

# A Reference Model for Future Computer Networks

Hoda Hassan

Computer Science and Engineering Department,  
American University in Cairo  
Cairo, Egypt.  
mhelali@aucegypt.edu

**Abstract**—Future Internet design demands revolutionary approaches unfettered by legacy constraints and concepts. This paper presents a clean-slate Concern-Oriented Reference Model (CORM) for architecting future computer networks based on novel network design principles. CORM realizes the network as a software-dependent complex system. It defines the network design space in terms of function, structure and behavior, and perceives each of these design space elements within the context of network concerns, identified as Application, Communication, Resource and Federation. CORM adopts a bottom-up approach in network construction, focusing on the network building block, whose structure and behavior are inspired by evolutionary bacterium cell. Hence, CORM refutes the long endorsed concept of layering, and intrinsically accounts for emergent behavior, while ensuring network congruency. CORM's basic abstraction unit is validated using the Function-Behavior-Structure engineering framework. The paper concludes by presenting and evaluating an architecture derived from CORM.

**Keywords**- Complex systems; Computer Network design; Computer Network Reference Model.

## I. INTRODUCTION

Designing future computer networks dictates an eclectic vision capable of encompassing ideas and concepts developed in contemporary research unfettered by today's operational and technological constraints. However, unguided by a clear articulation of core design principles, the process of network design may be at stake of falling into similar pitfalls and limitations attributed to current network realizations. We opine that deficiencies apparent in current network realizations can be traced to the following underlying causes [1];

- The general trend towards network science and engineering lacks a systematic formalization of core principles that expresses essential network features required to guide the process of network design and protocol engineering;
- The prevalence of a top-down design approach for computer network architecture demonstrated as confining intelligence to network edges, and maintaining a dump core; and
- The absence of a general reference model, which embodies core network design principles, and acknowledges the multidimensionality in design entailed in architecting computer networks that reach beyond core networking requirements.

In this paper, we present a clean-slate Concern-Oriented Reference Model (CORM) for architecting future computer networks. CORM has sprouted as a generalization to our concepts, design principles and methodology presented in [2, 3].

Our initial endeavor, CellNet [3], was a bio-inspired network architecture, which was tailored to operate in accordance to the TCP/IP suite. However, CORM is a reference model for architecting any computer network. It expresses the most fundamental design principles for engineering computer networks at the highest level of abstraction. CORM stands as a guiding framework from which several network architectures can be derived according to specific functional, contextual, and operational requirements or constraints. CORM conceives computer networks as a distributed software-dependent complex system that needs to be designed along two main dimensions: a vertical dimension addressing structure and configuration of network building blocks; and a horizontal dimension addressing communication and interactions among the previously formulated building blocks. For each network dimension, CORM factors the design space into function, structure and behavior, applying to each the principle of separation of concerns (SoC) for further systematic decomposition. Perceiving the network as a complex system, CORM constructs the network recursively in a bottom-up approach (In this research work the term bottom-up refer to network composability as opposed to its more frequent use to refer to layer organization in the Internet layered architecture). CORM defines the network cell (NC) as the network building block. The NC's structure and behavior mimic the structure and behavior of evolutionary bacterium cell. The network is then synthesized from NCs according to a structural template that defines different structural boundaries.

Being a reference model for computer networks, CORM can be considered a definitional model; it expresses the required characteristics of a system at an appropriate level of abstraction [4]. CORM expresses the characteristics of adaptable complex systems, and network functionalities within its basic abstraction unit (CORM-NC), and enforces both to be synthesized into the network fabric by construction. Therefore, we validated CORM by validating the derivation of CORM-NC. In this respect, we used the Function-Behavior-Structure (FBS) framework [5] as our validation model. The FBS is applicable to any engineering discipline for reasoning about, and explaining the nature and process of design [13]. Furthermore, we present CAHN as an architecture for ad hoc networks derived from CORM, and evaluate CAHN's performance through simulation.

The paper is organized as follows; Section 2 overviews related work. Section 3 introduces CORM, and validates CORM's basic abstraction unit using the FBS framework. In Section 4, we derive CAHN, an architecture for ad hoc networks based on CORM, and evaluate CAHN's performance through simulation. Section 5 concludes the paper.

## II. RELATED WORK

The Internet has been criticized for lack of security, difficulty in management, incognizant protocol operation, and

inadequate support for mobility [6], thus motivating a plethora of proposals; some attempting point solutions to specific problems, while others aimed architectural innovations. We identify two network dimensions along which most architectural proposals can be classified; a vertical dimension addressing structure and configuration of protocols, and a horizontal dimension addressing communication. We claim that most of the proposals focus on one dimension while diminishing the other. Below, we present proposals in [7], [8], and [9] as examples supporting our previous claim.

The SILO Project in [7] proposes an architecture based on fine-grained service elements that can be composed based on ontology of functions and interfaces. A SILO-enabled application can thus specify high-level functional requirements, and request service elements to be composed accordingly to meet these requirements. The Recursive Network Architecture (RNA) presented in [8], is based on recursive composition of a single configurable protocol structure. RNA avoids recapitulation of implementation, as well as encourages a cleaner cross-layer interaction. This is achieved by using, a single meta-protocol module, which facilitates the inter-protocol interactions at different layers. Content Centric Networking (CCN) presented in [9], creates a network architecture based on named data instead of named hosts by making the address in packets correspond to information or elements reachable on the Internet, rather than machines. CCN proposes a layered node model that resembles the structure of TCP/IP layering model, but differs in layers' responsibilities.

SILO and RNA have been presented as clean-slate architectural attempts towards Future Internet. However, layering, as a design paradigm, is still the prevailing model. An essential goal of both proposals is to gracefully embrace cross layering into the present network stack. Although considered clean-slate architectures, we argue that by adhering to layered stacks as the underlying model, both proposals might suffer from shortcomings attributed to the Internet model. First, both architectures do not give guidance to engineers as how to handle cross interests among composed protocols: The single control agent in SILO, as presented, is a monolithic unit representing a single point of failure for all protocols working under its control, as well as imposing scalability problem as service diversity, granularity, and operational parameters increase. As for RNA, we note that confining the logic for horizontal and vertical interlayer communication into a single entity, is a very challenging task that is error prone. Furthermore, it lacks explicit representation for interactions leaving it to be decided on at runtime. This allows for implicit assumptions to creep into protocol design and implementations. Second, both architectures have undermined monitoring and resource management failing to express both functions as first class architectural constructs. Finally, as presented, both architectures focus on the vertical dimension of the network without suggesting how the horizontal dimension will be incorporated in terms of naming and addressing. On the other hand, CCN focus mainly on naming and addressing and disregard the need for managing on-node interactions. Similar to SILO and RNA, CCN also adheres to a layered stack and fails to provide explicit specifications for inter-protocol interactions and cross-interest management.

### III. CORM: A CONCERN-ORIENTED REFERENCE MODEL FOR COMPUTER NETWORKS

For completeness, this section gives a synopsis of CORM's design principles and methodology presented in [2, 3]

#### A. CORM Design Principles and Methodology

CORM derivation process was initiated by identifying two core network-design principles that, we assert, are applicable to all computer networks regardless of their size, purpose, operational context, or capabilities. The first principle states that a computer network is a complex system, while the second principle states that a computer network is a distributed software system. From a complex system perspective, computer networks need be composed of *autonomous entities* capable of *emergent behavior* that can act coherently to perform the global system function, in spite of *intricate interactions* occurring at the micro and macro level [10]. On the other hand, as a distributed software system, computer networks need to be designed according to Software Engineering (SE) principles and concepts. Separation of Concerns (SoC) is a prominent SE principle that was extensively applied to the design of CORM for systematic decomposition of the network system. Guided by our principles, we formulated a Concern-Oriented Bottom-Up design methodology for deriving CORM. The Bottom-Up approach is motivated by our first design principle in general, and network composability of autonomous entities in specific, thus accentuating the importance of the entities composing the network system. These network-building entities need to imitate entities in a Complex Adaptive System (CAS is a complex system whose emergent behavior always lead to overall system stability, in contrast to unstable complex systems whose emergent behavior may result in system meltdown. In this paper the term complex system indicates CAS unless otherwise stated), by possessing adaptability, self-organization and evolvability as intrinsic features. The network will then be recursively synthesized from these network-building entities in a bottom-up approach substantiating the two main network-dimensions; a vertical dimension that addresses structure and configuration of network building entities, and a horizontal dimension that addresses communication and interactions among the previously formulated building entities. For the synthesized networks, the Concern-Oriented paradigm represents our vision in network functional decomposition realized at the micro (network-building entities), as well as at the macro (network horizontal and vertical dimensions) level.

As a direct consequence of our Concern-Oriented Bottom-Up design methodology, CORM does not differentiate between network core and network edge in terms of capabilities, thus contradicting the End-to-End (E2E) principle that has been central to the Internet design. It has been argued that the E2E principle has served the Internet well by keeping the core general enough to support a wide range of applications. However, we contend that, taken as an absolute rule, the E2E principle constrained core evolvability rather than fostered its capabilities rendering the Internet biased to those applications that can tolerate its oblivious nature, and forcing designers and protocol engineers to adopt point solutions to compensate for core deficiencies. Another consequence to our proposed bottom-up network composition is contradicting the prevailing misconception of abstracting a network in terms of an inter-network. Adopting a bottom-up approach to network

composition implies recursive construction of the inter-networks from networks, which are likewise recursively constructed from network components, which are constructed from one or more network building blocks.

B. CORM Components

A network reference model is an abstract representation of a network. It conveys a minimal set of unifying concepts, axioms, and relationships to be realized within a network [11]. For expressing a multi-dimensional system, such as a computer network, multiple abstract representations are required to capture the system from different perspectives. CORM abstracts a computer network in terms of function, structure and behavior, which are represented respectively as, the network-concerns conceptual framework (ACRF), the network structural template (NST), and the information flow model (IFM). Both the ACRF and the NST have been previously defined in relation to CellNet in [3]. However, in the following subsections, we will revisit their definition at the level of a reference model. The ACRF will be redefined in terms of the network requirement specification while the NST will be abstracted in terms of the basic network building block (NC).

1) ACRF: Conceptual Framework for Network Concerns

We postulate that the requirement specification of a computer network can be expressed as follows: “The network is a communication vehicle that allows its users to communicate using the available communication media”. Accordingly, we identify the network users, the communication media (physical and logical), and the communication logic as primary requirements, which the network design need to address and plan for. Applying the concept of SoC to the above requirement specification statement, we identify four main network concerns; Application Concern (ACn), Communication Concern (CCn), Resource Concern (RCn), and Federation Concern (FCn). The first three are core network concerns encompassing the network functional requirements, while the fourth is a crosscutting concern (non-functional requirement) representing the area of intersection or common interests among core concerns. Elaborating on each concern we have:

- The ACn encompasses the network usage semantics; the logic and motivation for building the network, where different network-based end-applications (network users) can be manifested.
- The CCn addresses the need for network route binding to provide an end-to-end communication path allowing network elements to get connected (communication logic)
- The RCn focuses on network resources, whether physical or logical, highlighting the need for resource management to efficiently address different trade-offs for creating and maintaining network resources (communication media).
- Finally, FCn orchestrates interactions, resolving conflicts and managing cross interests, where areas of overlap exist among the aforementioned core concerns.

These four network concerns are manifested as CORM conceptual framework for network concerns, referred to hereafter as ACRF. The ACRF represents the blueprint for the network functional design that need to be realized along both network dimensions; vertically on the network component and horizontally among network components.

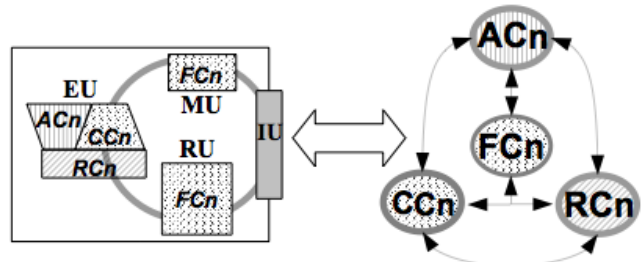


Figure 1. ACRF realization within an NC

Analyzing the Internet model (vertical dimension) and the current network realizations (horizontal dimension) with respect to the ACRF framework, we note that both RCn and FCn are absent. Vertically, the Internet-layered model accounts for ACn and CCn. However, the model did not apply the correct concern separation; a single concern was split along two layers. Moreover, the strict layered paradigm for functional decomposition curtailed all possibilities for considering FCn. As for RCn, it was assumed that resource-management functionalities, are either applications of specific type, and thus will be overlaid on top of the protocol stack, or are to be handled locally by the physical media. For the horizontal dimension, current network realizations account for both ACn and CCn, while the RCn and FCn are usually realized as point solutions. Servers and server farms represent ACn, while routers, switches, and DNS represent CCn. Both RCn and FCn, are implemented as add on functionalities conducted by the use of special protocols for network management and traffic engineering.

2) NST: Network Structural Template

The NST defines the structure of the network building-blocks, and the logic by which these blocks are grouped to compose the network. We classify network building-blocks into computational/decision capable entities, and a communication substrate. The former encompass the network-concern space (ACRF framework), while the latter is a passive interaction media for information exchange. Being the primary constituents of a software-based complex system, network-entities need to possess adaptability, self-organization and evolvability as intrinsic features thus mimicking bacterial cells in a bacterial colony; our adapted model of complex systems [12]. Hence, we define the Network Cell (NC) to be the primary network building block. The NC is a self-contained computational/decision entity capable of monitoring its state, adapting to perceived conditions, inferring decisions, recording its experience, and eventually evolving through self-learning and intelligent adaptations. One or more NCs make up a Network Component (Ncomp), which we define as the basic network entity capable of end-to-end communication. The ACRF is realized within the NC as illustrated in Fig. 1, thus forming the basic abstraction unit of CORM; the CORM-NC. For further details on the internal structure and units of the NC we refer the reader to [1, 2, 3].

Network Compositional Logic (NCL) defines the bottom-up network construction out of network-entities, and identifies the different interaction boundaries that can occur among network-entities (NC and/or Ncomp). NCL stems from our bottom-up definition of network and inter-network construction. NCL defines a computer network as two or more Ncomp connected by a communication substratum, where Ncomp interactions are

sustained, despite the heterogeneity of the hardware, middleware, and software of the connected Ncomps. As for a computer inter-network, NCL defines it as two or more computer networks connected by communication substrate, where interactions among Ncomps residing within each of the connected networks are sustained, despite the heterogeneity of the hardware, middleware, and software employed by the Ncomps composing the connected networks. Integrating NC, Ncomp, and NCL, we derive CORM NST, and define it using EBNF as follows:

CORM NST EBNF formal Definition:

**Notations**

- Trailing \* →repeat 0 or more times
- Trailing + →repeat 1 or more times

**Abbreviations**

- MU = Monitoring Unit
- CCS = Cell Communication Substratum
- RU = Regulation Unit
- Ncomp = Network Component
- EU = Execution Unit
- Net = Network
- IU = Interface Unit
- NCS = Network Communication Substratum
- NC = Network Cell
- INet = Inter-network

**Grammar Definitions**

- NC = MU RU EU IU CCS
- Ncomp = NC (CCS NC)\*
- Net = Ncomp (NCS Ncomp)+
- INet = Net ( NCS Net)+ = Ncomp (NCS Net NCS)+ Ncomp

3) IFM: The Information Flow Model

The Information Flow model (IFM) represents the horizontal dimension of the network. IFM depicts the interactions occurring among network entities, giving rise to the emergent behavior required for network adaptation and evolution. The IFM captures the aspects of information exchange by defining two sub-models: Data Representation sub-model (DR) and Data Communication sub-model (DC). Data representation and communication in CORM exist at both the vertical and the horizontal network dimensions. Vertically, data representation and communication occurs within an NC, as well as between the different NCs making up a Ncomp. Horizontally, data representation and communication occurs between Ncomps in the same network, or across networks. DR will provide categorization for the different types of information flowing in the system, according to the ACRF framework. As such, DR is mainly concerned with the “meaning” of information flowing within the network system. DR need to handle complexity in terms of the amount of information required to depict the system-states at the macro and micro level, taking decisions on the details that need to be exposed and those that need to be suppressed. DC, on the other hand, is concerned with communication aspects including interface compatibilities, data forming across different communication boundaries and majorly routing functions, including addressing, naming and forwarding. Similar to DR, the DC will need to address characteristics of complex systems, such as the free-scale small-world layout, when devising the routing functions. Detailing DC and DR is the focus of our future work.

C. CORM Features

CORM refutes the long endorsed concept of layering, introducing the CORM-NC as a novel abstraction unit. To our knowledge, CORM is the first reference model that addresses the need for engineering for emergent behavior by accentuating monitoring, knowledge acquisition, and regulation as first class intrinsic features of the basic abstraction unit (BAU) –the

CORM-NC. Furthermore, we argue that CORM maintains system integrity due to network construction congruency, where Ncomps, networks and inter-networks are defined recursively in terms of the BAU. In addition to the previously mentioned features, CORM facets acknowledge the multidimensionality of the networks, and accounts for concepts and notions proposed by contemporary designs and architectures including protocol composability out of fine-grained micro-protocols, dynamic protocol adaptation, protocol extensibility and flexibility, cross interest management and control, context awareness through monitoring, resource management as a standalone requirement, and inspired biological behavior and evolution. Table 1 highlights the differences between CORM and the more conventional layered network models (e.g., Internet, OSI, ATM, etc..).

D. CORM Validation

The FBS framework developed in [5], and illustrated in Fig. 2 is applicable to any engineering discipline, for reasoning about, and explaining the nature of the design process [13]. In this section, we aim to validate the derivation of CORM's BAU, the CORM-NC, using the FBS framework. The inception point for CORM-NC design is marked by our design principles. According to which, computer networks need to be designed as a software-dependent CAS that exhibit emergent behavior. CORM design principles formed our first set of requirements F<sub>1</sub> and expected behavior Be<sub>1</sub> as follows:

$$F_1 = \text{CAS (autonomous entities, complexity)}$$

$$Be_1 = \text{Emergent Behavior (adaptation, self-organization, evolution)}$$

Shifting to the structure that can deliver F<sub>1</sub> and Be<sub>1</sub>, we attempted a *catalog lookup* by exploring natural complex systems, and studying their structure (S), and the individual behavior of their components (Bs). Our research led us to a recent study on primordial bacterial colonies [12]. This point marked our first functional reformulation. We formulated new requirements F<sub>2</sub> for designing a network cell that mimics the bacterium cell behavior Be<sub>2</sub>. Accordingly, we synthesized the

TABLE I. CORM VS. LAYERED NETWORK MODELS

Features/ Model	CORM	Layered Models
BAU	NC	Layer
Operation of BAU	Independent: CORM-NC can exist and operate by itself	Dependent: a single layer can never exist or operate by itself
BAU responsibilities	(1)Execution of assigned network functions (2)Self-monitoring and regulation	Execution of assigned network function
BAU Relationships	Interdependent: NC realizes other NCs and cooperate to maximize the over all performance by adapting to context	Incognizant: A layer at one level uses services from the layer below and provides services to the layer above, while being incognizant to the presence of other layers that are at one level further
System Level awareness	Global awareness: CORM-NCs have a sense of global system goal	Unaware of the global system: Awareness is restricted to the layer boundary
Network Composition	Bottom-up recursive	Top-down incremental overlaying

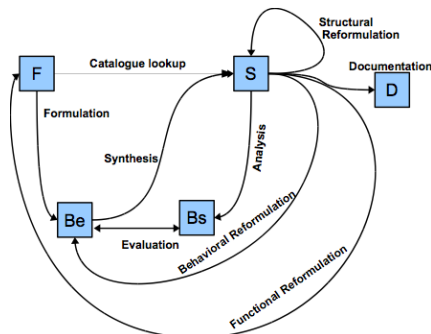


Figure 2. Gero's FBS (adapted from [5])

structure  $S_2$  from  $Be_2$  presenting the NC. However,  $F_2$ ,  $Be_2$ , and  $S_2$  needed further reformulation to detail network requirements. At this point, we defined the network requirement specification that led to the derivation of the ACRF framework for network concerns, yielding a new set of requirements  $F_3$ .  $F_3$  was integrated with  $F_2$ , and super-imposed over  $Be_2$ , and  $S_2$  to customize each towards the context of computer networks leading to the derivation of the CORM-NC.

CORM-NC delineates the BAU from which the network can be recursively built. However, at this point of our research, we still have not completely defined  $Bs$  for CORM-NC, since this will involve defining performance variables, and their range of values for the software code that will run within each unit of the NC structure. Nevertheless,  $Bs$  is accounted for by specifying the IFM as an essential part of CORM.

#### IV. DERIVING AND EVALUATING A CORM-BASED ARCHITECTURE

The key difference between a reference model and an architecture is the level of concept abstraction that the model conveys, as well as the degree of requirement specifications that the model addresses. CORM expresses the most fundamental design principles for engineering computer networks at the highest level of abstraction. To derive an architecture from CORM further specifications regarding network operational context, performance requirements, and/or constraints need to be identified.

##### A. CAHN: A CORM-Based Architecture for Ad Hoc Networks

We define CAHN's requirement specifications as follows;

- Minimal architecture that provides core network functionalities: CAHN should be able to provide basic communication and transport services equivalent to that supported by the TCP/IP suite.
- Cross-interest management: CAHN should provide a systematic way for dealing with cross interests among the supported network functionalities.
- Modular: CAHN abstractions should separate functions into modules with clear defined interfaces.

Based on CORM's NST and ACRF, and guided by the above requirements, we define CAHN-Ncomp to be composed of four CORM-NCs, each instantiating the concerns defined by the ACRF. Accordingly, CAHN abstractions are the following concern-specialized CORM-NCs; Application Network Cell (ANC), Communication Network Cell (CNC), Resource Network Cell (RNC), and Federation Network Cell (FNC).

CAHN networks will be composed of CAHN-Ncomps, each of which will be composed of ANC, CNC, RNC and FNC.

##### B. Engineering Protocols for CAHN

Protocol engineering in CAHN need to be classified according to the ACRF framework, and thus executed by the corresponding concern-specialized NC. Moreover, the task performed by each protocol (NC) will be internally classified according to the ACRF, as defined by the CORM-NC. To clarify this recursive assignment of the ACRF framework, we present an example for the routing function in CAHN.

According to the ACRF classification, the routing function is a CCn, which will be represented as a CNC in CAHN. However, routing as a function is a composite task that can be further divided into several subtasks such as, naming, addressing, forwarding, routing table creation and maintenance, etc. These identified subtasks will be recursively classified according to the ACRF. Following is an example of such classification:

- CNC-ACn: The application concern (ACn) of the CNC will be responsible for setting the routing protocol policies, which determines the quality of the routes to be discovered, and how the routes will be maintained. The CNC-ACn decisions will partially depend on the communication profile that is received from the ANC. This communication profile will indicate the destination and priority of the flow that is to be administered into the network, and the quality required for the end-to-end route
- CNC-CCn: Depending on the CNC-ACn requirements and policies, the CNC-CCn will decide on the appropriate routing protocol to be instantiated. The instantiation of a routing protocol depends on the micro-routing-protocols available on the CNC, from which a routing function can be devised. Alternatively, a default routing protocol can be adapted to the ACn requirements. CNC-CCn will also decide on link parameters, since route definition depends mainly on link characteristics. This introduces a cross interest between CNC and RNC, which will be handled by the FNC. Other communication tasks handled by the CNC-CCn include resolving routes, sending and receiving route requests and replies, communicating with neighbors, forwarding packets, etc.
- CNC-RCn will be responsible for estimating and managing the resources assigned to the CNC.
- CNC-FCn is responsible for monitoring and regulating the performance of the CNC. Parameter monitored by the CNC-FCn are specified once the CNC get specialized, and are subject to adjustments and/or amendments if required. Parameters monitored can either be specific, pertaining to the communication task assigned to the CNC, or general, relating to the over-all performance of the CNC. The CNC-FCn has a regulation cycle that will constantly check the performance of the communication related functions in specific, and the CNC operation in general, by comparing the values of the monitored parameters to thresholds values previously defined in a knowledge database stored in the FCN. If the monitored values fall below the indicated thresholds, the FCN will

interfere to regulate the operation of the CNC. Furthermore, the FCn can decide on any optimizations required to improve the performance, or it can interfere to resolve any cross interests that might rise among the core-concerns within the CNC. For example, the memory required by the routing table could exceed the space assigned to the CNC. In such a case, the FCn, after consulting its knowledge-base, could either instruct the CCn to alter its route-purging policy, or command the RCn to request more memory space.

### C. CAHN Evaluation

CAHN is evaluated by simulating a CAHN-based network in the ns2 simulator [14]. Our simulation is based, in part, on the simulation in [15], in which a cross-layer power adaptation algorithm was devised for ad hoc networks. The algorithm in [15] integrated the operation at the Network, MAC and the physical layers to tune the transmission power of a node according to the number of its neighbors, in an attempt to minimize MAC contention, while maintaining network connectivity. However, such optimization had adverse effects on TCP traffic due to network oscillation between connectivity and dis-connectivity. This highlights the pitfalls of cross-layer adaptations that result in unintended consequences, when protocols at different layers operate with conflicting interests. We conjecture that CAHN-based networks can counteract such conflicting interests. Hence, we simulated CAHN-Ncomp on ns2 nodes by adjusting the ns2-code for the TCP, AODV and MAC to comply with the ACRF framework, as well as with CAHN-NCs. Thus any subsequent reference to these protocols will relate to their modified version. We define the performance parameters in CAHN simulation as; 1) the power level that results in minimum MAC contention, while sustaining next-hop transmission at RNCs and CNCs, respectively, 2) next hop neighbor at the CNCs, 3) and the TCP congestion window size at the source ANC (refer to [3] for details of adjusting congestion window size to path capacity). These parameters will be monitored and regulated by the FCns of the corresponding CAHN-NCs. Our simulation is divided into two phases. Phase 1 is a learning-adaptation phase, where an adapted version of the power adaptation algorithm in [15] controls the transmission power. In this phase, the FNC populates its knowledge-base with information about the level of performance attained relative to the values assumed by the monitored parameters. In phase 2, the FNCs residing on CAHN-Ncomps, manage cross-interests among the performance parameters, and choose combined optimal-values that support the TCP flow. Hence, the FNCs prevent the oscillations reported in [15]. Figs. 3 and 4 plot the recorded TCP throughput at the sink nodes in ns2 simulations, in case of the cross-layer power algorithm as implemented in [15], versus CHAN.

### V. CONCLUSION

This paper proposes CORM, a concern-oriented reference model for future computer networks. CORM is based on two design principles that realize the network as a software-dependent CAS. CORM refutes the long endorsed concept of layering, intrinsically accounts for emergent behavior, and ensures network congruency. We used the FBS engineering framework to validate CORM's BAU, the NC, then derived and evaluated an architecture based on CORM through simulation.

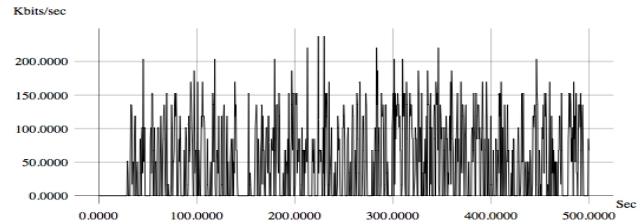


Figure 3. TCP-Sink throughput using cross-layer power adaptation[15]

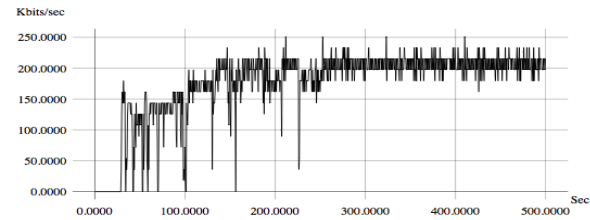


Figure 4. TCP-Sink throughput using CAHN

### VI. REFERENCES

- [1] H. Hassan, "A Reference Model and Architecture for Future Computer Networks," Ph.D. Dissertation, Virginia Polytechnic and State University, Blacksburg VA, May 26, 2010
- [2] H. Hassan, R. Eltarras, and M. Eltoweissy, "Towards a Framework for Evolvable Network Design," Collaborate Com 2008, Orlando, FL, USA.
- [3] H. Hassan, et al., "CellNet: A Bottom-Up Approach to Network Design," 3rd International Conference on New Technologies, Mobility and Security, Cairo, Egypt, December 2009.
- [4] F. Polack et. al., "Complex Systems Models: Engineering Simulations," in ALife XI. MIT press, 2008.
- [5] J.S. Gero "Design Prototypes: A Knowledge Representation Schema for Design," AI Magazine, vol. 11, pp. 26–36, Winter,1990.
- [6] Clark, "NSF Future Internet Summit," Meeting Summary, Washington, DC October 12-15, 2009. Version 7.0 of Jan. 5, 2010
- [7] D. Stevenson, et. al., "On the Suitability of Composable Services for the Assurable Future Internet," Military Communications Conference, Orlando, FL, USA, 2007.
- [8] J. Touch, Y. WangV. Pingali, "A Recursive Network Architecture," <http://www.isi.edu/touch/pubs/isi-tr-2006-626/isi-tr-2006-626.pdf>. <retrieved, 12, 2011>
- [9] V. Jacobson, et. al., "Networking Named Content," CoNEXT'09, December 1–4, 2009, Rome, Italy.
- [10] Melanie Mitchell, "Complex systems: Network thinking," Artificial Intelligence, Volume 170, Issue 18, December 2006, pages 1194-1212.
- [11] "Reference Model for Service Oriented Architecture 1.0," Committee Specification 1. Aug.2006. <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf> <retrieved, 12,2011 >
- [12] E. Jacob & H.Levine, "Self-engineering Capabilities of Bacteria," Jr of Royal Society. Interface, <http://star.tau.ac.il/~eshel/papers/Interface.pdf>.<retrieved, 12, 2011 >
- [13] P. Kruchten, "Casting Software Design in the Function-Behavior-Structure Framework," IEEE Software, vol.22, pp.52-58, 2005.
- [14] "The Network Simulator – ns-2," <http://isi.edu/nsnam/ns/><retrieved, 12, 2011 >
- [15] V. Kawadia & P. R. Kumar, "A Cautionary Perspective on Cross Layer Design," IEEE Wireless Commun., vol. 12, no. 1, Feb. 2005, pp. 3–11.