On the Way to a Theory for Network Architectures

Thi-Mai-Trang Nguyen

On the Way to a Theory for Network Architectures

Thi-Mai-Trang Nguyen[1],

[1] University Pierre et Marie Curie (UPMC) – Laboratoire d'Informatique de Paris 6 (LIP6),
4 Place Jussieu, 75005 Paris, France
Thi-Mai-Trang.Nguyen@lip6.fr

Abstract. The design of the future Internet is facing real challenges on network architecture. Attempts to resolve the issues related to naming/addressing, middle boxes, QoS-Security-Mobility interactions, cross-layer and inter-domain usually lead to endless debates. This is because the structure of the Internet has become too complex and has evolved with new functions added which are not always compatible with the existing functions. In order to analyze the architecture of the Internet in a strict manner, it is necessary to understand in depth the composition of functionalities within a protocol or between protocols. This paper presents a study on the composition of network functionalities and highlights future directions towards a theory for network architectures which includes the principles that network architectures should follow to ensure the normal operation of the member functions, detect all possible conflicts between them as well as figure out impossibilities.

Keywords: Network architecture, future Internet.

1 Introduction

The architecture of the Internet has been much evolved since its creation. In comparison with the beginning of its design, the today's Internet has many new protocols, communication paradigms, device types and applications. Important new protocols include Real-time Transmission Protocol (RTP) [1] for real time multimedia transmission, Mobile IP [2] for mobility at Internet Protocol (IP) level, IPsec [3] and Transport Layer Security (TLS) [4] for security at IP and transport levels respectively, SCTP [5] for multihoming, and IPv6 [6] for a new addressing scheme. New wireless access technologies such as Wi-Fi, Wi-Max, GPRS/3G/4G increase the mobility in the Internet and allow new communication paradigms such as ad-hoc networking and vertical handover. High speed access technologies such as ADSL and optical fiber offer a higher bit rate to end-user enabling service convergence between voice, television and data in the Internet at large scale.

When adding new protocols into the Internet, new architectural elements have been introduced. Mobile IP has defined a Home Agent to make IP address changes transparent to the correspondent node. Network Address Translation (NAT) [7] entity has been added into IP router to translate addresses between private IP addresses and public IP addresses. The complexity of the Internet architecture has been increased with new protocols and elements added. Many conflicts have been raised in the

literature. NAT and firewalls conflict with the end-to-end principle of the Internet [8]. Wireless link and mobility make Transmission Control Protocol (TCP) congestion control confuse the reason of packet loss and get poor performances [9]. When incompatible design principles are implemented within the same network without regard to the presence of each other, the protocols cannot operate properly to provide the services expected from the design. The general consensus of the research community is that the methodology of continuously patching the Internet is not sustainable with the continuing growth on size, demands and complexity.

Future Internet design has become a federating theme of European research since 2007 with the activities in the 7th Framework Programme (FP7) [10]. The FP7 4WARD project [11] follows a clean-slate approach to rethink the design of the Internet architecture from scratch taking into account current user needs and technological advents on transmission and devices. Two of main studies carried out within the 4WARD project are the Generic Path (GP) concept [12] and the Architectural framework [13]. The Generic Path concept defines basic entities needed to support a variety of communication paradigms and enable the dynamic setup and control of communication paths which can be rich in functionalities. The Architectural framework defines the concepts of Netlet and Strata as two basic entities encapsulating the functionalities provided by the composed functional blocks. In both studies, the composition of network functionalities plays an important role in the design of protocols for Generic Path, Netlet or Strata. This paper presents a study on network functionality composition which can be used during the design of protocols encapsulated within these entities. This study can be also useful for network protocol design in general and may be combined with existing theory such as graph theory for a theory of network architectures.

The organization of the paper is as follows. Section 2 presents the relationship between network architecture and the composition of functionalities. Section 3 proposes taxonomy of main network functionalities within a network. Section 4 describes the order which can be followed during the protocol design to integrate network functionalities into a protocol. Section 5 presents the design of a multihoming protocol based on functionality composition. Section 6 discusses the possibilities to integrate the functionality composition principles into existing communication theory for a theory of network architecture. Finally, section 7 concludes the paper.

2 Network Architecture

Network architecture specifies the elements defined within a network, their functionalities and the communication between them. Elements can be physical elements or logical elements. Logical elements are defined and associated with different functionalities but may be implemented within the same physical entity in practice. For example, the Session Initiation Protocol (SIP) architecture [14] defines the proxy server and the registration server as two elements with two distinct functionalities, one relaying the SIP requests towards the destination and the other maintaining the binding between SIP address and IP address, but they can be

implemented within the same physical machine in practice. The communications between elements are defined as interfaces and protocols. The Global System for Mobile Communications (GSM) architecture [15] has named interfaces between elements and specifies the protocols used over each interface. The Internet architecture only specifies the protocols between the elements without naming the interfaces.

The Internet that we have today relies on the interconnection of many networks following different architectures, one relying on or collaborating with another. The interconnection between networks can follow the layer model, a.k.a. the Open System Interconnection (OSI) model, or in a collaborating model which is not clearly layered. For example, the IP network using the IP protocol relies on the underlying networks which can be Ethernet, Wi-Fi or cellular networks and have completely different network architectures and protocols. This interconnection follows the layer model in which the IP packets are encapsulated within the underlying network protocol's Protocol Data Unit (PDU). However, the collaboration between SIP and RTP is not layered but rather shares the same layer. SIP has its own architecture with User Agent, Proxy Server, etc. and RTP has its own architecture with sender, receiver, translator and mixer. We can consider the nodes supporting the SIP protocol as a sub network and the node supporting the RTP protocol as a sub network. These sub networks co-exist and provide complementary functionalities which are signaling and transport. This concept of sub network corresponds to the concept of Strata [16] or compartment [12] defined in the 4WARD project.

Stratum is modeled as a set of nodes containing functionality for data processing, and a medium which defines how data can be transferred between nodes in the stratum [16]. The nodes within a stratum correspond to the elements defined for a network. The medium corresponds to the communication protocols between them. Compartment is defined as a set of entities with an associated namespace and they can communicate between them using a common (set of) protocol(s) [12]. The entities within a compartment correspond to the elements defined for a network. They can use a protocol or a protocol suite defined for the communication between them. The communication between entities within a compartment is defined as a GP. Both stratum and compartment are expected to be composed not only in a layering manner but also in an arbitrary manner. While stratum concept characterizes the relations between strata by service provider-customer relationship via SSP and peering relationship via SGP, the compartment concept studies the relations between compartments by looking inside the nodes participating in different compartments. It's inside these nodes that the GPs belonging to different compartments are interconnected via hooks or Mediation Point. A hook table can be considered as a simple realization of Mediation Point. More complex Mediation Point can realize complex functionalities such as multiplexing, routing, switching, or scheduling to interconnect different GPs.

While both stratum and compartment can model any network by composing the specified strata or compartments in a flexible manner, there is a need to analyze the interaction between the functionalities encapsulated within these strata or compartments and determine the design trade-offs when composing the functionalities. The study on functionality composition and interaction is also useful to solve the issues within the current Internet. The protocols added to the Internet to

provide some new functionality may have bad impact on the functionalities provided by the existing protocols within the network. This study will help network architects for a better design when adding functionalities within a layer or a protocol as well as when adding a layer or a protocol into a protocol stack or a protocol set.

3 Network Functionalities

When having a look at the functionalities that we can have within a network, we can see that network functionalities can be divided into six categories as shown in **Fig. 1**.

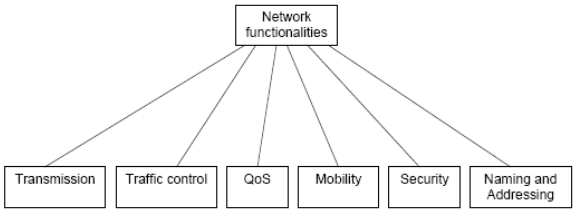

Fig. 1. Network functionality taxonomy.

The *transmission* functionality category includes functionalities such as sending, receiving, forwarding, routing, switching, storing, and processing PDUs. Data transmission is the basic functionality in any communication system. The simplest communication system is composed of two nodes, one sending data and the other receiving data without any other additional functionality such as QoS, naming/addressing or traffic control. A more advanced network node could also store data along the path such as in the case of Delay Tolerant Network [17] or even process data in case of network coding [18].

The *traffic control* functionality category includes flow control, error control, congestion control, and reordering control. Flow control is responsible to control the data rate at the sender. Error control deals with errors occurring during the transmission. Congestion control reacts to the congestion experienced in the network. Reordering concerns the out-of-order transmission and puts PDUs back in order. These functionalities can be composed and integrated into a protocol according to the requirement. For instance, if we add error control or flow control functionality to the IP protocol, we will have a network layer protocol with error control or flow control. These functionalities are not dedicated to the transport layer or any other specific layer. A typical example of a protocol using error and flow control is HDLC (High-level Data Link Control) [19], a protocol at layer 2.

Similarly, QoS, Mobility, Security, and Naming/Addressing are main functionality categories in a communication network which are composable. Queuing discipline and admission control are parts of *QoS* functionality group. Handover and location management are examples of *Mobility* functionality group. Authentication, authorization, encryption, data integrity, and key distribution are elementary functionalities of the *Security* functionality group. Naming schemes and name resolution techniques are part of the *Naming/Addressing* functionality group.

4 Composition of Functionalities

We can design new protocols and get a desired network by composing the functionalities, or more concretely composing the functional blocks implementing the functionality needed. **Fig. 2** illustrates the idea of functional composition.

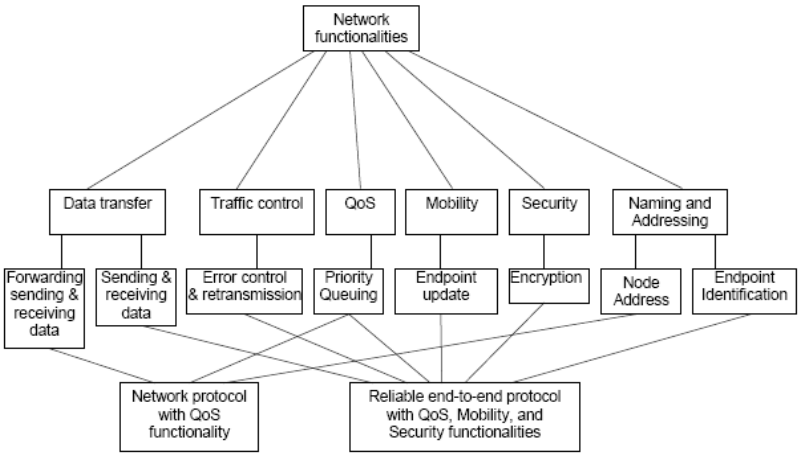

Fig. 2. Examples of functionality composition.

As illustrated in **Fig. 2**, we have in the first composition a network using an addressing scheme, with the ability of sending, receiving and forwarding data, with QoS support (e.g. by selecting the Priority Queuing discipline [20]). In the second composition, it is completely possible to select the functionalities needed to design a new reliable end-to-end protocol with QoS, Mobility, and Security functionalities. This protocol uses an endpoint identification scheme (i.e. node addresses and port numbers). The reliable transmission is provided by the Error control and retransmission functionalities. This protocol can also provide QoS because it uses Priority Queuing in the terminal to serve different connections. To provide security functionality, it integrates an encryption algorithm for user data encryption. To support mobility of endpoint, a functionality detecting endpoint changes and updating endpoint identification can be invented and integrated to the protocol.

Lessons learned from the design of protocols in the Internet [21] show that show that the composition of functionalities should follow the steps illustrated in **Fig. 3**. In the first step, network architect should consider how to identify uniquely the entities participating in the protocol or in the protocol suite. Depending on the size of the network, flat or hierarchical addressing can be chosen. The size of the address is also chosen based on the size of the network in term of number of participating entities. The architect must decide whether a completely new and independent identifier scheme needs to be defined or an existing identifier scheme can be partially reused. For example, TCP endpoint is identified uniquely within the network by a port number, the TCP protocol number and the IP address. This identifier scheme reuses

the IP address which is the identifier scheme of the IP protocol. Mobility has impact on addressing design. If the identity scheme of a protocol reuses the whole or part of the identity scheme of another protocol, the change of address happened within the underlying protocol due to mobility can break the normal operation of the protocol under design. This is the case of mobility issue with the TCP and IPsec protocols. Multi-homing protocol such as LS-SCTP (Stream Control Transmission Protocol for Load Sharing) [22] necessitates two levels of identifier, association level and path level. Virtual circuit based protocol uses two identifier schemes in a collaborative manner, address for signaling and label for data switching. When two protocols use two independent identity schemes, the architect should choose the method for addressing translation such as ARP-based method or address translation server. IP and Ethernet use ARP-based method while ATM (Asynchronous Transfer Mode) [23] and IP use address translation server. Finally, if network architect wants to use a naming scheme in conjunction with an addressing scheme, the naming scheme and the name resolution method have to be defined.

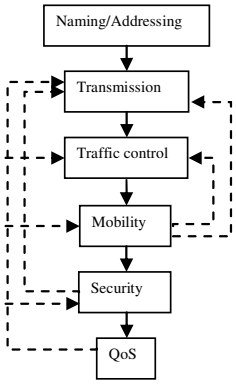

Fig. 3. Order of functionality composition.

In the second step, the transfer mode and other details related to data transmission should be defined. In this step, network architect should determine whether the protocol is byte oriented or message oriented, connection oriented or connectionless, and define the minimal PDU format (e.g. PDU length or payload length). Except for end-to-end protocol, the architect should consider the functionalities integrated into intermediate nodes such as routing, multiplexing, switching, duplicating, storing and coding. Depending on the functionality selected, necessary control information will be added into the PDU header and necessary algorithms (e.g. routing algorithm) will be integrated into the protocol as the behavior of intermediate nodes.

In the third step, traffic control mechanisms should be defined. Traffic Control is related to the way to moderate the data rate or to protect data against errors. If there can be errors during the transmission and the application does not tolerate errors, error control such as error detection (e.g. CRC - Cyclic Redundancy Check) or error recovery (e.g. FEC - Forward Error Correction) should be integrated into the protocol. Necessary behaviors of the participating entities, necessary control information carried in the protocol header, and redundant information should be integrated into the

protocol. If the application needs in order delivery, sequence number should be added. If the application does not tolerate data loss, retransmission should be defined. If there is a need to avoid overflow that the receiver or there is a need for a fair resource sharing between communications, or there is a need for control the sending rate of a communication, flow control should be integrated. If there is a need to react to network congestion, congestion control should be defined. For each type of control, there are many algorithms available along with the necessary control information to be carried in the header.

In the fourth step, mobility support should be considered. A protocol supporting mobility needs mechanisms reacting to the consequences due to mobility. If the identifier (e.g. IP address) can be changed due to mobility, binding update mechanism should be defined. Binding update can be end-to-end (e.g. SHIM6 [24] or SCTP) or endpoint-to-network (e.g. Mobile IP). If the topology is changed due to mobility, dynamic routing with route updates is necessary (e.g. ad-hoc routing protocol). During this step, it is necessary to check whether the design in steps 2 and 3 need to be adapted. For example, trigger for routing update due to mobility should be integrated into the routing protocol defined in step 2. Handover trigger should be defined. Path condition changes requiring an adaptation of flow control and congestion control parameters after a handover should be taken into account in traffic control mechanisms defined in step 3.

In the fifth step, security should be considered. Basic security services such as authentication, authorization, and data encryption should be integrated into the protocol depending on the application's needs. For each security service, mechanisms, algorithms along with necessary message exchanges should be integrated into the protocol. Many well known authentication methods, authorization mechanisms, and encryption algorithms are available in the literature. Each security service can be integrated in different steps. Authentication and authorization can be integrated into connection establishment defined in step 2. Encryption can be integrated into the transmission mechanism defined in step 2 before sending data at the receiver or intermediate nodes. Mobility should be taken into account during the design of security. Authentication may need to be triggered after a handover. For example, 802.11 terminal needs to be authenticated when changing to a new access point.

In the sixth step, QoS should be considered. As all other categories can impact the QoS of the communication, QoS design is put at the end of the composition procedure to take all these impacts into account. Different dedicated QoS mechanisms such as explicit resource reservation (e.g. Intserv-RSVP [25]) and class-based QoS provisioning (e.g. Diffserv [26]) are available. Each QoS mechanism will need additional message exchanges (e.g. reservation messages) and algorithms (e.g. scheduling algorithms such as priority queuing and weighted fair queuing) to be integrated into the protocol or the protocol suite. The design during all the last steps has impact on the QoS offered to the communication. Virtual circuit facilitates the resource reservation or resource provisioning because we know the nodes belonging to a path and these nodes can keep reservation state for the communication while it's hard to provide QoS in datagram (i.e. routing based) network because every node within the network may involve in the communication. Packet size has impact on the delay. Flow control and congestion control mechanisms have great impact on the throughput or the bandwidth offered to the communication. Retransmission increases

the delay. Handover introduces data loss. Interference reduces the bandwidth offered. Security at the same time introduces delay and increases the required bandwidth. Multi-homing can increase packet reordering. These impacts are architectural trade-offs and should be considered in this step regarding QoS requirements of the application. Refinements of the design in the previous steps (e.g. soft handover, pre-authentication, flow control and congestion control adaptation, QoS routing) should be identified and integrated into the protocol.

There exist a large number of network functionalities in both data and control planes within a network. The above discussions show that each functionality can have different implementations. Mobility can be supported by end-to-end or end-to-network binding updates. QoS can be provided by per-flow reservation or class-based priority. Security can be provided by public key or secret key encryption. The choice of specific implementations depends on the degree of the application's requirements on each feature. The interoperability between functionalities should be check during the design following the six steps described in this section especially regarding architectural trade-offs. As the relations between functionalities are complex, the functionality categories and the six steps are organized in the way that these impacts can be all checked. To minimize the protocol refinements, network architect should take important integrations between functionalities into account in advance. If the integration between security and naming/addressing is intended (e.g. Host Identity Protocol [27]), the integration can be done during the first step and only checked and refined during the fifth step. If the integration between QoS and transmission is intended (e.g. QoS routing), the integration should be considered in the second step when designing routing protocol and checked or refined during the last step.

5 Example of Protocol Design

In this section, we will use the procedure of functionality composition presented in section 4 to design a transport protocol supporting multihoming for mobile terminals. Today's mobile terminals are equipped with several interfaces. However, transport protocols widely used by applications do not support multi-homing feature. TCP and UDP do not simultaneously send data over several interfaces. If an application wants to benefit from the bandwidths coming from several interfaces, flow distribution over multiple TCP or UDP connections must be implemented within the application in an ad-hoc manner. The objective of this design is to have a new transport protocol in which the support of multihoming and mobility are in the first priority, with a minimum support of QoS, security and traffic control. It's worth noting that the specification of the requirements of the protocol and hypothesis on network infrastructure should be much more in detail and precise at the beginning. In this paper, the requirement details and hypothesis will be presented during the design to justify the design choices. In practice, network architect should make a list of detail requirements and hypothesis before starting the protocol design. If the requirements and hypothesis change, the designed protocol(s) may have fundamental changes.

In the first step, we consider the Naming/Addressing functionalities. As we are designing a transport protocol for IP networks, we can follow the design of current

transport protocol using a port number, protocol number and IP address to identify uniquely a protocol endpoint within the network. The protocol number should be carried within the IP packet while the port number is carried within the transport protocol PDU. Lesson learned from SCTP shows that for multihoming support, each endpoint should be identified by a list of IP addresses instead of a single IP address. The communication between the two endpoints is called an association. Each path within this association is identified by a pair of source IP address and destination IP address. We suppose that each interface is associated with one IP address. No new naming scheme is needed. The organization of the identifier scheme used in this protocol is shown in **Fig. 4**.

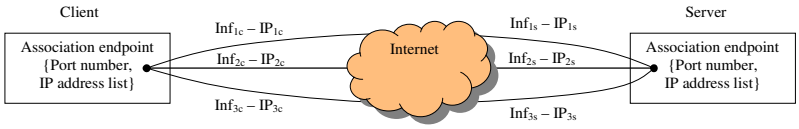

Fig. 4. Identifier scheme used in the multihoming transport protocol.

In the second step, we consider the transmission functionalities. We expect to have a multi-homing transport protocol in which data can be sent over different paths associated with different available interfaces. That means we need a flow distribution functional block which distributes data over several paths. Without much consideration about QoS, we can put in this functional block a simple scheduling algorithm which is Round-Robin in which the first PDU is sent over the first path, the next one over the second path and so on. As stream based transmission of TCP is quite complicated in an unnecessary manner in comparison with the requirements of the protocol under design, we decide to use message-based protocol like UDP. Multi-streaming feature like SCTP are judged as unnecessary. Routing or forwarding is not necessary because this is an end-to-end protocol. This is a connection oriented protocol in which a minimum amount of control information is exchanged before sending data between endpoints. The list of IP addresses of each endpoint are exchanged for the maintenance of path list in each endpoint. Message size can also be determined during this exchange if specified in the requirement list. Otherwise, a default value can be defined for the protocol.

In the third step, traffic control is considered. For the sake of simplicity, only sequence number is required for reordering packets and loss detection at the receiver. Flow control and congestion control are not needed. Simple error detection is necessary for PDU header.

In the fourth step, mobility support is considered. End-to-end binding updates learned from SCTP are selected for mobility support in order to reserve the end-to-end communication paradigm of the Internet, which can be considered as a hypothesis or requirement for the protocol design. End-to-end binding update necessitates the definition of control messages for updating with the other endpoint the changes of IP address due to a handover. This binding update helps the other endpoint to avoid sending data to an unreachable address. We can also suppose that there is not mobile IP or SHIM6 used within network. If IP address used as part of endpoint identifier never changes, binding update and mobility support in term of IP address changes are not necessary.

In the fifth step, security is considered. For example, a simple secret key-based authentication with challenge-response and a simple secret key-based encryption are required. Authentication will be integrated within the connection establishment. Data encryption is done for user data before transmission over a path.

In the sixth step, QoS is considered. Suppose that no special QoS constraint in term of delay bound, minimum bandwidth, and jitter are defined. If some QoS requirements are defined, hypothesis about the network infrastructure (e.g. whether the network infrastructure can provide some bound delay should be checked in order to determine whether the QoS requirements of an end-to-end protocol can be met by integrating some QoS mechanism. For example, jitter can be ameliorated by using a buffer at the receiver.

The protocol has been implemented in C++ [12] for the proof-of-concept of functional composition in network protocol design and for further studies on multihoming transport protocol design.

6 Towards a Theory for Network Architectures

Discussions and examples presented in the previous sections show that network functionalities can be classified and flexibly composed during network protocol design. The composition of functionalities within one protocol is quite simple and controllable. Within a protocol suite, the interaction between functionalities provided by different protocols is more complex and will be subject to another paper. Functionality composition principles integrated with existing theories such as graph theory and queuing theory may lead to a theory for network architectures in which both functionalities and performances of a network are evaluated. Tools for both simulation and prototype are needed for functional and performance evaluation. Research on Future Internet Architecture [29,30] shows that a theory for network architecture design and evaluation is needed.

Network architecture encapsulating elements with defined functionalities and communication protocols between them needs to be evaluated for both functional and performance perspectives. Conflicts or design trade-offs between functionalities should be integrated within the design and evaluation tools (e.g. simulator). An end-to-end protocol using IP address as endpoint identifier put over an IP infrastructure with the presence NAT elements should be detected as a conflict on design principle. A multihoming transport protocol wishing may not be able to sent data over a specific interface if the routing protocol at the network level is only based on the destination address.

The composition of functionalities principles integrated with graph theory can also be used to validate the functional design of a network. Two nodes which don't have direct link between them cannot be reachable one by another if there is not forwarding functionality implemented within intermediate nodes.

In a real test-bed, useful information related to network functionality composition should be available to be collected by an architectural validation program which can detect architectural conflicts or trade-offs.

Within the 4WARD project, some initial integration of functional composition principles within the GP concept and the architectural framework is in progress [11]. Within the GP concept, the functionalities are composed within the Mediation Point. In the architectural framework, the functionalities are composed by the composition of functional blocks during the Netlet and Strata design.

7 Conclusions

This paper considered the interaction and composition of network functionality as an important piece towards a theory for network architectures. Taxonomy of network functionalities and a procedure to compose them during network protocol design have been presented. A simple example of multihoming protocol design based on composition of functionality has illustrated the concept. Further directions towards a theory allowing designing and evaluating network architectures have been highlighted. In the next steps, studies on functionality composition and interaction within protocol suite and between networks are needed. Combination between the obtained principles and existing theory in communication networks is an interesting research topic. Tools for both simulation and prototype are also necessary.

Acknowledgments

This work has been partially funded by the FP7 4WARD project and the ANR 3MING project. The author would like to address special thanks to Martin Johnsson, researcher at Ericsson EAB for his useful comments and discussions within the 4WARD project's works.

References

1. H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, RTP: A Transport Protocol for Real-Time Applications, RFC 3550, July 2003.
2. C. Perkins, IP Mobility Support for IPv4, RFC 3220, January 2002.
3. S. Kent, R. Atkinson, Security Architecture for the Internet Protocol, RFC 2401, November 1998.
4. T. Dierks, C. Allen, The TLS Protocol Version 1.0, RFC 2246, January 1999.
5. R. Stewart, Stream Control Transmission Protocol, RFC 4960, September 2007.
6. R. Hinden, S. Deering, Internet Protocol Version 6 (IPv6) Addressing Architecture, RFC 3513, April 2003.
7. P. Srisuresh, K. Egevang, Traditional IP Network Address Translator (Traditional NAT), RFC 3022, January 2001.
8. B. Carpenter, Architectural Principles of the Internet, RFC 1958, June 1996.
9. G. Xylomenos, G.C. Polyzos, P. Mahonen, M. Saaranen, TCP Performance Issues over Wireless Links, IEEE Communications Magazine, Vol. 39, N°. 4, pp. 52-58, 2001.
10. P. Stuckmann and R. Zimmermann, "European Research on Future Internet Design", IEEE Wireless Communications, October 2009.

11. The FP7 4WARD project, http://www.4ward-project.eu/
12. T. Biermann and al. “Description of Generic Path Mechanism”, Deliverable D5.2.0, 4WARD project, May 2009.
13. M.Á. Callejo, M. Zitterbart and al., “Draft Architectural Framework”, Deliverable D2.2, April 2009.
14. J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, SIP: Session Initiation Protocol, RFC 3261, June 2002.
15. J. Eberspächer, H-J Vögel, C. Bettstetter, “GSM Switching, Services and Protocols”, John Wiley & Sons, 2001.
16. M. Johnsson, J. Huusko, T. Frantti, F-U. Andersen, T-M-T.Nguyen, M. Ponce de Leon, "Towards a new architecture framework - The Nth Stratum concept", Proceedings of the 4th International Mobile Multimedia Communications Conference (MobiMedia'08), Oulu, Finland, July 2008.
17. V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, H. Weiss, Delay-Tolerant Networking Architecture, RFC 4838, April 2007.
18. T. Ho, D.S. Lun, “Network Coding – An Introduction”, Cambridge University Press, 2008.
19. F. Halsall, Data Communications, Computer Networks, and Open Systems, Addison Wesley, 1996.
20. K. Nichols, S. Blake, F. Baker, D. Black, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, RFC 2474, December 1998.
21. D. F. Macedo, A. Luiz dos Santos, G. Pujolle, « From TCP/IP to Convergent Networks: Challenges and Taxonomy”, IEEE Surveys & Tutorials, Vol. 10, Issue 4, pp. 40-55, September 2008.
22. Ahmed Abd El Al, Tarek Saadawi, Myung Lee, “Bandwith Aggregation in Stream Control Transmission Protocol”, Proceedings of the 9th IEEE Symposium on Computers and Communications (ISCC”04), 2004.
23. H.G. Perros, “Connection-oriented Networks, SONET/SDH, ATM, MPLS, and Optical Networks”, John Wiley & Sons, 2005.
24. E. Nordmark, M. Bagnulo, Shim6: Level 3 Multihoming Shim Protocol for IPv6, RFC 5533, June 2009.
25. R. Braden, D. Clark, S. Shenker, Integrated Services in the Internet Architecture: an Overview, RFC 1633, June 1994.
26. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, An Architecture for Differentiated Services, RFC 2475, December 1998.
27. R. Moskowitz, P. Nikander, Host Identity Protocol (HIP) Architecture, RFC 4423, May 2006.
28. T.M.T Nguyen and X. Zhang, Composition of Functionalities Implementation for multihoming transport protocol design, http://www-phare.lip6.fr/~trnguyen
29. European Future Internet Portal, “Architecture: Future Internet”, http://www.future-internet.eu/home/future-internet-assembly/valencia-april-2010/session-agendas/architectures.html
30. J. Robert, “The clean-slate approach to future Internet design: a survey of research initiatives”, Annals of Telecommunications, Vol. 64, N° 5-6, pp. 271-276, 2009.