Workshop on Economic Traffic Management, ETM (2010)
27. Congestion Exposure Working Group, IETF. Latest status at `http://tools.ietf.org/wg/conex` (2010)
28. Trilogy project, `http://trilogy-project.org/` (2010)