Cognitive Information Service: Basic Principles and Implementation of A Cognitive Inter-Node Protocol Optimization Scheme

Dzmitry Kliazovich[3], Fabrizio Granelli[1], Nelson L. S. da Fonseca[2], Radoslaw Piesiewicz[3]

[1] DISI – University of Trento
Via Sommarive 14, I-38050 Trento, Italy
E-mail: granelli@disi.unitn.it

[2] State University of Campinas
Av. Einstein, 1251, Campinas, Brazil
nfonseca@ic.unicamp.br

[3] CREATE-NET
Via Alla Cascata 56D, Trento, Italy
[kliazovich,piesiewicz]@create-net.org

Abstract—**Cognitive networks are becoming extremely popular in the network domain. This paper proposes a novel concept in cognitive network management and protocol configuration, where any protocol of the TCP/IP protocol reference model can be extended to dynamically tune its configuration parameters based on "immediate past" performance. The approach is focused on inter-node cognitive adaptation which is fostered by the proposed Cognitive Information Service (CIS). Performance evaluation results are obtained for cognitive adaptation of main TCP flow control parameters and show good agreement with design objectives.[1]**

Keywords-cognitive networks, cognitive pilot channel CPC, cognitive information exchange and optimization

I. INTRODUCTION

Cognitive networking, motivated by evolution of communications towards an increased use of wireless technologies, is aimed at overcoming inefficiency in network setup, configuration and management [1].

Being different from cognitive radios [2], cognitive networks offer performance optimization for end-to-end services using adaptation and learning techniques. The following paragraphs introduce some relevant efforts in the framework of cognitive network design.

Literature overview: Initially, research projects in cognitive networks considered Beyond-3G (B3G) network architectures, allocating complete control to the network core and enabling easy introduction of additional functionalities. Research projects in this category are E^2R [3] and m@ANGEL platform [4].

The focus of E^2R is on all-IP network fully integrated with reconfigurable equipment with the assumption of the availability of simultaneous reconfigurability support at all the protocol stack layers for the involved devices.

In m@ANGEL, a cognitive process is considered to be implemented in the access part of the network, between base stations and mobile users. Each m@ANGEL entity is responsible for monitoring, resource brokerage, goals management, and reconfigurable element control functionalities.

Differently from B3G-focused approaches, cognitive node architecture presented in [9] separates reconfigurable functions at node level, while outsourcing cognitive engine implemented into the network.

Another approach, called CogNet [6], proposes an architecture where each TCP/IP protocol layer is extended with a software interface allowing monitoring, control, and coordination performed at the cognitive plane.

All approaches described above are architecturally focused on distributed cognitive engine implementation. Moreover, both B3G and pure IP approaches limit the scope of the distribution to a provider's network or segment of the network.

Focusing on the ways cognitive information is distributed within the network segment the research consortium of E^2R project defined the notion of Cognitive Pilot Channel (CPC) [5]. In its initial configuration, CPC is a channel which operates in certain geographical areas and carries information about operators, access networks, and spectrum frequencies allocated to terminals. As a result, CPC assists mobile terminals lifting the requirement to scan the whole spectrum. Indeed, CPC is completely focused at distribution of physical layer information – including geographical coordinates, description and assigned frequency ranges of access networks operating in the region, as well as maximum transmitted power levels.

Main contribution: In this paper, we extend the notion of CPC by introducing the Cognitive Information Service (CIS) aimed at the organization of a logical channel in a local

[1] This work has been partially supported by the Italian National Project: Wireless multiplatfOrm mimo active access netwoRks for QoS-demanding muLtimedia Delivery (WORLD), under grant number 2007R989S

978-1-4244-4148-8/09/$25.00 ©2009

provider network in order to provide cognitive information exchange at all protocol layers among network nodes. Such service enables collaboration and operational data aggregation in the spatial domain, which represents a core requirement for cognitive network approaches.

Paper organization: Section II presents the core of the proposed CIS approach outlining its basic principles and operation; Section III outlines implementation details of the proposed approach as well as presents performance evaluation results for cognitive tuning of window evaluation parameters for TCP protocol; Section IV underlines conclusions and outlines future work of the topic.

II. COGNITIVE INFORMATION SERVICE

A. Motivation Example

Configuration of communication protocols impacts the overall performance of applications and data transfer services. However, it is not always possible to choose the right protocols and optimal setup of their configuration parameters due to unpredictable characteristics of network state.

For example, focusing on TCP protocol, the choice for Initial Window Congestion (w) size has changed over time. Original TCP specification [12, 13] defined the size of w equal to 1 TCP segment. Later, due to a bug in NetBSD implementation, was increased to 2 segments [10]. Adapted in April 1999, RFC 2581 allowed w of both 1 and 2 segments. The most recent specification from 2002 [11] permits upper bound for w of roughly 4K, which is equivalent to 3 TCP segments. In [11], the authors propose w size as large as 4 segments in order to speed up TCP startup performance.

In steady state, TCP performance is defined by window evolution algorithm and its configuration parameters α and β [13]. For example, in state-of-the-art TCP NewReno [14] α component of linear increase is equal to 1 and β component defining aggressiveness of multiplicative decrease is fixed at ½ halving congestion window every time the loss is detected. This relation is modified in Scalable TCP [15] which operates with $\alpha = 0.01w$ and $\beta = 0.125$. Other, approaches like FAST [16], HighSpeed TCP [19], Compound TCP (CTCP) [20], and BIC-TCP [17] try to scale α and β parameters with current size of the window or Round-Trip Time (RTT) components of the connection.

Summarizing, many approaches exist for tuning basic TCP startup and steady-state parameters, but none of them is recognized as a true winner over others. This is mainly due to the fact that proper choice of initial window and window evolution algorithm largely depends on the dynamics of network parameters (bandwidth, delay, loss rate, etc.) defining the end-to-end heterogeneous path between the sender and receiver nodes. Unfortunately, such information is not available prior to connection establishment and is difficult to estimate reliably [17].

A previously proposed approach by the authors, called CogProt, tries to solve this problem using cognitive approach by constantly trying different values for the parameters of interest and choosing the best configuration based on the history of performance experience. Using CogProt approach each network node is able to converge to the suboptimal configuration of its protocol stack. However, CogProt does not make considerations on interactions between nodes which can be solved by the proposed approach.

B. CIS Network Service

The introduction of the Cognitive Information Service (CIS) in the local network is presented in Fig. 1.

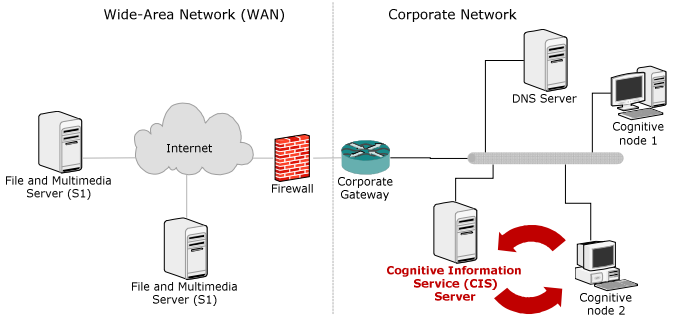

Figure 1. Cognitive Information Service (CIS) operation in the network.

CIS provides the means for network nodes to locally exchange cognitive information they obtained at all the protocol layers.

The proposed CIS architecture enables three types of cognitive functionalities:

- *Intra-layer cognitive functionality* refers to cognitive algorithms operating with a single or several protocols configuration parameters logically located within the boundaries of a single protocol layer. A good example is a cognitive adaptation of TCP startup and steady state parameters based on the past (previous flows) or current (parallel flows) performance.

  However, maximizing throughput performance of one protocol could give a negative effect on another protocol at the same node which may have different the throughput/goals.

- *Intra-node cognitive functionality* aims at conflict resolution and joint optimization among layers. For example, TCP window evolution may be limited with a maximum bandwidth available in the cell or network segment, which can be estimated at the physical/link layer. Furthermore, intra-node cognitive engine is responsible for tracking the diversity of application goals. For example, it may limit the throughput of outgoing file transfer flows to unload buffers if VoIP flows get out of delay budget.

  With intra-node cognitive engine it is possible to achieve a local performance maximum for a single node and all its applications. However, an optimal configuration of single nodes does not necessarily imply optimal performance for the whole network sector.

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

- *Inter-node cognitive functionality* offers spatial dimension of cognitive optimization between network nodes: the nodes entering the network can ask for optimal parameter configuration from other nodes and then adjust them dynamically during operation.

The assumption that configuration offered by CIS to a newly entered node to the network will be optimal is not necessarily true, since this configuration was made without the knowledge of node's running applications, traffic demands, and peers it is willing to communicate. However, such configuration represents a good temptative and will potentially offer better startup performance than using fixed default values for the protocols.

Table I contains examples of protocol stack parameters and controlled quality metrics that can be of interest for cognitive optimization of the three types presented above.

Application layer provides the environment of running user applications and is concerned aboutperformance for file transfers, quality of voice and video flows. Transport layer is generally represented by Transmission Control Protocol (TCP), which performance is commonly measured using flow goodput, depending on congestion window (or outgoing rate) evolution. Network layer is responsible for routing packets across interconnected networks, hence the main performance metric can correspond to the quality of the selected route in terms of the number of hops or end-to-end delay. The quality of the link layer is determined by the data rate it offers to higher layers as well as medium access delay. Most of the link layer configuration parameters depend on the underlining technology while some of the shared parameters include contention window (*cwnd*), fragmentation threshold (MTU), and retransmission configuration for wireless network. Physical layer basically controls the tradeoff between transmission power, type of modulation, coding, and achieved data.

TABLE I. TABLE TYPE STYLES

Protocol layer	Protocol name	Parameter	Metric
Application	File Transfer (FTP)	Number of parallel transfers	FTP Goodput
	VoIP	• Coding rate • Coding interval • FEC strength	• Mean Opinion Score (MOS)
	Video Streaming	• Streaming bitrate • Frame Rate • Keyframe interval	• Peak signal-to-noise ratio (PSNR) • Structural Similarity (SSIM) • Video Quality Measurement (VQM)
Transport	Transmission Control Protocol (TCP)	• Congestion window (w) • Slow start threshold (ssthreshold) • Agressiveness of window increase (α) and decrease (β) • Protocol version	• Flow Goodput
Network	Routing	• Routing type	• Route setup delay • End-to-end delay
Link	MAC	• Contention window (cwnd) • Fragmentation threshold (MTU) • Retransmission scheme	• Data rate provided to higher layers • Medium access delay
Physical	PHY	• Transmit rate, power • Type of modulation, coding • Frequency channel	• Transmission rate • Bit error rate

C. Signaling Methods

Depending on the nature and requirements for information exchange, three different signaling methods can be designed for CIS: in-band signaling, on-demand signaling, and broadcast signaling.

In-band signaling is the most effective signaling method from the point of overhead reduction point of view. Cognitive information can be encapsulated into ongoing traffic flow, for example into optional packet header fields [9], and delivered without the waste of bandwidth resources.

Due to low overhead in-band implementation of CIS is best suited for networks with wireless technologies in the access such as WiFi, WiMAX, and cellular where the bottleneck of the end-to-end connection is typically at the wireless link.

Another advantage of in-band signaling is that cognitive information can be associated with a piece of application data it is delivered with.

However, the main shortcoming of in-band signaling is in the limitation of signaling to the direction of the packet flow, making it not suitable for cognitive schemes requiring instant communication between nodes having no ongoing data exchange.

On-demand signaling method operates at the request-response basis and can be complementary for in-band signaling. It is designed for cases requiring instant cognitive information delivery between network nodes. Cognitive information becomes available at the requesting node following round-trip time delay which makes it well suited for Wired/Wireless LAN scenario.

One of the core signaling protocols considered in on-demand signaling is Internet Control Message Protocol

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

(ICMP). Generation of ICMP messages is not constrained by a specific protocol layer and can be performed at any layer of the protocol stack. However, signaling with ICMP messages involves operation with heavy protocol headers (IP and ICMP), checksum calculation, and other procedures which increase processing overhead.

Broadcast signaling method allows point-to-multipoint cognitive information delivery from CIS server to the network nodes located in the same segment while keeping overhead at the low level. Broadcasting is especially suited for wireless networks following cellular organization.

Cognitive information is encapsulated into a beacon periodically broadcasted by wireless gateways (access points or base stations), and thus fits scenarios where cognitive information delivery is tolerant to delays and can be performed in regular intervals.

III. PERFORMANCE EVALUATION

A. CIS implementation

We implemented Cognitive Information Service (CIS) as an extension of ns2 network simulator [18]. We focused on cognitive adaptation of the main parameters controlling TCP protocol steady state behavior, i.e. speed of linear window increase α and aggressiveness of multiplicative decrease β. As a result, this implementation dynamically controls the window evolution parameters of TCP NewReno - adapting them during runtime.

At the beginning of each flow, α and β are set to their default values (1 and ½, respectively). Then, after TCP slow start is finished and congestion avoidance begins, a timer is started which provides sampling for our cognitive engine implemented inside each network node (corresponds to intra-layer cognitive functionality). Whenever the timer expires, intra-layer cognitive engine performs steps defined by Algorithm 1. First, it computes, analyses, and stores throughput performance metric weighted with relevance parameter. Then, it selects α and β values corresponding to optimal throughput performance and initializes with them mean values of two random generators of normal distribution. Finally, these generators are used to obtain α and β values that will be used in the next sampling interval.

The inter-node level implementation of cognitive engine, presented as Algorithm 2, follows a similar approach. At regular intervals, each cognitive node communicates currently chosen α and β values and the weighted throughput performance evidenced during immediate past experience to the CIS service. This communication is implemented using ICMP protocol, introducing relevant signaling overhead. However, typically CIS service is considered to be implemented in the same network segment with cognitive nodes and should not lead to performance degradation of data flows. Nevertheless, the interval used for cognitive nodes to report to CIS should be chosen carefully considering possible overhead issues.

Algorithm 1: Intra-layer Cognitive Engine

Analyze Performance Metrics

1. **Get** number of bytes received since last time interrupt nBytes
2. **Get** time elapsed from the last timer interrupt smapleInterval
3. **Calculate** average throughput R_i as nBytes/sampleInterval
4. **Calculate** weighted throughput $R_i = R_{i-1} \times (p) + R_i \times (1-p)$, where p is a weight given to history significance
5. **Store** weighted Ri in two-dimensional array on a position defined by current α and β values [α, β]

Get parameters corresponding to optimal performance

6. **Set** Max Throughput to zero
7. **For** each element of two-dimensional array with Ri **do**
8. **If** current throughput Ri is greater than Max Throughput **then**
9. **Set** Max Alpha equal to α
10. **Set** Max Beta equal to β
11. **Endif**
12. **Endfor**

Choose parameters for next sampling interval

13. **Set** Normal Distribution Mean to Max Alpha
14. **Get** new α from the distribution
15. **Set** Normal Distribution Mean to Max Beta
16. **Get** new β from the distribution

Algorithm 2: Inter-node Cognitive Engine

Receive feedback and obtain cumulative throughput maximum

17. **Set** Total Throughput R_{TOT} equal to zero
18. **For** each cognitive node **do**
19. Receive α, β, and measured Throughput Ri
20. Increase R_{TOT} for Ri
21. **Endfor**
22. **Store** R_{TOT} along with α and β parameters for each node
23. **Get** max R_{TOT} and α_{max} and β_{max} parameters corresponding for every cognitive node involved into R_{TOT} calculation

Configure each cognitive node parameters

24. **For** each cognitive node **do**
25. **If** current α is not equal to α_{max} **then**
26. **Set** α equal to α_{max}
27. **Endif**
28. **If** current β is not equal to β_{max} then
29. Set β equal to β_{max}
30. **Endif**
31. **Endfor**

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

B. Scenario Description

Simulated network topology is presented in Fig. 2. It consists of four cognitive nodes *S* running intra-layer cognitive engine and CIS server performing inter-node cognitive operations all connected using 100 Mb/s, 0.1 ms links in star topology with router R1. Such connectivity aims to mimicing operation of Ethernet network segment. Similar configuration is followed by destination nodes *D*, which do not implement any cognitive functionality.

Routers R1 and R2 are connected with four links *L* with propagation delays of 10ms, 50ms, 100ms, and 200 ms, and Packet Error Rates (PERs) of 0.0, 0.05, 0.1, and 0.15.

A maximum four flows can be started in simulated topology between corresponding *S* and *D* nodes routed though corresponding the link *L*. For example, the first flow is initiated between S1 and D1 and flows through L1 link.

Different values of link PERs will require different window increase strategies, controlled by parameter α, from the flows to obtain optimal performance. High values of α are expected to bring better throughput for high PERs. However, in case of no errors, high values of α will lead to multiple congestion-related losses and throughput degradation due to retransmission.

Different RTTs are designed to influence the choice of both α and β parameters through different time required for S nodes to react to congestion- or link-related losses.

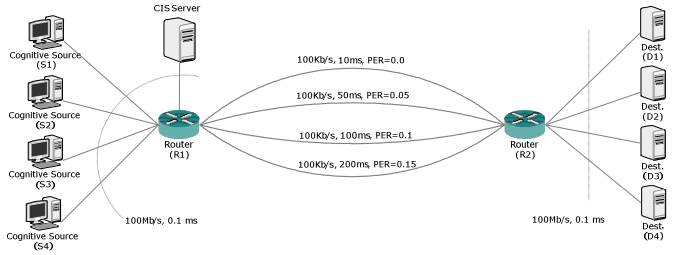

Figure 2. Simulated topology.

C. Intra-layer Cognitive Optimization Results

In order to obtain the details of inter-layer cognitive optimization, we first limited our scenario to only two TCP connections simulated in separate experiments: flow F1 between S1 and D1 involving L1 link and flow F3 between S3 and D3 involving L3 correspondingly. Average flow throughput is chosen as the main performance metric which is measured for different values of α with β constantly fixed at ½ for this experiment. We used 1s as sampling interval for cognitive engine and 0.5 for standard deviation parameter for random generator producing normal distribution.

The results presented in Fig. 3 show the ability of cognitive engine to always converge to the optimal value of parameter α. For F1, the optimal value of α is equal to 1 due to the absence of link errors - since every packet loss is caused by congestion; while for F3 the optimal value of α is 4, which corresponds to the balance between increase rate after loss related to link errors and the amount of retransmissions performed due to multiple congestion-related losses in the bottleneck buffer.

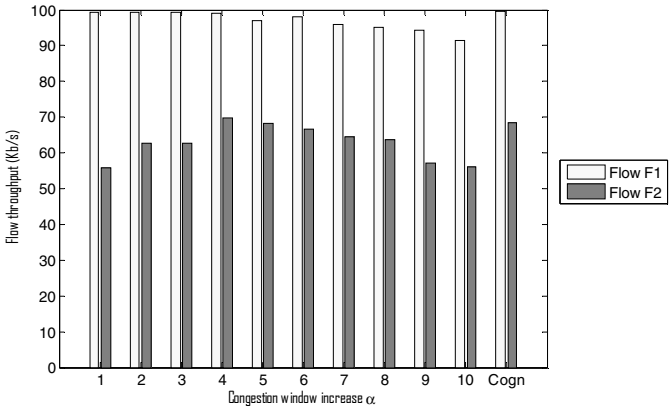

Figure 3. Average throughput performance of a single TCP flow with different fixed congestion window increase (α) parameters and cognitive adaptation approach.

Fig. 4 presents the distribution of α values chosen by intra-layer cognitive engine during a TCP flow runtime. Each value is obtained as the time at which TCP flow congestion control is using a given value of alpha divided by the total simulation time. As expected for the flow F1 most of the chosen α values group around α=1, while flow F3 spends most of the time (almost a half of simulation time) with α=4.

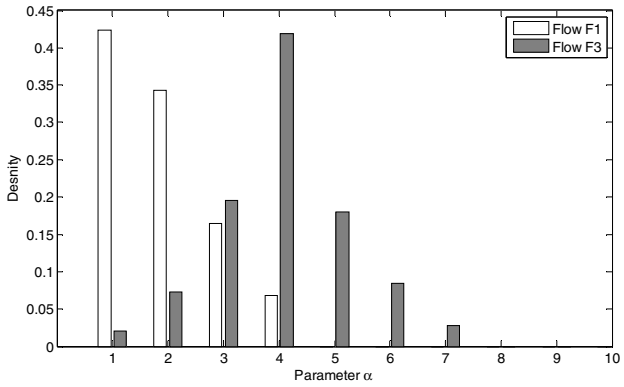

Figure 4. Density of congestion window increase (α) parameters chosen by intra-layer cognitive engine.

D. Inter-node Cognitive Optimization Results

Previous section outlines performance benefits of using intra-layer cognitive engine for tuning single TCP flow performance. In this section, we focus on joint performance optimization of multiple TCP flows. To this aim, we simulated four TCP NewReno flows between corresponding *S* and *D* nodes allocated on the different links *L* (see Fig. 2).

Fig. 5 presents throughput performance results for each individual flow as well as cumulative throughput for the following three cases: i) no cognitive adaptation, ii) with intra-layer cognitive adaptation, and iii) with inter-node cognitive adaptation (CIS server). In both cases when cognitive

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

adaptation is used, it is performed simultaneously on α and β TCP flow control parameters as outlined in the section above.

As expected, the throughput decreases for long links with high error rates. The main reasons for such throughput reductions are link errors and well-known RTT unfairness for flows with different RTTs competing in the same buffer.

It can be observed that intra-layer cognitive engine can easily solve the problem of link losses by adapting α and β parameter and converging to the optimal throughput for each flow. However, it cannot cope with the problem of RTT unfairness, which requires coordination between flows. Such coordination is performed at inter-node level by CIS server, leading to higher performance of individual flows as well as cumulative performance.

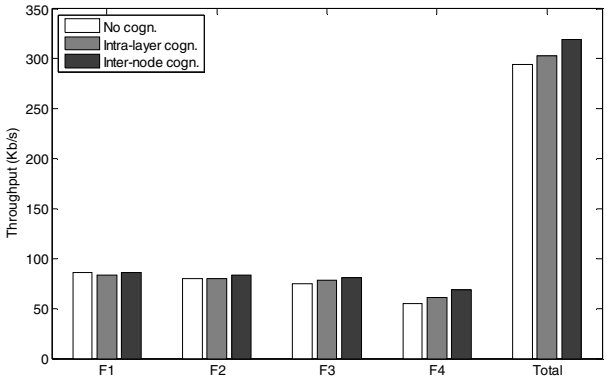

Figure 5. Multiflow TCP throughtput performance for case with no cognitive adapatation, intra-layer cognitive adaptation, and inter-node cognitive adaptation.

IV. Conclusions and Future Work

In this paper a novel concept is introduced which brings cognitive algorithms into network management and protocol configuration. Based on the proposed concept, any protocol from the TCP/IP protocol reference model can be extended to dynamically tune protocol configuration parameters based on immediate past performance.

The paper focuses on inter-node cognitive adaptation and presents the Cognitive Information Service (CIS), which is designed to foster operation of cognitive algorithms at inter-node level.

Performance evaluation validates the implementation of intra-layer and inter-node (CIS service) cognitive algorithm for dynamic tuning of main parameters controlling runtime TCP congestion window evolution. The results obtained demonstrate good agreement with the design objectives.

Future work will be focused on further evaluation of the proposed techniques and study of its application to other protocols.

Acknowledgement

The EUWB consortium would like to acknowledge the support of the European Commission partly funding the EUWB project under Grant Agreement FP7-ICT-215669.

References

[1] R. W. Thomas, D. H. Friend, L. A. DaSilva, and A. B. MacKenzie, "Cognitive networks: adaptation and learning to achieve end-to-end performance objectives," IEEE Communications Magazine, vol. 33, no. 12, pp. 51 – 57, December 2006.

[2] J. Mitola III, "Cognitive radio for flexible mobile multimedia communications," Mobile Networks and Applications, vol. 6, no. 5, Sep 2001.

[3] D. Bourse, S. Buljore, A. Delautre, T. Wiebke, M. Dillinger, J. Brakensiek, K. Moessner, K. El-Khazen, and N. Alonistioti, The End-to-end Reconfigurability (E2R) Research, SDR Forum Technical Conference, Orlando, USA, 2003.

[4] P. Demestichas, V. Stavroulaki, D. Boscovic, A. Lee, and J. Strassner, m@ANGEL: autonomic management platform for seamless cognitive connectivity to the mobile internet, IEEE Communications Magazine, vol. 4, no 6, pp. 118- 127, June 2006.

[5] "End to End Reconfigurability Phase II (E2R2)", IST Project, http://e2r2.motlabs.com

[6] B. S. Manoj, R. R. Rao, and M. Zorzi, "CogNet: a cognitive complete knowledge network system," IEEE Wireless Communications, vol. 15, no. 6, pp. 81 – 88, December 2008.

[7] J. Perez-Romero, O. Salient, R. Agusti, and L. Giupponi, "A Novel On-Demand Cognitive Pilot Channel Enabling Dynamic Spectrum Allocation," 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN), pp. 46 – 54, April 2007.

[8] Q. Wang and M. A. Abu-Rgheff, Cross-layer signaling for next-generation wireless systems, IEEE Wireless Communications and Networking (WCNC), pp. 1084 – 1089, 2003.

[9] M. Allman, "Fixing Two BSD TCP Bugs," Technical Report CR-204151, NASA Lewis Research Center, October 1997.

[10] M. Allman, S. Floyd, and C. Partridge, "Increasing TCP's Initial Window," RFC 3390, October 2002.

[11] R. Brade, "Requirements for Internet hosts – Communication Layers," RFC 1122, October 1989.

[12] R. Stevens, "TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms," RFC 2001, January 1997.

[13] S.Floyd, T.Henderson "The NewReno Modification to TCP's Fast Recovery Algorithm," RFC 2582, Apr 1999.

[14] T. Kelly, "Scalable TCP: improving performance in highspeed wide area networks," ACM Comput. Commun. Rev., vol. 33, pp. 83 – 91, April 2003.

[15] D. X. Wei, C. Jin, S. Low, and S. Hegde, "FAST TCP: Motivation, Architecture, Algorithms, Performance," IEEE/ACM Transactions on Networking, vol. 14, no. 6, pp. 1246 – 1259, December 2006.

[16] L. Xu, K. Harfoush, I. Rhee, "Binary increase congestion control (BIC) for fast long-distance networks," Proc. of the IEEE INFOCOM, Hong Kong, 2004.

[17] R. S. Prasad, M. Murray, C. Dovrolis, and K. C. Claffy, "Bandwidth Estimation: Metrics, Measurement Techniques, and Tools," IEEE Network, November/December 2003.

[18] The network simulator ns2. Available from: http://www.isi.edu/nsnam/ns.

[19] S. Floyd, "HighSpeed TCP for Large Congestion Windows," RFC 3649, Experimental, December 2003.

[20] M. Sridharan, K. Tan, D. Bansal, and D. Thaler, "Compound TCP: A New TCP Congestion Control for High-Speed and Long Distance Networks," Experimnetal Internet Draft, January 2008.

978-1-4244-4148-8/09/$25.00 ©2009