Automatic Handover Decision in Content Centric Networking

Yunqi Luo, Chunlei An, Jonas Eymann, Andreas Timm-Giel

Automatic Handover Decision in Content Centric Networking

Yunqi Luo, Chunlei An, Jonas Eymann, and Andreas Timm-Giel

Hamburg University of Technology
Institute of Communication Networks (ComNets)
Schwarzenberg str. 95E, 21073 Hamburg, Germany
{yunqi.luo,chunlei.an,jonas.eymann,
timm-giel}@tuhh.de

Abstract. The current Internet architecture was designed more than 30 years ago for a very different set of services than those used today. Several new architectures have been proposed for a Future Internet to better meet todays and future requirements. Content Centric Networking (CCN) is one of those new architectures gaining worldwide attention by researchers and the focus of this article. CCN is based on naming content instead of hosts allowing routers to cache popular content. It has been shown that CCN can also support point-to-point real-time conversations, for example voice or video calls. However, it has not been defined how node mobility can be achieved in such a real-time scenario with strong time constraints. In this paper a close study to the handover algorithms has been carried out. A handover decision maker, which is based on neuro-fuzzy system (NFS) is proposed and evaluated. The results show that the proposed decision maker meets the design targets.

Keywords: Future Internet, Content Centric Networking, Mobility, Handover, Adaptive Neuro-Fuzzy System

1 Introduction to Future Internet Architecture

The Internet architecture and its core protocols have proven for more than 30 years to be remarkably flexible. The Internet nowadays supports services which were never foreseen and for which it was never designed originally. However, the ever increasing number of users and the proliferation of mobile devices such as smartphones and tablets challenge some of the fundamental design principles of todays Internet [15]. The growth of routing tables and the overloading of Internet Protocol (IP) address semantics as both identifier and locator are considered key problems of the current Internet architecture [16]. To overcome these challenges, several new Internet architectures have been proposed, like the Locator/Identifier Separation Protocol (LISP) [3], the Network of Information (NetInf) [17] and Content Centric Networking (CNN) [6]. CCN is one of the approaches to overcome these challenges using the content name for addressing rather than own addresses.

LISP was proposed by CISCO and builds on the idea of using Endpoint Identifiers (EIDs) and Routing Locators (RLOCs) to decouple location and identity. This enables node mobility and supports the aggregation of RLOCs to reduce routing table sizes for better scalability.

In contrast to LISP, which is largely compatible with todays network and requires little changes, Information Centric Networks (ICN) are more radical redesigns ("clean-slate") and based on naming content instead of addressing end hosts. This approach better matches todays prevalent usage of the Internet where users are usually more interested in the content itself than where (on which exact server) the content resides. One key advantage of ICNs is the possibility of caching content wherever many users require the same content, e. g. a current news video, which saves transmission bandwidth and reduces delay.

NetInf is an example of an ICN architecture. In NetInf, an InformationObject (IO) is a unit of information such as a file which has a globally unique identifier. An IO may be stored anywhere in the NetInf network. The big advantage of NetInf is that different applications can share the data with little effort.

CCN is another ICN approach proposed by the Palo Alto Research Center (PARC) and has gained much attention over the past few years. CCN already incorporates several important aspects for a Future Internet architecture such as security or a possible transition from todays architecture to CCN. However, while the original publication claims that node mobility is inherently supported by the architecture [6], we see several problems, especially for mobile nodes running time sensitive applications such as voice calls or real-time video streaming. This paper focuses on these mobility problems in CCN, which to our knowledge have not yet been addressed.

The CCN architecture is based on naming content objects with a sub file size granularity. In CCN, each piece of content can be uniquely identified by a hierarchical name. Hosts retrieve content based on those names in a request–response manner. The unique naming of content objects enables any CCN node or router in the network to cache these content objects. Similar to peer-to-peer networks, this is especially effective for the distribution of content which many users request close to the same time. For example, if many users request a current news video, the segments of this video can be cached by the routers in the network, hence decreasing the load on the original server and at the same time improving the end user experience through a reduced response time.

The general nature of CCN thus favors multicast transmissions with several receivers of static content. Nevertheless, CCN also supports point-to-point communication such as voice calls. The suggested Voice over CCN service (VoCCN) [5] is based on the Session Initiation Protocol (SIP) [18], but does not need any proxy servers for the signaling path, in this way simplifying the call setup. However, mobility for such real-time applications was not considered in the publication.

In the Scalable and Adaptive Internet Solutions (SAIL) project [19], which is funded by the European Community's Seventh Framework Program, three different aspects of the Future Internet are investigated: Network of Information

(NetInf), Cloud Networking (CloNe) and Open Connectivity Services (OConS). OConS is a component-based architecture with three different functional entities, namely the Information Management Entity (IE), the Decision Making Entity (DE) and the Execution and Enforcement Entity (EE) [1].

The assignment of these functional entities to concrete devices or nodes can be dynamically done, depending on the communication context, i.e., a device can host one, two or all of these functional entities. With these functional elements different connectivity services can be orchestrated. These connectivity services can be on flow, network, or link level and do not necessarily follow the traditional OSI layering, but can comprise cross-layer functions. An important aspect are the open interfaces supporting the orchestration of suitable services based on new and existing mechanisms.

As illustrative examples we may have: a Router that measures the traffic, decides on the route and changes the routing table accordingly; an Access Point that measures the radio signal strengths and it sends them to a controlling node in the Core Network where a decision is taken, then this node sends a handover command to be executed at the mobile terminal and the Mobile IP Home Agent.

In general two of the most basic requirements for mobility are *Reachability* and *Continuity*. Reachability means that a user or service should be reachable independently of its current location and subnet it is attached to. Continuity, also referred to as handover, terms the fact that connections of applications should not break when a user or service moves to another location and/or changes the network. This paper first illustrates the peculiarity of mobility for real-time applications in CCN.

In this publication we refine our proposals described in [13] and present three approaches that ensure both reachability and continuity for applications in a CCN based network environment. For achieving reachability, we extended the VoCCN approach generalized the connection setup procedures for any kind of connection-oriented applications. Continuity is achieved through the presented handover procedures. Both mobility requirements are not yet supported by the current CCN architecture, but achieved by the proposals in this paper.

The main contribution of the paper are the following, 1) analyze the mobility requirements for the Future Internet, especially analyze the mobility problem of CCN, 2) summarize the exiting handover decision making schemes and establish the cost functions to evaluate the cost for the handover procedure, 3) propose an Adaptive Handover Decision Maker (AHDM) based on Neuro-fuzzy control theory.

The rest of this paper is organized as follows. The problem of mobility in CCN based networks is introduced in Section 2 and also presents state of art for using Artificial Intelligence to make a handover decision maker. An adaptive neuro-fuzzy based decision maker for CCN is proposed in Section 3. The implementation and performance for evaluation for the Adaptive Handover Decision Maker (AHDM) are shown in Section 4. Section 5 concludes the paper and gives some directions for future works.

2   Mobility in Content Centric Networking

In this section the key ideas of the Content Centric Networking are introduced, and the according mobility schemes, as well as the problems raised by these schemes are discussed afterwards.

2.1   Content Centric Networking

The communication in CCN bases on the exchange of two basic message types called *Interests* and *Data*. Data messages contain (besides the payload) a unique identifier—their name. Names are hierarchically structured in order to enable the aggregation of large collections of content with a common prefix. An Interest represents the request for one Data message and contains (at least) its name.

If a request cannot be answered locally, the Interest is forwarded to one or more neighboring nodes. The forwarding decision is based on the Forwarding Information Base (FIB), a data structure in each node which contains entries with prefixes of content names and the corresponding links to the neighboring nodes. The FIB is therefore similar to the forwarding table in IP routers. Instead, prefix entries in the FIB can point to multiple sources where content with a certain prefix might be available.

In case the FIB does not contain any prefix entry for the name of an incoming Interest, for example after node startup or reset, the Interest can be either dropped or broadcast on all available links. If an entry exists and matches, this entry is normally used, but multicast or broadcast can be used as fallback options. The entries in the FIB can be established by using routing protocols such as IS-IS or OSPF for intra-domain and BGP for inter-domain announcements of prefixes [6].

Another important aspect of CCN is security. Due to the caching capability of CCN nodes, it is required that the content itself is effectively secured against unauthorized alteration, that the receiver can verify the integrity and, to ensure privacy, that content can be encrypted. In CCN, the integrity of Data messages and names is based on cryptographic signatures which are transferred as part of the Data message. This way, any node can validate the correct binding of the Data with its name. The hierarchical structure of names can be used in this context to serve as the basis for a hierarchical public key infrastructure (PKI), so that the use of a namespace is certified by the authority controlling the superordinate namespace. For this work, we assume such a PKI and for more details on the security part of CCN refer to [20].

2.2   The problem of mobility

Though it has been shown that VoCCN works in a network with static nodes [5], several problems arise for real-time applications such as voice communication when nodes become mobile, e. g. smartphones, tablet computers or netbooks. Fig. 1 illustrates this with Mobile Node 1 (MN1) and Mobile Node 2 (MN2) in two different networks.

A bidirectional real-time communication in CCN is realized by expressing Interests at MN1 for content being created at MN2 and vice versa. Here, we assume that the application on MN2 initially tries to setup a connection to the application on MN1 (i. e. setup a call). If MN1 is not in the network where the CCN core routers FIB entries point to (Network B), a connection cannot be established until the FIB entries have been updated, e. g. by a routing protocol.

The same problem can also occur in the case of an ongoing connection: when MN1 moves to another network (step 1 in Fig. 1), all Interests from MN2 are still forwarded to Network B (step 2). As no Interests reach MN1, no Data messages (containing for example the voice data) will be sent from MN1 to MN2 and the continuity can therefore not be maintained. The connection (step 3) can only be established after the FIB entries in the routers have been updated to the new location.

While using normal CCN routing updates for FIB entries (seconds or even minutes [14]) might be acceptable in case of static content, this mechanism is clearly not sufficient to support mobility of interactive or real-time applications as it cannot guarantee continuity, and reachability is not given for considerable time periods. In [13], we proposed three proposals to solve this mobility problem. For determining the suitable time to start the handover, this paper proposes a decision maker for choosing a suitable time to make the handover decision automatically.
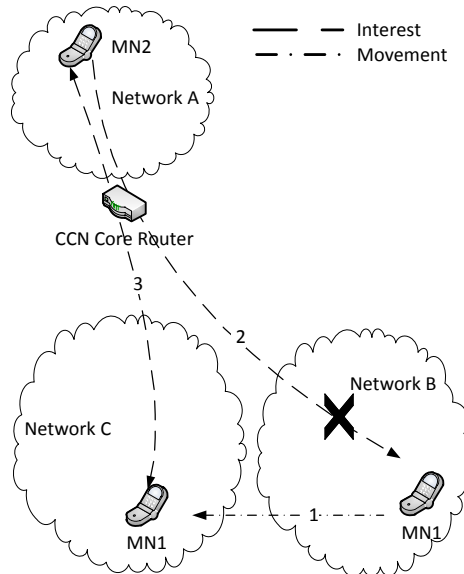

Fig. 1: Mobility problem in CCN for mobile nodes 2

2.3 Handover Decision

To prevent service discontinuity from happening, a handover over should be done proactive. There are different ways to do a handover in such a case. According to [10], there are five different ways to classify handover procedures. In this work two of them are referenced. For instance, depending on who has initiated a handover procedure, handovers can be classified as mobile-initiated handover and network-initiated handover. As the names reflect, in the first case a mobile node initiates the handover procedure, while in the latter case the network starts it. Based on the preparation that a mobile node does before start the handover procedure, handover procedures can be categorized as proactive and reactive handover types. In case of a proactive handover, both the network and the mobile nodes are aware about the handover, and some signaling can be done beforehand to assure the connection continuity between the mobile nodes the and network. The other way around, in reactive handover no signaling will be performed.

In this paper, the handover decision maker is designed based on adaptive neuro-fussy logic. This algorithm is explained in the following sections.

2.4 State of the Art

Handover is the key point maintain the reachability and continuity for end users. Handover decision proposals make decision at the suitable time to switch to another network attachment (like, access point) for vertical or horizontal handover, to guarantee the quality of service for end users.

There are several traditional strategies for making a handover decision based on different mobility characters, like cost function based proposal [4], which established the cost function models to network selection algorithm; and Bit Error Rate based proposal [11], which analyzes the performance on bit error rate and relative signal strength to make a handoff decision; Received Signal Strength (RSS) based proposal [2], which making a handover decision mainly based on RSS and focus on the hard handover for UTRAN LTE network

For the traditional handover decision maker algorithms, typically threshold based decision algorithms are used. These algorithms can cause the "ping-pong" effect. So the handover decision will occurs repeatedly when the parameters are around the handover decision thresholds. This will leads to the effect for low Quality of Experience (QoE). Hysteresis values are used to address this issue. In this case, a handover is initiated only if the value of a metric is sufficiently stronger by a hysteresis value than that of the current one. In [23] an adaptive handover algorithm is developed by dynamically determining the hysteresis value. However, the proposed algorithm has relatively higher complexity, and it is only evaluated in simulations. So the artificial intelligence based handover decision making algorithm is proposed for solving these problems.

3 Proposed Solution

3.1 Fuzzy Logic and Artificial Neural Networks

Fuzzy logic (FL) and artificial neural networks (ANN) are two main divisions in control theory. Both belong to the area of artificial intelligence. Different to the traditional logical system, fuzzy logic is more closer to human thinking logic and natural language. FL is able to convert the linguistic control strategy based on expert knowledge into an automatic control strategy. Experience shows that FL outperforms the conventional quantitative techniques when the process is too complex, or when the available information sources are interpreted qualitatively and inexactly[12].

Fuzzy algorithms consist of three parts: namely fuzzification, inference, and defuzzification. In fuzzification the input values are divided into different groups, such as negative big (nb), negative medium (nm), zero (z), positive medium (pm) and positive big (pb). In inference stage a rule base is created to map the fuzzified inputs to fuzzified outputs. The rule base must cover all permutations of the input variables [22].

The idea of artificial neural networks is inspired by biological neural networks, which are composed by numerous neural cells with limited processing power. ANN targets on bringing intelligence to computer programs, so that some complex tasks, such as pattern classification, function approximation, and prediction, can be performed by computers. An ANN usually has one input layer, one or more hidden layers, and one output layer. It can be viewed as a weighted directed graph, in which the artificial neurons are nodes, and directed edges (with weights) are connections between neuron outputs and neuron inputs. There are three main learning modes, namely supervised, unsupervised, and hybrid. In supervised learning mode, the weights are determined by learning from a set of given inputs and correct outputs. Different strategies can be used for determining the weights, such as Back Propagation (BP) and Radial Basis Function (RBF). In unsupervised mode no correct answers are provided, so the network tries to determine the weighs by studying the characteristics of the input data. Hybrid learning mode combines both approaches [7].

At the beginning of this work, both FL based and ANN based decision makers are evaluated to ease the design process of the decision maker. However, none of them provides satisfying performance. This is due to the fact that FL lacks the learning capability, hence can not adapt itself to the change of inputs. This problem is more distinct if there are multiple inputs, and each of which has numerous fuzzy levels. ANN, on the contrary, is not capable to take linguistic information (fuzzy rules) from human experts [8].

3.2 Neuro-fuzzy System

Therefore, to overcome the shortcomings of FL and ANN, another approach based on neuro-fuzzy system (NFS) is taken. NFS is a combination of neural networks and fuzzy logic, and aims to take advantage of both [8]. Generally

the process of a NFS consists of two steps: as the first step, the structure of the inputs is learned by the system to decide the fuzzy rule set. Afterwards the coefficients of each rule is tuned through parameter learning [9].

A NFS based handover is explained in the following section.

4   Implementation and Evaluation
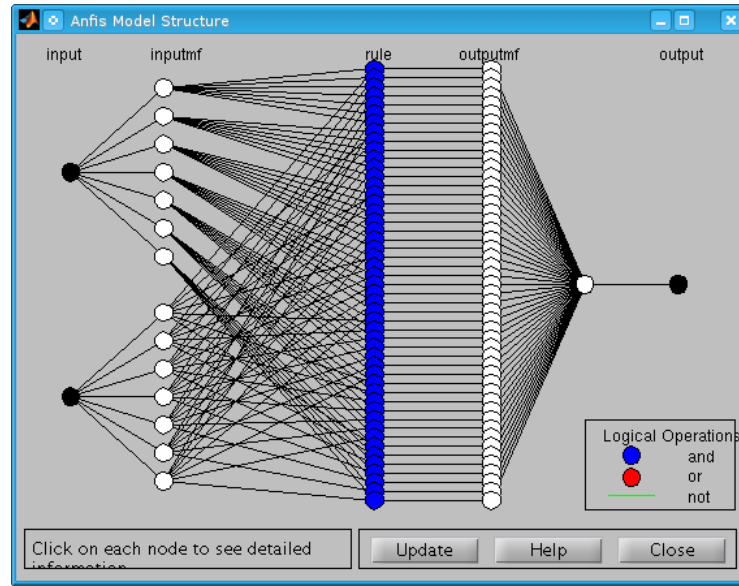
4.1   Neuro-fuzzy based Decision Maker

Fig. 2: Model Structure of the Neuro-fuzzy based Decision Maker

The design of a neuro-fuzzy based automatic handover decision maker (AHDM) is shown in Fig. 2. The model consists of five layers. The first layer, i.e. the input layer, takes two inputs. In this case the two inputs are the (link propagation) delay and the received signal strength (RSS). In the second layer, Each input is mapped to seven different fuzzy levels according to the predefined membership functions (MF). Afterward the fuzzified inputs are processed by the rules in the rule base in layer three, and computed as outputs in the following layer four, where defuzzification is performed. The final result is made by layer five as the model output.

As mentioned in Section 3, the correct answers are needed to train an ANN in supervised mode. Equation (1) - (5) describe the algorithm of deriving the corresponding correct answers for given inputs, received signal strength (RSS) and transmission delay in this case. The principle is to perform handovers only

when it is really needed, so that the overall cost is minimized (Equation (1)). The cost is composed of two parts: the extra cost of performing a handover $Cost_{Handover}$ and the cost of using a link $Cost_{Link}$ (Equation (2)). Equation (3) gives the definition of $Cost_{Handover}$. This cost is solely dependent on the delay of performing a handover $Delay_{Handover}$, whose output is scaled to the range [0 1] by the correspondent scaling function $f_s^{Handover}$. The result is further weighted by a weighting factor α, which lies in the range [0 1]. $Cost_{Link}$ is defined in Equation (4). This cost depends two parameters: the received signal strength RSS and the transmission delay of the current link $Delay_{Link}$. The higher the $Delay_{Link}$, the higher the link cost; the higher the RSS, the lower the link cost. ϵ is a small positive number, which prevents the whole link cost $Cost_{Link}$ from going to infinity in case RSS is zero. f_s^{Link} and β are the scaling function and the weighting factor, accordingly.

As discussed in [21], different scaling functions can be chosen depending on different criterion value range. In this work it is assumed that the values of all the parameters are limited by their respective minimums and maximums. Therefore the general form of scaling functions $f_s^{Handover}$ and f_s^{Link} are represented by linear functions given in Equation (5), where a and b are constants that are determined by the respective minimum and maximum.

$$Objective : min(Cost) \tag{1}$$

$$Cost = Cost_{Handover} + Cost_{Link} \tag{2}$$

$$Cost_{Handover} = \alpha * f_s^{Handover}(Delay_{Handover}) \tag{3}$$

$$Cost_{Link} = \beta * f_s^{Link}(\frac{Delay_{Link}}{RSS + \epsilon}) \tag{4}$$

$$f_s(x) = ax + b; \qquad x_{min} < x < x_{max} \tag{5}$$

4.2   Performance Evaluation

The performance of the proposed AHDM is evaluated using a set of input data. Results are shown in Fig. 3. It can be seen that in most cases the neuro-fuzzy decision maker makes correct decisions. In this scenario, one mobile user is moving on a road which passing by several access points. As mentioned in Section 4.1 before, the two inputs are RSS and delay. The end device of the mobile user can get the information of the RSS and delay when it enters the coverage of each access point. The RSS value is generated based on the distance but with some added randomness. Delay reflects the load of the according access point.

In the following figures, currently the optimized decision (the solid line) is obtained through observation. In the future work, it is to be verified using the cost functions given in Section 4.1. Here the decision generated by the threshold based method is represented by the dashed line which is shown in Fig. 4. The RSS and delay value first have been normalized between 0 and 1. The threshold of RSS is defined as 0.4, therefore all values greater then 0.4 are considered acceptable. Whereas for delay the threshold is set to 0.6, and values below 0.6 indicate that the delay is short enough. A handover decision is made if the RSS is less than 0.4, and the delay is greater than 0.6 respectively. Still sometimes incorrect decisions are made. However, most of them only result in the trigger of a slightly early/late handover, and do not degrade the overall performance much. One ping-pong effect is observed at sample 137 when using AHDM, but it significantly outperforms the threshold based decision maker (shown in Fig. 4).
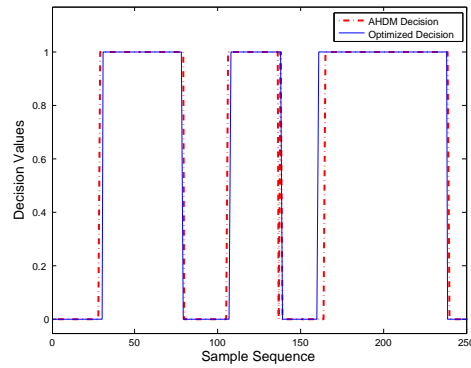

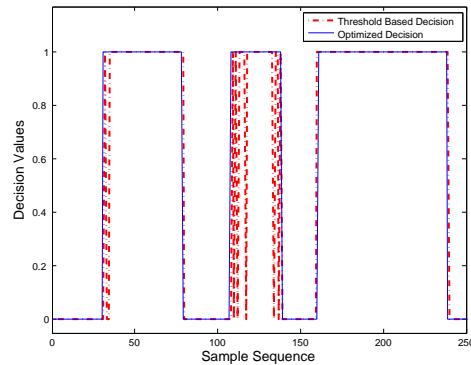Fig. 3: Test Results of Neuro-fuzzy based Decision Maker

Fig. 4: Results of a Threshold Based Decision Maker

5   Conclusions and Future Work

Continuity and reachability are amongst the most important requirements to enable mobility for real-time applications in CCN. Though the current CCN proposal eventually reestablishes reachability by normal routing updates, the suitable time to do the handover is still unclear. To have a better quality of experience of the end users, this paper proposes an automatic handover decision maker – AHDM. AHDM uses an Adaptive Neuro-Fuzzy controller to decide the suitable time to switch to different network attachments (like Access Point). To evaluate the performance for the handover decision, this paper models the cost function for mobile node. The experiments results show AHDM's decision close to the optimized solution. Also the handover decision which given by AHDM is on-line, while the optimized solution could be given only after the experiment.

However, the AHDM performance well if the character of the network parameters do not change extremely different. So for the future work, a self-learning system could be implemented for the handover decision maker, which can learn the system and adaptive itself real-time. Under this case, the new handover decision maker will be suitable for different network characters.

Also the handover decision maker in this paper can be used in OConS, where the IEs can be configured to run on CCN routers and access points, which can provide more information from the access network or even information from the core network. Then the DEs could make a better handover decision based on both the network's and the end user's QoS information.

References

1. Agüero, R., Caeiro, L., Correia, L.M., Ferreira, L.S., García-Arranz, M., Suciu, L., Timm-Giel, A.: OConS: Towards open connectivity services in the future internet. In: Sargento, S., Agüero, R. (eds.) MONAMI 2011, Third International ICST Conference on Mobile Networks & Management. Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, Springer-Verlag Berlin, Heidelberg, Germany (Sep 2011)
2. Anas, M., Calabrese, F., Mogensen, P., Rosa, C., Pedersen, K.: Performance evaluation of received signal strength based hard handover for UTRAN LTE. In: Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th. pp. 1046 –1050 (Apr 2007)
3. CISCO: The Locator Identifier Separation Protocol (LISP) (July 2010), `http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_11-1/111_lisp.html`, accessed 26 March 2012
4. Hong, K., Lee, S., Kim, L., Song, P.: Cost-based vertical handover decision algorithm for WWAN/WLAN integrated networks. EURASIP J. Wirel. Commun. Netw. 2009, 15:1–15:11 (2009), `http://dx.doi.org/10.1155/2009/372185`
5. Jacobson, V., Smetters, D.K., Briggs, N.H., Plass, M.F., Stewart, P., Thornton, J.D., Braynard, R.L.: VoCCN: Voice-over Content-Centric Networks. In: Proceedings of the 2009 Workshop on re-architecting the Internet. pp. 1–6 (2009)
6. Jacobson, V., Smetters, D.K., Thornton, J.D., Plass, M.F., Briggs, N.H., Braynard, R.L.: Networking Named Content. In: Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies. pp. 1–12 (2009)

7. Jain, A., Mao, J., Mohiuddin, K.: Artificial neural networks: a tutorial. Computer 29(3), 31 –44 (Mar 1996)
8. Jang, J., Sun, C.: Neuro-fuzzy modeling and control. Proceedings of the IEEE 83(3), 378 –406 (Mar 1995)
9. Juang, C., Lin, C.: An online self-constructing neural fuzzy inference network and its applications. Fuzzy Systems, IEEE Transactions on 6(1), 12 –32 (Feb 1998)
10. Kojo,   M.,   Manner,   J.:   Mobility   related   terminology. http://tools.ietf.org/html/rfc3753, `http://tools.ietf.org/html/rfc3753`
11. Kumar, P., Holtzman, J.: Analysis of handoff algorithms using both bit error rate (BER) and relative signal strength. In: Universal Personal Communications, 1994. Record., 1994 Third Annual International Conference on. pp. 1 –5 (Oct 1994)
12. Lee, C.: Fuzzy logic in control systems: fuzzy logic controller. i. Systems, Man and Cybernetics, IEEE Transactions on 20(2), 404 –418 (Apr 1990)
13. Luo, Y., Eymann, J., Angrishi, K., Timm-Giel, A.: Mobility Support for Content Centric Networking: Case Study. In: LNICST 97. pp. 76–89 (Sep 2011), mONAMI 2011 conference, Aveiro, Portugal, ISBN 978-1-936968-16-9
14. Malkin, G., Harkin, A.: TFTP Timeout Interval and Transfer Size Options. RFC 2349 (Draft Standard) (May 1998), `http://www.ietf.org/rfc/rfc2349.txt`, accessed 26 March 2012
15. de Meer, H., Hummel, K.A., Basmadjian, R.: Future Internet services and architectures: trends and visions. Telecommunication Systems (Mar 2011)
16. Meyer, D., Zhang, L., Fall, K.: Report from the IAB Workshop on Routing and Addressing. RFC 4984 (Informational) (Sep 2007), `http://www.ietf.org/rfc/rfc4984.txt`, accessed 26 March 2012
17. NetInf: Network of Information (2011), `http://www.netinf.org/home/home/`, accessed 26 March 2012
18. Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., Schooler, E.: SIP: Session Initiation Protocol. RFC 3261 (Proposed Standard) (Jun 2002), `http://www.ietf.org/rfc/rfc3261.txt`, accessed 26 March 2012
19. SAIL: Scalable and Adaptive Internet Solutions (SAIL) (2011), `http://www.sail-project.eu/`, accessed 26 March 2012
20. Smetters, D.K., Jacobson, V.: Securing Network Content. Tech report, PARC (October 2009)
21. Wenning, B.: Context-Based Routing in Dynamic Networks. Vieweg and Teubner, 2010 edn. (Aug 2010)
22. Yanik, P., Ford, G., McDaniel, W.: An introduction and literature review of fuzzy logic applications for robot motion planning. In: In Proceedings of ASEE Southeast Section Conference (2010)
23. Zhu, H., Kwak, K.S.: An adaptive hard handoff algorithm for mobile cellular communication systems. ETRI Journal 28(5) (2006)