Contents

1 Architectures and Protocols for Next Generation Cognitive Networking
B. S. Manoj, Ramesh R. Rao,, Michele Zorzi 1
1.1 Introduction 1
1.2 Definition of Cognitive Networking 2
1.3 Architectures for Cognitive Networking 2
1.3.1 Autonomous Cognitive Networking 3
1.3.2 Distributed Cognitive Networking 7
1.4 CogNet: Cognitive Complete Knowledge Network 8
1.4.1 CogPlane and CogBus 10
1.4.2 Case Study: CogTCP 11
1.5 Summary 15
References 16

Index 17

List of Contributors

B. S. Manoj
Department of Electrical and Computer Engineering

University of California San Diego, CA 92093

bsmanoj@ucsd.edu

Ramesh R. Rao
Department of Electrical and Computer Engineering

University of California San Diego, CA 92093

rrao@ucsd.edu

Michele Zorzi
Department of Information Engineering

University of Padova, Italy

zorzi@die.unipd.it

1

Architectures and Protocols for Next Generation Cognitive Networking

B. S. Manoj, Ramesh R. Rao, and Michele Zorzi

[1] Department of Electrical and Computer Engineering, University of California San Diego, CA 92093
bsmanoj@ucsd.edu, rrao@ucsd.edu
[2] Department of Information Engineering, University of Padova, Italy,
zorzi@dei.unipd.it

Summary. In the late 1990s the introduction of cognitive radios paved the way for physical layer capabilities in an entirely new dimension, namely cognition. Since then, significant amount of research has gone into software defined radios and cognitive radio systems. Recent innovations in physical layers for network interface cards and base station front-ends for cellular systems utilize the benefits of cognitive radio systems. While cognitive radio systems are well researched, less-studied cognitive networking appears to have significant potential to be part of next generation wireless systems. In this chapter, we present some benefits of cognitive wireless networks, different architectural paradigms for such systems, and a new architecture specifically designed for cognitive wireless networks. This chapter also presents the details of the CogNet AP, a cognitive network access point, as an early implementation of autonomous cognitive networking systems.

1.1 Introduction

A cognitive radio [8, 9] is different from traditional Software Defined Radios (SDRs). SDRs provide in software the radio frequency (RF) processing functions, for example, waveform synthesis, traditionally implemented in hardware, thereby making reconfiguration of the radio properties very easy. On the other hand, cognitive radios are intelligent radio devices which can learn their own capabilities, radio environment, user behavior, and the physical environment in order to execute complex adaptations and configure themselves to best suit the situation. Such devices can even alter their physical layer interfaces from one access technology to another by over-the-air downloading a new software-defined waveform. While software radios are fairly well understood, cognitive radios are actively under research. On the other hand, cognitive networking is in its early infancy. In most cases, except for MAC issues, there is relatively little focus on networking in general and the overall network system in particular.

1.2 Definition of Cognitive Networking

Clarke *et al.* provided in [2] the first definition of cognitive capability for the Internet where a knowledge plane is assumed to have significant cognitive capability and to be able to build and maintain high level models of what the network is expected to do such that it can receive and execute high level instructions from the network administrators and report the result of such actions. The cognition capability expected from the cognitive plane is also expected to enable self correction and reconfiguration of the Internet if unexpected behaviors or failures happen. Such a system has the ability to translate the high level instructions to executable low level actions. The strategies suggested would involve the use of a unified approach which includes several system components with a global view of the events in the network. While the suggestions in [2] were useful to initiate potential benefits of using a cognitive paradigm for the Internet, no systems have been built or studied yet based on those concepts.

An important work in this direction was done by Thomas *et al.* in [10]. They provided the following definition for cognitive network systems: "A cognitive network has a cognitive process that can perceive current network conditions, and then plan, decide and act on those conditions. The network can learn from these adaptations and use them to make future decisions, all while taking into account end-to-end goals." This early definition has paved the way for more sophisticated definitions in [6] where the fully distributed nature of today's computer networks is taken into account.

While the above mentioned two definitions provided early aspects essential for cognitive networking, a more comprehensive definition is provided in [6] according to which a cognitive networking system has a distributed set of cognitive processes which collect spatio-temporally tagged network environmental information, including the network parameter behavior from every layer of the network, from every network element within a node, and from other network nodes in order to identify the right network parameters to be used for achieving the individual and end-to-end network goals. The definition in [6] focused on an distributed approach which underlines the spatio-temporal tagging of information and collection, storage, and analysis of information for a larger networking perspective.

1.3 Architectures for Cognitive Networking

We provide a classification of cognitive networking approaches in this section which is motivated by the need to identify the existing solutions within the general research areas in cognitive networking. Figure 1.1 shows such a classification. The first and most simplistic approach for cognitive networking is termed autonomous cognitive networking according to which the cognitive wireless networking devices observe and learn about their environment based on which appropriate actions are taken. While these devices have significant cognition capability, they do not communicate among themselves or with any central entity. There are several situations where autonomous behavior is essential, for example, where there are no centralized network resources. The second type of approach is more useful where there exist centralized resources such as central repositories. These two approaches will be described in subsections 1.3.1 and 1.3.2.

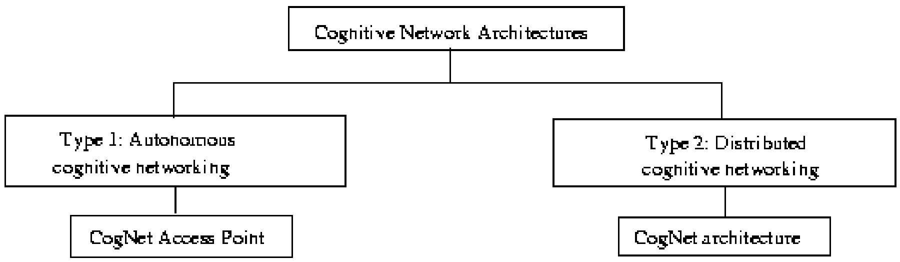

Fig. 1.1. Classification of cognitive network architectures.

1.3.1 Autonomous Cognitive Networking

In the autonomous cognitive networking architecture, nodes adapt by observing and learning from the environment. Similar to the traditional cognitive radio networks, a particular node observes its networking and radio environment in order to obtain cognitive information such as traffic periodicity, traffic pattern, and protocol parameters for every layer. This is illustrated in Figure 1.2 where each node follows the Observe, Orient, Decide, and Act (OODA) loop discussed in [1, 8–10] which acts as the classical state machine for cognitive systems. In the autonomous CogNet, the OODA loop is maintained independently by each of the CogNet nodes. The primary question in this kind of autonomous Cognitive Networking is how successful is the independent use of higher layer information for an autonomous CogNet node. An early prototype, CogNet AP, for studying the impact of the performance of such systems is proposed in [7] and briefly described in the following section.

CogNet AP

As the development of sophisticated software defined radios and cognitive radios are hindered by expensive subsystems, cognitive radio systems may take longer than expected to be popular. On the other hand, the cognition activity applied at higher networking layers can be beneficial and therefore an inexpensive architecture could popularize the concept of cognitive networking.

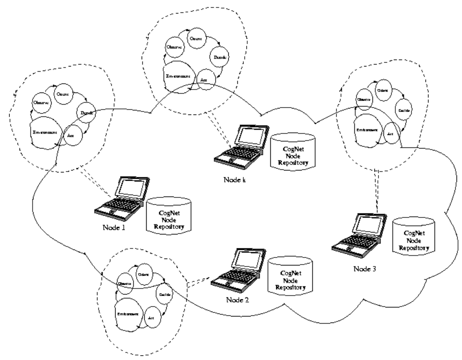

Fig. 1.2. The autonomous cognitive network approach.

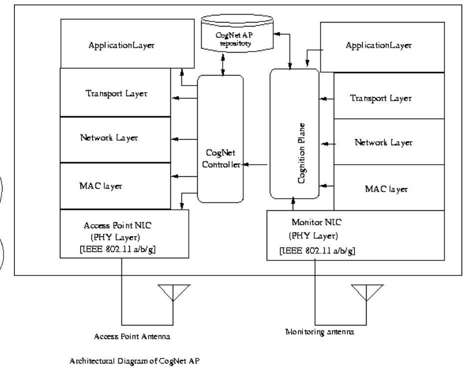

Fig. 1.3. The architecture of CogNet Access Point.

Cognitive Network Access Point (CogNet AP) is one such early prototype in this direction which has many applications and significant research potential. CogNet AP is built from off-the-shelf wireless hardware and open source software in order to demonstrate the benefits of cognitive networking. It gathers, processes, analyzes, and stores information available through its IEEE 802.11 standard based interfaces in order to build a cognitive local repository which holds spatio-temporally tagged network traffic information. The inexpensiveness of the components used for building CogNet AP shows the flexibility of building cognitive network elements compared to cognitive radio devices.

The CogNet AP belongs to the category of autonomous CogNet nodes which have the capability of using higher layer traffic information for efficient management of the network resources. Figure 1.3 shows the architecture of CogNet AP. The CogNet AP has two network interfaces: (i) the service interface and (ii) the monitoring interface. The service interface is used for providing network services to the users or client nodes which are associated to the CogNet AP. The second interface is used for constantly monitoring the channels. Both these interfaces are built from commercial, off-the-shelf, and inexpensive IEEE 802.11 based WLAN Network Interface Cards. Here, the cognition plane constantly monitors the network and the radio environment and populates a local CogNet AP repository. The information is spatio-temporally tagged before being stored in the repository. The CogNet controller makes appropriate decisions for physical, MAC, network, and transport layers. The CogPlane within the CogNet AP receives coarse physical layer information and all the receivable MAC layer frames and higher layer packets through the monitoring interface. From the received packets, any information related to higher layer protocol packets is extracted. Since the CogNet AP is designed for 802.11 spectrum, it is necessary to monitor activities in all the 11 channels.

Monitoring all the channels simultaneously is a resource-expensive activity, therefore, CogNet AP uses a channel sampling approach.

The received frames are used to populate a number of data structures from which higher layer packet information is extracted. The CogNet AP builds statistical models based on its traffic observations. There are two levels of models built: MAC layer models and higher layer models. These models are tagged spatio-temporally in order to exploit the temporal behavior of the network activity in any given geo-spatial point. The temporal cycle is repeated and the traffic model parameters built for every corresponding hour are averaged. For example, in our system, a seven day 24 hour temporal cycle is used in which the week days from Monday to Sunday are represented as Day 1 to Day 7. The MAC layer parameters observed are the following: (i) mean inter-arrival time of MAC frames, (ii) mean inter-arrival time for different frametypes[3], (iii) mean length of frames, (iv) mean length of frames for different frametypes, (v) frame count, and (vi) count of frames for each frametype. In order to generate traffic models for all channels, the channel switching is done in a cyclic manner across the channel range (*i.e.,* from channel 1 to channel 11 for 802.11b systems).

Similar to the MAC layer model, the network, transport, and application traffic models are also built from the packets received at the cognition plane. The main traffic parameters extracted for building traffic models are the following: (i) mean inter-arrival time for IP packets, (ii) mean inter-arrival time for different protocols, (iii) mean length of IP packets received, (iv) mean length of packets for different protocols, (v) packet count, and (vi) count of packets for different protocols. The measured parameters are averaged over multiple samples and temporally stamped for every hour of operation to build the hourly model as mentioned above. Thus CogNet AP generates cyclic hourly models for mapping the network activity in the 802.11 spectrum and builds a Network Activity Time Table (NATT) which can be used by the CogNet Controller to make decisions to improve the performance of devices. An example action is the service channel selection which decides the channel to be used for serving the users.

Channel selection in 802.11 spectrum is important for network access points, especially WLAN APs, though conventional APs do not provide a mechanism for dynamically choosing the best possible channel. In conventional APs, the channel is configured manually by the user. The manual channel setting causes several problems for the residential users as well as the enterprise users of WLAN equipment. In most residential deployments, users do not modify the channels and instead work with the channel preset by the device manufacturers. Thus the use of the preset channel or the dependence on manual channel setting have led to certain channels being heavily used while the remaining ones are mostly unused. For example, since most equipment

[3] Frametype refers to the category of MAC layer frames defined as part of the IEEE 802.11 standard.

manufacturers preset their APs to channel 6 as the default channel, channel 6 is the most heavily used channel in most residential environments. In enterprise WLAN deployment applications, the manual channel selection has made the WLAN deployment a complicated and expensive process. For example, in an enterprise network, optimal channel allocation to the production network requires consideration of a lot of factors such as co-channel or adjacent channel interference. Therefore, CogNet AP with its cognitive abilities to sample channels for network activity and building models for every hour of the day could exploit the periodicity of traffic on every channel and dynamically use the best channel for every hour of the day. In order to choose the channel, the CogNet AP estimates the channel activity and cumulative channel activity of every channel. The activity of a given channel is defined as the mean number of frame transmissions occurred, averaged across all sample durations within a particular hour. Cumulative activity of a channel not only represents the activity of a particular channel, but also considers the activity in other overlapping channels. For example, the cumulative activity is estimated by the following equation:

$$CA_i = \sum_{\substack{k=i-COF \\ k \in CH}}^{i+COF} A_k \tag{1.1}$$

where CA_i represents the cumulative activity in channel i, COF is the channel overlap factor which in 802.11b is three, CH is the set of channels in the system (set of channels 1 to 11 in 802.11b), A_k is the activity in a given channel k. Therefore, the cumulative activity is the sum of the activities in the overlapping neighboring channels. For the physical layer, the CogNet controller chooses the best possible channel based on the following equation.

$$\text{OperatingChannel} = \arg\min_i CA_i \tag{1.2}$$

Measurements taken from a CogNet AP based testbed in a residential apartment for about three weeks are provided here as an example. The objective of this measurement setup was to see the traffic pattern differences present in a real network environment. During the measurement period, the traffic parameters observed are averaged across the same time points. Figures 1.4 and 1.5 show the traffic pattern observed. The collected results were divided into days of a week. For example, the observed traffic averaged across three Sundays is shown in Figure 1.4. We noticed crowded activity in that particular locality on two different orthogonal channels (Channels 6 and 11) and very low activity on Channel 1. The day time traffic on Sundays showed to be almost normal whereas we noticed traffic surge in the evenings. This trend was similar across all the three orthogonal channels. We noticed no significant activity on the non-orthogonal channels (802.11b channels which are not in the set of 1, 6, and 11).

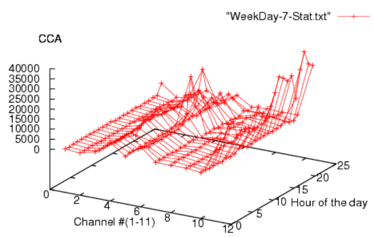

Fig. 1.4. Observed traffic pattern for Sundays.

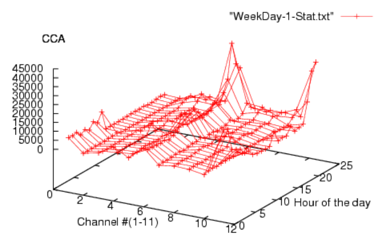

Fig. 1.5. Observed traffic pattern for Mondays.

On Mondays, an entirely different pattern which was determined by the residents' work and living pattern is noticed. For example, channel 11, which is the most heavily loaded channel on Sundays, experienced much less activity. In addition, during the hours from 9am till 8pm, the traffic seemed to be very low. After 9pm, once again the traffic on channel 11 started growing. This trend remained similar for other orthogonal channels as well.

The throughput achieved by the end-user devices when communicating with the CogNet AP is also provided here. In this experiment, results from ten runs are provided and the throughput obtained from all these runs were averaged. The channel was set to one of the preset channels before transferring the files. The dynamic channel selection algorithm was turned off and files of size approximately 4MB were transferred from a client mobile computer to the CogNet AP. This measurement provided the throughput achieved when not using CogNet AP's channel selection algorithm based on the traffic observations. During the second experiment with CogNet AP, the dynamic channel switching algorithm was turned on and the CogNet AP chose the best channel based on its traffic observations. The location of CogNet AP as well as the mobile computer was not changed during both the above experiments. Though the data rate is set to auto-rate, the AP and the mobile computer were kept static at their locations in order to avoid any potential data rate changes during our experiments. A throughput improvement is noticed as shown in Figure 1.6. The average throughput obtained from the traditional AP is about 3.25 Mbps in the residential testbed (Figure 1.6) and a throughput improvement of 10-15% is observed.

1.3.2 Distributed Cognitive Networking

Distributed cognitive networking is another possible approach that is expected to better exploit cognitive capabilities. These cognitive devices can interact among themselves, with centralized data bases, and across a variety of heterogeneous systems in order to utilize the cognitive network information efficiently. For example, in the future it is very likely that networking entities

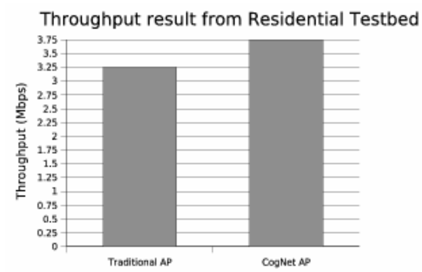

Fig. 1.6. Throughput result.

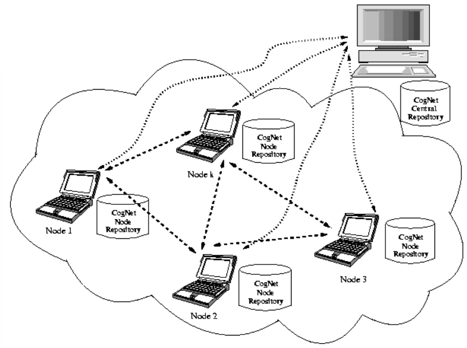

Fig. 1.7. The distributed cognitive networking paradigm.

of different kinds will co-exist and, therefore, any capability to work across heterogeneous cognitive networking forms may be a great advantage. Figure 1.7 shows the architecture of distributed cognitive networks where the cognitive networking system can work in a fully distributed fashion. The interaction with centralized or distributed data base storage which contains spatio-temporally tagged network information is a significant value addition of this approach. The following section presents an example of distributed cognitive networking architecture.

1.4 CogNet: Cognitive Complete Knowledge Network

In this section, we present one of the recently proposed architectures [6] in cognitive networking called Cognitive Complete Knowledge Network (CogNet). In the CogNet architecture, a Cognitive Bus (CogBus) within a Cognitive Plane (CogPlane) is used for exchanging and passing the data and information necessary for the efficient functioning of the CogNet framework. This architecture provides a unique solution that fills the gap in learning the spatio-temporal aspects of the protocols' performance at every layer. For example, the temporal and spatial periodicity of higher layer traffic is not utilized in any of the existing protocol solutions for medium access protocols [3, 5]. Thus, the primary advantage of CogNet is the network performance benefit that can be derived from the system wide cognitive capability. A Cognitive Executive Function (CEF) within the CogPlane provides an analytical approach for translating the observations to actable information within the spatio-temporal context. In addition, the CEF coordinates the use of CogBus, CogPlane, and the inter-nodal exchange of information.

In comparison to the architecture proposed by Thomas *et al,* [10], which proposes a monolithic cognition layer, the CogNet architecture is fully distributed. For example, in CogNet a cognition module (also called a cognitive

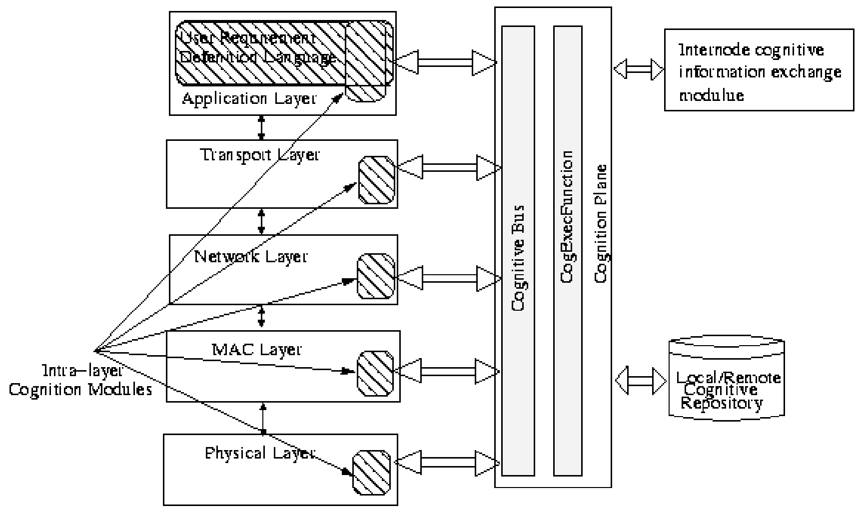

Fig. 1.8. The distributed CogNet Framework.

agent) is present at every layer and can gather information and control parameters related to that particular layer. In a way, the cognitive agent serves as a local sensor, controller, and actor of each particular layer. Two important aspects of this architecture are the following: (i) this architecture can still maintain the layered abstraction of the networking protocol stack which is one of the primary factors behind the successful evolution of today's computer networks, and (ii) this architecture can simplify the complexity of cognitive processes which otherwise may become unmanageable. In addition, the semantic interpretation of network events, behavior of protocol parameters, and the actions taken at every layer can be more efficiently handled if each layer has a cognitive module of its own. The cognitive plane helps coordination of the cognitive modules and of the information and data exchange through an internal cognitive bus. Furthermore, the CogPlane also helps the communication with other CogNet enabled nodes. This is particularly important in our architecture because, in many situations, cognitive information can be better accumulated if there is a framework for communication between CogNet nodes.

The CogNet architecture uses a language for defining the end-user requirements or end-to-end goals. The difference between our approach and existing approaches lies in translating the end-user requirements and network observations into what needs to be executed at every layer. In our case, the CogPlane is responsible for translating the end-goals to the responsibilities or action items required for each layer. A joint layer optimization module within the CogPlane develops the interaction models across the layers. This module is very important, as an optimization that does not consider the potential negative impact can be counterproductive as discussed in detail in [4].

1.4.1 CogPlane and CogBus

The primary elements that enable the cognitive information exchange in the CogNet architecture are the Cognitive Plane (CogPlane) and the CogNet Bus (CogBus). The design of CogPlane is very important in developing cognitive network architectures similar to CogNet. The cognitive modules at each layer of the networking protocol stack report their observations which will be collected by the CogPlane and stored in the local or remote repository. Upon the user applications' request, the CEF within CogPlane executes optimization algorithms for joint optimization and scheduling of resources. These optimization algorithms will generate the proper parameters to be chosen at each of the network layers and the cognitive modules are responsible for reconfiguring the protocols at each layer. Thus the CogPlane provides an opportunity for dynamic resource allocation and management with the help of the past history of the user, the device, and the network. The joint resource optimization within the CogPlane is aimed at managing the resources and the scheduling framework across multiple layers in such a way that it can achieve a satisfactory user experience. Essentially, the CogPlane intends to do a fast service compositioning across the networking layers. In addition to the coordination among the cognitive modules within a given node, the CogPlane also helps coordination between cognitive modules across nodes. The CogPlane uses the inter-node cognitive information exchange module which runs protocols such as the Cognitive Information Exchange Protocol (CIEP) [6] to manage the inter-node communication across homogeneous or heterogeneous CogNet entities.

In order to enable communication between the modules and the CogPlane and in order to achieve the cross layer cognition information exchange, a bus architecture providing a broadcast medium is used. This cross layer CogNet bus (CogBus) will provide an infrastructure for publishing or exchanging cognitive information across various layers. The CogBus is also used to override the intermediate layers between a set of source-destination layer pairs. For example, if necessary, the physical layer or MAC layer can now directly communicate with the application layer without passing through the intermediate layers. Such short-cut communication scenarios may be of significant benefit. The first challenge in designing a CogBus architecture is the requirement for a light weight design of the bus architecture. Since some of the layers in the protocol stack are implemented in the Operating System (OS) kernel, it is necessary to provide a simple and efficient design for CogBus. The second challenge in designing a CogBus is in the design of an information format for exchanging CogNet specific information across all layers, that must be simple as well as extensible. One example for such information format at the application layer is the eXtensible Markup Language (XML) which might need design level changes to work in CogPlane. For example, XML is a text-based protocol which may be of high overhead, and therefore a binary-based light weight design for information exchange format may be preferable.

CogNet uses the CIEP protocol [6] for exchanging information across the devices in the same networking eco-system or between devices and repositories. Implementation of CIEP depends on the nature of the network. For example, in CogNet, the CIEP exploits the benefits of the broadcast nature of the wireless channel in order to query the neighbor nodes. CogNet implementation of CIEP targeted the neighbor repositories though it is possible to utilize remote or centralized repositories as well. Using CIEP, a given CogNet node broadcasts a request to its neighbors asking for their past experiences including the network parameters and network performance they observed for a particular destination node or destination network for a specific temporal domain. The requesting CogNet node can broadcast a *CIEP-Request* packet containing the necessary information about source and destination and the requested parameters. Upon receiving a *CIEP-Request* packet, the neighbor CogNet node that maintains a local CogNet repository queries its storage and replies with the *CIEP-Reply* packet. The *CIEP-Reply* packet contains the information needed by the requesting CogNet node.

1.4.2 Case Study: CogTCP

In this section, we present an example realization of CogNet with a simple transport layer solution which we call CogTCP. CogTCP is a TCP transport layer with a Cognitive Transport Module (CTM). The CTM has intra-layer, inter-layer, and inter-node cognitive capability. Intra-layer cognitive capability refers to the ability of the module to learn from various internal transport layer (e.g., TCP) functional modules (e.g., socket structures and transmission control blocks). In a situation where a busy Internet server accepts thousands of TCP connections every second from a large number of networks, every new TCP connection, today, has to undergo the same transport layer behavior, *e.g.,* the slow start phase, congestion management phase, and transmission window behavior. In CogNet, mapping of the TCP behavior models, current and past, to the host addresses and network addresses in a spatio-temporal manner will help the wireless clients optimize the protocol parameters, thereby improving their performance. Example TCP parameters are: average congestion window size, slow start threshold, probability distribution of run time throughput, bandwidth delay product, smoothed round trip delay and its average value, and spatio-temporal distribution of throughput. The inter-layer learning capability refers to the ability of the CTM to interact with other layers through the cross layer cognition bus. The inter-node cognitive capability helps a node to obtain cognitive information from other nodes. In CogNet, the CTM helps inter-node (*i.e.,* across the transport layers of different devices) information exchange across existing TCP connections through centralized nodes such as base stations/access points. Therefore, a new TCP connection in a wireless client can query the existing connections, clients, and/or the past history information in order to avoid slow start and unnecessary congestion-control related throughput degradation. Thus, the intra-layer, inter-layer, and inter-node cognitive information exchange in CTM is particularly useful in wireless systems where the network parameter behaviors are obtained from a repository. In such cases, the inter-layer interaction may also be useful in querying the external counterpart, proxy-servers, base stations, and/or end-hosts, in order to obtain the necessary information for improving the host's connection performance.

Simulation Results

In this section, we present the basic simulation environment and the experimental setup used for a preliminary performance evaluation of the CogTCP solution. The simulation engine was built around the GlomoSim simulation tool and the routing protocol was Distance Vector along with the IEEE 802.11 DCF MAC layer protocol. The physical layer data rate used was 2 Mbps and the radio propagation model used was two-ray model. The network topology (Figure 1.9) chosen was a grid topology with grid dimension set to 300 meters and a transmission power of about 15.0 dBm which gives an approximate transmission range of 375 meters when simulated with two-ray propagation model. We used a 25 node network with an FTP server running on Node 24 and Nodes A and B running FTP clients. Initially, when Node A runs an FTP session, the CTM module within the transport layer keeps track of the parameters and updates them in the central repository which can be centralized or distributed, and is not shown explicitly in Figure 1.9. When Node B wants to open a TCP connection with the FTP server, it queries the central repository about the right protocol parameters observed by previous nodes (in this case Node A) within the spatio-temporal domain. Hence, Node B receives the stored values for the average congestion window and the slow start threshold from Node A, and uses them as its initial values of the congestion window and of the slow start threshold and begins its data transfer session with the FTP server. In order to estimate the advantages of CogTCP, we studied the behavior of congestion window and throughput achieved which are presented below.

Figures 1.10 and 1.11 show the congestion window evolution in time for both Node A and Node B. Figure 1.10 presents the congestion window variation for short file transfer sessions. At time 20 s, Node A begins an FTP session with the FTP server for transferring a short file of length 10KB (chosen to illustrate a simple example). At time 20.8 s, Node A's connection completes and it registers the connection parameters such as the current congestion window, average congestion window, and slow start threshold. At time 21 s, Node B initiates a new FTP connection with the FTP server and obtains the protocol parameters that are stored in the repository, *i.e.,* the average congestion window (4196 Bytes) and the last value of the slow start thresh-

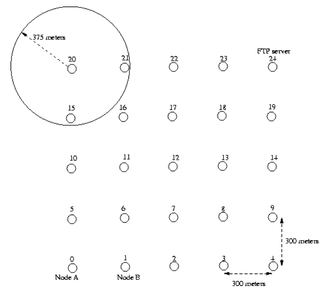

Fig. 1.9. The network topology used for simulation experiments.

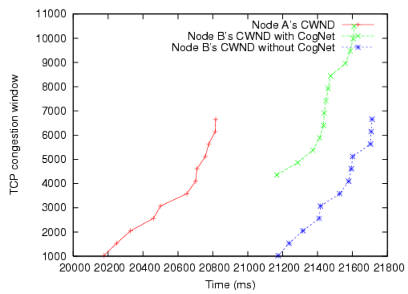

Fig. 1.10. Congestion window Vs time for 10KB file transfer.

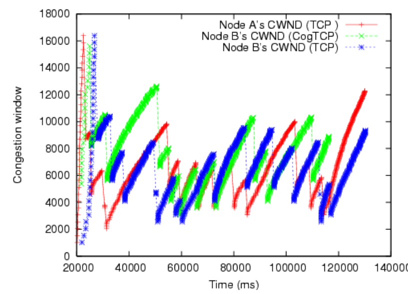

Fig. 1.11. Congestion window Vs time for 1MB file transfer.

old (16 KB) experienced by Node A. In this case, Node B uses the average congestion window from the repository as the initial congestion window, *i.e.,* its initial congestion window is set to 4196 Bytes. The time evolution of the congestion window for the new connection originated by Node B, with TCP and CogTCP is shown in Figure 1.10. We noted that the average congestion window at the end of Node B's connection improved from approximately 4000 Bytes to about 5376 Bytes when Node B used CogTCP. Therefore, Node B's transport layer connection could benefit from the experience gained by Node A and made available through the CogNet repository. Figure 1.11 shows the congestion window as a function of time for long file transfer sessions where we noted improvement in the average congestion window and the transfer time. When Node B used CogTCP, the file transfer ended approximately at 115 seconds in comparison to normal TCP which took approximately about 130 seconds. The file transfer time for Node B's session has been reduced by approximately 15 seconds when Node B used CogTCP.

We ran a 100 seed simulation campaign for estimating throughput improvement in CogTCP due to the exploitation of the information available from repositories. We used short and long file transfer sessions for these experiments. For the short file transfer experiments, we set up FTP sessions with file size fixed at 10 KB. In these experiments we studied two versions

Table 1.1. Throughput performance with and without background traffic.

Throughput without background traffic (kbps)			
	TCP	CogTCP-1	CogTCP-2
Short file transfer sessions (10 KB)	127	139	139
Long file transfer sessions (1 MB)	62	66	68

Throughput with background traffic (kbps)			
	TCP	CogTCP-1	CogTCP-2
Short file transfer sessions (10 KB)	120	135	135
Long file transfer sessions (1 MB)	55	57	58
Long file transfer sessions (10 MB)	44	50	50

of CogTCP, *i.e.,* CogTCP-1 and CogTCP-2. CogTCP-1 uses the value of the average congestion window experienced by Node A as the initial congestion window whereas CogTCP-2 uses half that value as the initial congestion window. In both cases, Node B uses the same slow start threshold. Initially we attempted data transfer sessions without background traffic, some results are presented in Table 1.1. The throughput performance for short files shows throughput improvement for CogTCP-1 compared to traditional TCP which does not exploit cognitive information. CogTCP-2 does not show significantly different performance compared to CogTCP-1 as the initial congestion window chosen was smaller than that of CogTCP-1, and the connection was terminated sooner than expected due to the short file size used. Table 1.1 also shows the throughput performance over traditional TCP when CogTCP was used for transferring longer files of length 1 MB. Here again, we noticed similar throughput improvement for CogTCP when compared to traditional TCP. In this case, CogTCP-2 does show slightly better throughput than CogTCP-1. When CogTCP-2 takes half the value of the average congestion window obtained from the repository as its initial congestion window, it takes a longer time to touch the ceiling of the congestion window. The congestion window ceiling occurs when either the congestion window meets the receiver advertised window or there is a congestion-related packet loss. Therefore, for long connections, using half the neighbor's average congestion window works slightly better than using the average congestion window. This also shows that the choice of parameters may be sensitive not only to time, space, source destination pair, but also to the session parameters for a particular data transfer session. A more detailed study to reveal the right choice of parameters is left for future research.

The throughput performance in the presence of background traffic generated by Constant Bit Rate (CBR) sources is also shown in Table 1.1. The background CBR connections were originated from random source nodes and all terminated at Node 24. Also, in this experiment, we used short file transfer sessions (10 KB) and long file transfer sessions of 1 MB and 10 MB for the FTP sessions from Nodes A and B to the FTP server. When we used 10 CBR sessions for creating the background traffic, we noticed that the average of throughput achieved by TCP, CogTCP-1, and CogTCP-2 is reduced slightly when compared to the throughput achieved in the absence of background traffic. However, the relative throughput gain for CogTCP-1 and CogTCP-2 compared to traditional TCP remained almost the same as that of the experiments without background traffic.

1.5 Summary

Cognitive networking has significant potential to contribute to the development of next generation wireless networks, especially when one considers the system level optimization that these cognitive networks can provide. While the early concepts provided by Clarke *et al.* and Thomas *et al.* are good initial thoughts, they do not provide a solution which can be integrated with existing layer-oriented network architectures. CogNet is an architecture that does provide the unique property of co-existence with today's network architecture. CogNet also provides the important benefits that can be derived from spatio-temporal information. In this chapter, we presented an architectural classification for Cognitive Wireless Networks, the design and results from an autonomous cognitive access point, and a novel distributed cognitive network architecture along with some performance results for a specific case study. Although much work remains to be done, early results provide an encouraging view on the future of cognitive wireless networking.

List of Abbreviations and Symbols

CogBus CogNet Bus
CogPlane Cognitive Plane
CogNet Cognitive Complete Knowledge Network
CIEP Cognitive Information Exchange Protocol
WLAN Wireless Local Area Network

References

1. J. Boyd. A Discourse on Winning and Losing: Patterns of Conflict. In *Lecture Notes, US Department of Defense, Pentagon*, 1986.
2. D. D. Clark, C. Partridge, J. C. Ramming, and J. T. Wroclawski. A Knowledge Plane for the Internet. In *Proceedings of ACM SIGCOMM 2003*, August 2003.
3. C. Doerr, M. Nuefeld, J. Fifield, T. Weingart, D. C. Sicker, and D. Grunwald. MultiMAC – An Adaptive MAC Framework for Dynamic Radio Networking. In *Proceedings of IEEE DySPAN 2005*, pages 548–555, November 2005.
4. V. Kawadia and P. R. Kumar. A Cautionary Perspective of Cross-Layer Design. In *IEEE Wireless Communications Magazine*, pages 3–11, February 2005.
5. P. Kyasanur and N. H. Vaidya. Protocol Design for Multi-hop Dynamic Spectrum Access Networks. In *Proceedings of IEEE DySPAN 2005*, pages 645–648, November 2005.
6. B. S. Manoj, Michele Zorzi, and Ramesh R. Rao. CogNet: A Cognitive Complete Knowledge Network System. In *Technical Report, Department of Electrical and Computer Engineering*, University of California San Diego, CA, USA, January 2006.
7. B. S. Manoj, Michele Zorzi, and Ramesh R. Rao. CogNet AP: An Access Point for Cognitive Networks. In *Technical Report, Department of Electrical and Computer Engineering*, University of California San Diego, CA, USA, September 2006.
8. J. Mitola. Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio. In *Ph.D Thesis, Royal Institute of Technology (KTH)*, Sweden, 2000.
9. J. Mitola and G. Q. Maguire. Cognitive Radio: Making Software Radios More Personal. In *IEEE Personal Communications Magazine*, pages 13–18, Snowbird, UT, August 1999.
10. R. W. Thomas, L. A. DaSilva, and A. B. MacKenzie. Cognitive Networks. In *Proceedings of IEEE DySPAN 2005*, pages 352–360, November 2005.

Index

CIEP, 10
CogBus, 8
CogNet, 3
CogNet AP, 1
Cognitive Bus, 8
Cognitive Information Exchange Protocol, 10
Cognitive Plane, 8
Cognitive Transport Module, 11
CogPlane, 8
CogTCP, 11