volume 1 number 1
december 2010

International Magazine on Advances in Computer Science and Telecommunications

Contents

Association for Computer Science and Telecommunications AKOMNAT TEL DOO

IMACST 1 (1) 1-47 (2010)
ISSN 1857-7202
www.imacst.com

International Magazine on Advances in Computer Science and Telecommunications

VOLUME 1 NUMBER 1 DECEMBER 2010

Editors in Chief

Associate Editors

International Magazine on Advances in Computer Science and Telecommunications
VOLUME 1 NUMBER 1 DECEMBER 2010

CONTENTS

IMACST is published by the Association for Computer Science and Telecommunications AKOMNAT TEL DOO.
IMACST is an open access publication.

IMACST 1 (1) 1-47 (2010) ISSN 1857-7202

www.imacst.com

Partially-Redundant Systems: Examples, Reliability, and Life Expectancy

Ali Muhammad Ali Rushdi

Abstract— **This paper is a brief tutorial exposition of some recent developments in the evaluation of the reliability of partially-redundant (k-out-of-n) systems. A novel contribution of the paper is that it identifies many practical examples of such systems, which are spread across a wide spectrum of engineering disciplines, including, in particular, the areas of computer and telecommunication engineering. Some formulas for the reliability and life expectancy of these systems are discussed in the case of equal-reliability components. Certain celebrated formulas are shown to be numerically unstable and totally useless in the case of large systems with high-reliability components. In fact, these formulas are highly susceptible to round-off errors and severely suffer from catastrophic cancellations. The paper also reviews how the Boole-Shannon expansion (or equivalently, the pivoting or factoring technique) is used to derive pertinent recursive relations, leading to a highly efficient algorithm for k-out-of-n reliability evaluation. This algorithm has a nice interpretation in terms of a regular Mason signal flow graph, which turns out to be (a) a reduced ordered binary decision diagram representing a monotone symmetric switching function, and (b) analogous to the minimal circuit realization of this function. In the worst case, the temporal and spatial complexities of this algorithm are shown to be quadratic and linear, respectively, in the number of system components. The paper lists some extensions and applications of this algorithm and compares it with a few related algorithms. The paper concludes with a quick consideration of some important issues in the area of k-out-of-n system reliability, including the issues of useful redundancy, criticality measures, and cost.**

Index Terms— **Coherent system, k-out-of-n system, Normalized life expectancy, Reliability, Catastrophic cancellation, ROBDD, Useful partial redundancy.**

I. Introduction

THE reliability R(t) of a component or a system is the probability that it will adequately perform its specified purpose/job/function for a specified semi-closed period of time (0, t] under specified environmental conditions. Implicit in this definition is the assumption or condition that the pertinent component or system is initially good (*i. e.,* at t = 0, R = 1.0). The field of system reliability deals with the relation between the reliability of a system and the reliabilities of its components. At the core of this field is the concept of the k-out-of-n:G(F) system (also called a partially-redundant system), which is a system of n components that functions (fails) if at least k out of its components function (fail) [1-14]. Typically, the k-out-of-n system (for $1 \le k < n$) is intended to provide *useful redundancy*, i. e., to have a reliability better than that of the simplex or single-component reliability. This necessitates that the simplex reliability itself be good enough (based on the values of k and n). For the same simplex reliability, more useful redundancy is achieved the lower k is for a fixed n, and the higher n is for a fixed k.

The k-out-of-n system has many attractive features. It has a symmetric structure that has many convenient mathematical descriptions such as Boolean expressions, recursive equations, generating functions and so on. Nevertheless, for $1 < k < n$, the k-out-of-n system lacks a graphical representation in the form of a network or a fault tree, unless replicated network edges or fault tree inputs are allowed. The k-out-of-n system plays a central role for the general class of coherent systems, as it can be used to approximate the reliability of such systems [15], and its reliability function is the steepest function among all coherent reliability functions of order n [8]. While virtually all nontrivial network reliability problems are known to be NP-hard for general networks [16], the regular structure of the k-out-of-n system allows the existence of simple efficient algorithms for its reliability analysis that are of quadratic-time linear-space complexity in the worst case [2-4, 6].

The k-out-of-n system is being rediscovered in the literature from time to time without being identified as such (See, e.g., [17] and [18], where that system, as well as extensions and compositions thereof appear in disguise). We hope that the present exposition may remedy this situation (at least partially), by familiarizing a large readership with an extensive number of examples of the k-out-of-n system. Though the results of this paper are relating system reliability to component reliabilities, these results are also applicable in the context of availability.

This paper is intended to be a review or a tutorial exposition, and we hope to make it of a significant pedagogical utility. We strive strongly for simplicity and clarity to allow a non-expert reader to easily follow our discussion. Therefore, we deliberately include certain details and explanations of

Manuscript received August 30, 2010.

A. M. A. Rushdi is with the Department of Electrical and Computer Engineering, King Abdulaziz University, P. O. Box 80204, Jeddah 21589, Saudi Arabia. (Email: arushdi@kau.edu.sa)

1857-7202/1008007

terminology that experts might consider obvious or even trivial. The absence of such details in some papers has occasionally led to misunderstanding and pitfalls. Mostly, we do not start our reliability analysis directly in the probability (algebraic) domain, but we initiate it in the switching (Boolean) domain, and then transform our results to the algebraic domain. We try to fully utilize the wealth of knowledge available for switching (Boolean) algebra [19]. We adopt the special symbols representing the operators of this algebra instead of borrowing symbols from real algebra, which could be confusing and misleading [20]. In particular, we use the symbol (**∨**) (rather than the symbol (+)) for the *OR* switching operation. For convenience, we will use R(**p**) or R(p) to denote the system reliability for non-identical component reliabilities **p** or a common component reliability p, i.e., we will make the time dependence of R implicit through the time dependence of **p** or p.

The organization of the remainder of this paper is as follows. Section II lists the assumptions, notation and nomenclature employed throughout the rest of the paper. Section III identifies many reliability systems related to the k-out-of-n system, while Section IV lists many practical examples of that system, which are spread across a wide spectrum of engineering disciplines, including, in particular, the areas of computer and telecommunication engineering. Closed-form formulas for k-out-of-n system reliability and life expectancy are given in Section V for the case of equal-reliability components. Section V also adds some pictorial insight to the topic, and points out a pitfall that went practically unnoticed in the reliability literature. Section VI discusses how the Boole-Shannon expansion (or equivalently, the pivoting or factoring technique) is employed to derive binary-recursive relations as well as an efficient iterative algorithm for computing the reliability of a k-out-of-n system with non-identical components. Section VII discusses complexity issues for this algorithm and compares it with other notable algorithms. Section VIII concludes the paper.

II. Assumptions, Notation, and Nomenclature

A. Assumptions

- Both the components and the system are of two states, i.e., either good or failed.
- Component states are statistically independent.
- The system is a mission-type one, i. e., without repair.

B. Notation

n = number of system components, $n \geq 0$.

X_i = success of component i = indicator variable for successful operation of component i = a switching random variable that takes only one of the two discrete values 0 and 1; ($X_i = 1$ iff component i is good, while $X_i = 0$ iff component i is failed).

$\overline{X_i}$ = failure of component i = indicator variable for unsuccessful operation of component i, ($\overline{X_i} = 0$ iff component i is good, while $\overline{X_i} = 1$ iff component i is failed). The success X_i and failure $\overline{X_i}$ are complementary variables.

$\boldsymbol{X}$ = a vector of n elements representing the component successes $\boldsymbol{X} = [\, X_1\, X_2 \ldots\, X_n\,]^T$.

$S(\boldsymbol{X})$ = indicator variable for the successful operation of the system, called system success.

$S(k, n, \boldsymbol{X})$ = success of a k-out-of-n:G system of component successes $\boldsymbol{X}$, $0 \leq k \leq (n+1)$.

$Sy(A, \boldsymbol{X})$ = a symmetric switching function of the switching *variables* $\boldsymbol{X}$, *where*:

A = characteristic set = $\{a \mid 0 \leq a \leq n$, a is the number of 1's assignments of the X_i's for which the symmetric function is 1$\}$.

$Pr[\ldots]$ = probability of the event $[\ldots]$.

$E[...]$ = expectation of the random variable [...].

p_i, q_i = reliability and unreliability of component i; Both p_i and q_i are real values in the closed real interval [0.0,1.0].

$p_i = Pr[\, X_i = 1\,] = E[\, X_i\,] = 1.0 - q_i$.

$\boldsymbol{p}$ = a vector of n elements representing the component reliabilities, $\boldsymbol{p} = [p_1\ p_2 \ldots\ p_{m-1}\ p_m\ p_{m+1} \ldots\ p_n]^T$.

$\boldsymbol{q}$ = a vector of n elements representing the component unreliabilities = $\mathbf{1.0} - \boldsymbol{p}$, where **1.0** is an n-tuple of real 1's.

$\boldsymbol{p}/p_m$ = a vector of (n-1) elements obtained by omitting the mth element of vector $\boldsymbol{p}$.

$\boldsymbol{p}|_{j_m}$ = a vector of n elements obtained by setting the mth element of $\boldsymbol{p}$ to j which is either 0 or 1.

$R(\boldsymbol{p}), U(\boldsymbol{p})$ = reliability and unreliability of the system. Both $R(\boldsymbol{p})$ and $U(\boldsymbol{p})$ are real values in the closed real interval [0.0,1.0].

$R(\boldsymbol{p}) = Pr\,[\, S(\boldsymbol{X}) = 1\,] = E[\, S(\boldsymbol{X})\,]$.

$U(\boldsymbol{p}) = 1.0 - R(\boldsymbol{p})$.

$R(k, n, \boldsymbol{p})$ = reliability of a k-out-of-n:G system of component reliabilities $\boldsymbol{p}$, $0 \leq k \leq (n+1)$.

$R(k, n, p)$ = the value of $R(k, n, \boldsymbol{p})$ when component reliabilities are all equal to a common value p.

$U(k, n, \boldsymbol{p})$ = unreliability of a k-out-of-n:G system of component reliabilities $\boldsymbol{p}$, $0 \leq k \leq (n+1)$.

$U'(k, n, \boldsymbol{q})$ = unreliability of a k-out-of-n:F system of component unreliabilities $\boldsymbol{q}$, $0 \leq k \leq (n+1)$. This is the complemented dual of $R(k, n, \boldsymbol{p})$ in the sense that the successes of the k-out-of-n:G system and the k-out-of-n:F system are dual switching functions.

$$U(k, n, \boldsymbol{p}) = U'(n-k+1, n, \boldsymbol{q}). \quad (1)$$

$c(k, n)$ = the binomial (combinatorial) coefficient = the number of ways of choosing k objects from a set of n objects, when repetition is not allowed and order does not matter. Binomial coefficients satisfy Pascal's identity

$$c(k, n) = c(k, n-1) + c(k-1, n-1), \quad 0 < k < n, \tag{2}$$

together with the boundary conditions

$$c(k, n) = 1, \ (k = 0 \text{ or } k = n) \text{ and } n \geq 0. \tag{3}$$

E/F = the set difference of sets E and F, $E/F = \{j \mid j \in E, j \notin F\}$.
$|Y|$ = cardinality of the finite set Y = the number of elements in the set Y.
$\lceil y \rceil$ = the ceiling of the real number y = the smallest integer greater than or equal to y.

C. Nomenclature

Duality: strictly speaking, the dual of switching function is obtained by complementing the function and all its switching arguments (inverting both output and inputs) [6]. In the reliability literature, "duality" is sometimes freely and loosely used to indicate "similarity," "analogy," or "being mirror images."

Monotone: a monotone system is one whose reliability function is a non-decreasing function in each component reliability, i.e.,

$$R(\boldsymbol{p} \mid 1_m) - R(\boldsymbol{p} \mid 0_m) = \partial R(\boldsymbol{p}) / \partial p_m \geq 0.0, \quad 1 \leq m \leq n. \tag{4}$$

Relevant: component number m is relevant to the system if there exists a valid value for $\boldsymbol{p}$ such that $\partial R(\boldsymbol{p}) / \partial p_m \neq 0.0$. Relevancy means that $R(\boldsymbol{p})$ is not vacuous in (independent of) p_m.

Coherent: a coherent system is a monotone system whose components are all relevant [21]. If the reliability function $R(p)$ of a coherent system with equal-reliability components is plotted versus p within the square $0.0 \leq p \leq 1.0$, $0.0 \leq R(p) \leq 1.0$, then it satisfies $R(0.0) = 0.0$, and $R(1.0) = 1.0$, and exhibits an S-shape, *i. e.*, the curve $R(p)$ versus p is monotonically non-decreasing and if it crosses the diagonal (p versus p), it does so only once and from below [8, 22].

k-out-of-n:G system: a system that is good if and only if (iff) at least k out of its n components are good.

k-out-of-n:F system: a system that is failed iff at least k out of its n components are failed.

k-out-of-n (partially-redundant) system: a collective name for k-out-of-n:G and k-out-of-n:F systems; a k-out-of-n:F system is equivalent to an (n–k+1)-out-of-n:G system (as indicated by equation (1)). A k-out-of-n system is a coherent system in the practical case of $1 \leq k \leq n$, while it is only monotone for the hypothetical or fictitious limiting cases of k = 0 and k = (n+1).

s-p complex: a coherent system is series-parallel (s-p) complex iff it has no components in series or in parallel [23]. A k-out-of-n system is s-p complex for $1 < k < n$, and hence cannot be treated (even partially) by series-parallel reductions.

Pivoting: by pivoting on component number m, the system reliability $R(\boldsymbol{p})$ can be written as

$$R(\boldsymbol{p}) = q_m * R(\boldsymbol{p} \mid 0_m) + p_m * R(\boldsymbol{p} \mid 1_m), \tag{5}$$

where $R(\boldsymbol{p} \mid 0_m)$ and $R(\boldsymbol{p} \mid 1_m)$ are the reliabilities of the minors or subsystems of the original system with respect to component m. Pivoting is also called factoring and is equivalent to the total probability theorem [11] in the algebraic domain or to the Boole-Shannon expansion [19] in the Boolean domain.

III. Related Systems

The k-out-of-n:G system covers many interesting or limiting-case systems as special cases [6]. These include the fictitious perfectly reliable system (k = 0), the parallel system (k = 1), the voting or N-modular redundancy (NMR) system ($k = \lceil (n+1)/2 \rceil$), the fail-safe system (k = n – 1), the series system (k = n), and the fictitious totally unreliable system (k = n+1). Note that as k decreases (for a fixed n) from 0 to (n+1), the usefulness of the k-out-of-n:G system declines and finally diminishes. For $1 < k < n$, the k-out-of-n system is sometimes called a *partially-redundant system* [6, 24], as it lies somewhere between the extreme cases of the *(non-redundant) series system* and the *(fully-redundant) parallel system*. In this paper, we will view these two extremes of zero and total redundancies as limiting cases of partial redundancy, and hence consider a k-out-of-n system as synonymous to a partially-redundant system. The k-out-of-n:G system and the k-out-of-n:F system are *dual* or *mirror images* of one another; the k-out-of-n:G system being exactly equivalent to the (n–k+1)-out-of-n:F system (as formally indicated by equation (1) above).

The k-out-of-n: G system is a subclass of some important systems. Besides being a coherent system for the values $1 \leq k \leq n$, it is s-p complex for $1 < k < n$. It is also

a) a special case {r = n} of the consecutive-(n–k+1)-out-of-r-from-n:F system [25],
b) a limiting case { ℓ = n} of the generally non-coherent k-to- ℓ-out-of-n system [10, 26-28], which is useful in approximating the class of non-coherent systems [15], but does not precisely exhaust or cover such a class (contrary to a claim made in [28]),
c) a binary-state limiting case of the multi-state k-out-of-n system model [29].

An important generalization of the k-out-of-n:G(F) system is the *threshold system*, which can be neither symmetric nor coherent. A threshold system is a system whose success is a threshold (linearly-separable) switching function in the successes of its components [30]. This system is successful if and only if the weighted arithmetic sum of its component successes is equal to or exceeds a certain threshold. Therefore, a threshold system is characterized by (n+1) coefficients, namely, its threshold and the set of its n component weights (which are not necessarily unique). An important special case of the threshold system is the weighted k-out-of-n:G system, which is a coherent non-

symmetric system of strictly positive weights and a threshold equal to k [12, 31]. If further, all the weights are equal to 1, the weighted k-out-of-n:G system reduces to the ordinary k-out-of-n:G system. Therefore, the k-out-of-n:G system can be defined as a threshold system with a common positive weight for its components and a threshold equal to k multiplied by this common weight [30].

A system that is closely related to the k-out-of-n:G(F) system is the *consecutive k-out-of-n:G(F) system*, which functions (fails) if at least *k* consecutive components function (fail) [32, 33]. The k-out-of-n:G(F) system and the consecutive k-out-of-n:G(F) system are not generally comparable since neither of them is a subclass of the other (except when they overlap at their limiting cases $k \leq 1$ and $k \geq n$). The k-out-of-n system is a threshold system, but generally the consecutive-k-out-of-n system is not. The k-out-of-n system is structurally symmetric, i.e., the order of its components is immaterial, while the set of components in a consecutive k-out-of-n system is an ordered one (either on a line or on a circle, corresponding to linear and circular versions of the system). The failure switching function of the consecutive k-out-of-n:F system implies that of the k-out-of-n:F system, and hence the reliability of the latter system is a lower bound of that of the former system.

Yet another system that is also closely related to the k-out-of-n:G(F) system is the *(n, f, k) system*. This system is defined in [34] (based on a proposal in [35]) to consist of n components ordered in a line or a cycle, such that the system fails if, and only if, there exist at least f failed components or at least k consecutive failed components. This system reduces to the k-out-of-n:F system for $f \leq k$. A generalization of the (n, f, k) system is one with weighted components [36].

Occasionally, we might have a system consisting of certain subsystems, which in turn consist of lower-level subsystems, and so on, till we reach some innermost subsystems that consist of final components. If the relation between every sub(system) and its constituent components or subsystems is a k-out-of-n relation, the overall system is a *k-out-of-n composition* [8, 22]. Under appropriate conditions, the k-out-of-n composition can serve to achieve a dramatic increase in reliability by constructing ultra-highly-reliable systems out of typical or ordinary but somewhat good components. In such a composition, it is preferable to locate the more useful redundancies in the lower or innermost levels of the composition.

A system that would have been related to the k-out-of-n:G(F) system is the so called *strict consecutive-k-out-of-n:G(F) system* (or simply, *strict system*) proposed in [37]. The original definition of this system suffered from ambiguity or inconsistency. Rushdi [38, 39] attempted to produce well-defined versions of this system, either by accounting for statistical dependencies among its components, or by employing conditional probabilities. However, these versions lacked the claimed utility of the original system. Later, Rushdi [40] published his concerns about the strict system demanding a unique, precise, and consistent definition of it, enquiring about the real nature of some of its states, questioning whether it is coherent or non-coherent, and asking for an example showing its utility as a model for a real-life system. Evans (then Editor of the *IEEE Transactions on Reliability*) [40] responded that "*the problem as originally (and implied) [37] is an example of a collection of words that appears to make sense, but is actually self contradicting*." Fortunately, the questions posed by Rushdi [40] prevented the appearance of more algorithms that represent *correct* but *irrelevant* mathematics. These questions are hailed by Hwang [41] as an example of a critical review of literature that prunes overgrown branches in order to keep the growth in control. Such a pruning is always necessary when enthusiasm of expansion overextends its usefulness [41].

IV. Examples of Partial Redundancy

A model is a useful representation that captures the essence of a real system and behaves sufficiently like it in such a way that conclusions can be drawn from the model's behavior to aid in making prudent decisions about the real system. Situations in which the k-out-of-n:G(F) system serves as a useful model are frequently encountered in engineering practice and include the following examples:

a. A piece of stranded wire with n strands in which at least k strands are necessary to pass the required current behaves as a k-out-of-n:G system. The same concept generalizes to applications involving *supply-type* components with identical fixed ratings for their *capacity*, *flow*, *throughput*, *strength* or the like, such that system success is achieved when a minimum supply is met, or when a certain threshold is exceeded (see, e.g., [24, 30]).

b. A three-engine airplane which can stay in the air if and only if at least two of its three engines are functioning is a 2-out-of-3:G (also called a 2-out-of-3:F system or a triple modular redundancy (TMR)) [13]. A space vehicle requiring three out of its four main engines to operate in order to achieve orbit is a 3-out-of-4:G and also a 2-out-of-4:F system [9]. The common practice of having a spare tire in a 4-wheel car constitutes a 4-out-of-5:G and also a 2-out-of-5:F system. All these are examples of a *fail-safe system, i. e.*, a system that tolerates the failure of one (and only one) component, since such a failure reduces the system to a series system, which is still a working system (albeit with no more redundancy, and hence no capability to withstand any further failure). The idea of a fail-safe system works well provided the assumption of statistical independence among components holds. A car driver cannot rely on a single spare tire on a rough unpaved road that might result in a double flat tire (instantaneous or common-cause failures). Thanks to a merciful Providence, the fail-safe concept is entrenched in many biological systems. For example, a human being can survive the failure of one of his two kidneys, which constitute a 1-out-of-2:G system.

c. Reactor protection systems, sensor systems, alarm generation systems and other *decision mechanisms* usually employ a k-out-of-n:G voting logic [42]. *Voting* is also

used in the realization of ultra-reliable systems that are based on multichannel computations [43]. Likewise, voting is commonly used in faulty distributed computing systems to achieve mutual exclusion among groups of isolated nodes.

d. A bridge with n main supports that can survive an earthquake if and only if at least k supports remain intact is approximately modeled as a k-out-of-n:G system. Here, the modeling is *qualitative* rather than quantitative, since a bridge is usually not structurally symmetric with respect to its supports, while a k-out-of-n system is structurally symmetric with respect to its components.

e. In a binary communication channel, an *error-correcting code* might be employed in the transmission of n-bit code-words [44]. If the code is capable of correcting up to k bit errors, word transmission becomes a (k+1)-out-of-n:F and also a (n-k)-out-of-n:G system. A code lacking any error-correcting capability (k = 0) {such as the BCD code without any parity check} is a series system, while a code of a Hamming distance 3 (k = 1) {such as the famous (7,4,1) Hamming code} is a fail-safe system.

f. A bus-structured multiprocessor computer system consists of n processors sharing m memory units via b common buses. If this system is required to operate in *MIMD mode* (i.e., with Multiple Instruction streams and Multiple Data streams), then it is logically equivalent to the series connection of a k_1-out-of-n:G system, a k_2-out-of-m:G system and a 1-out-of-b:G system, where $k_1 \geq 2$ and $k_2 \geq 1$ with the precise values of k_1 and k_2 being determined by system requirements [45].

g. In the majority voting (MV) algorithm for managing replicated data, out of n copies of an object $\lceil (n+1)/2 \rceil$ copies must be up to form a quorum [18]. This is an $\lceil (n+1)/2 \rceil$-out-of-n:G system. A generalization of this algorithm, the *hierarchical quorum consensus* (HQC) algorithm is a multilevel system in which the availability of a group at level i expressed in terms of the availability of its subgroups at level (i+1) constitutes a k_i-out-of-n_i:G system. This means that HQC is nothing but an *iterative or repeated composition* (See, e. g., [8, pp. 202-203] or [22, pp. 203-206]) of the MV structure. Such a composition improves availability (making HQC superior to MV) provided the basic component availability is higher than a certain value [8, 22, 46].

h. The k-out-of-n model is useful in the study of *multistage interconnection networks* [47]. For example, the terminal reliability of a Gamma network [48] is represented by a ladder network (of unreliable nodes and perfect links) whose behavior can be approximated by that of a k-out-of-n system. Specifically, the ladder network in [48, Fig. 5] and in [47, Fig. 1] is logically equivalent to a series connection of two components with a structure that has 8 cut sets of two components each. The reliability of a 2-out-of-6:F system is a lower bound for the reliability of this structure.

i. The k-out-of-n model is used in the petro-chemical industry in the evaluation of the life of *furnace systems* and in decision making on when to replace the furnaces [49, 50]. The furnaces are considered to be systems while the tubes in the furnaces are the components of the corresponding systems. A tube is designed to provide an environment for methane, steam, and a catalyst to react at a high temperature to produce hydrogen. A tube is considered failed when it is unable to perform its intended function any more (for example, when it is ruptured or is pinched). The function of a furnace is to produce hydrogen at certain output, temperature, pressure, and efficiency [49]. If too many tubes (k or more out of n) are failed, the furnace's proper operation is affected, and it is failed. This is a k-out-of-n:F system.

j. In mining operations, a shovel-truck system in an open mine usually consists of a shovel and a fleet of n trucks (say 20 trucks). The system functions properly if at least k trucks (say 15 trucks) and the shovel are good. This is a series system of 2 subsystems: the shovel and a 15-out-of-20:G system. If the shovel is assumed to be perfectly reliable, the system becomes simply a 15-out-of-20:G system. In general, the k-out-of-n concept is useful in modeling many types of *fleets of vehicles*, including aircrafts, ships, buses, and trains.

k. In a *perfect secret sharing scheme (PSSS)*, a secret is to be divided into shares, and distributed among members. The secret can be determined, *i.e.*, the system works when k or more distinct members collaborate together, but only k shares are required to reconstruct the secret. In the context of perfect secret sharing, the secret can be reconstructed with any k or more members, but (k–1) or fewer members cannot reveal anything about the secret [51]. The PSSS is nothing but a k-out-of-n:G system, in the sense that the reliability of this system expresses the probability of constructing the secret as a function of the probabilities of member contributions to such a construction.

V. Systems with Components of Equal Reliabilities

A. Reliability

The reliability of a k-out-of-n:G system (with independent components of identical reliabilities p) equals the probability of at least k successes in n Bernoulli trials, and hence it is given by [6]:

$$R(k, n, p) = \sum_{i=k}^{n} c(\mathrm{i}, \mathrm{n})\ p^{\mathbf{i}}\ (1-p)^{\mathbf{n}-\mathbf{i}}, \tag{6}$$

$$= \sum_{m=k}^{n} (-1)^{\mathbf{m}-\mathbf{k}}\, c(\mathrm{k}-1, \mathrm{m}-1)\ c(\mathrm{m}, \mathrm{n})\ p^{\mathrm{m}}. \tag{7}$$

Formula (7) is considered more suitable than formula (6) for hand calculation [11, 52], because (7) expresses *R(k, n, p)* as a polynomial of *p* only, while (6) involves powers of both *p* and (1.0 – *p*). In fact, formula (7) seems more convenient for symbolic differentiation needed to express the instantaneous hazard rate *(– (dR(t)/ dt) / R(t))*, and for symbolic integration required to find the life expectancy according to the forthcoming equation (8).

B. Life Expectancy

The life expectancy or Mean Time To Failure (*MTTF*) of a general non-repairable or mission-type system is given (under appropriate assumptions on the behavior of *R(t)* as t tends to infinity) by

$$T = MTTF = \int_0^\infty R(t)\,dt. \tag{8}$$

For a *k*-out-of-*n*:G system having components subject to a common constant failure rate (CFR) λ, the component reliability is

$$p(t) = e^{-\lambda t}, \qquad t \geq 0, \tag{9}$$

so that the *MTTF* of a single component is (1/ λ), while the *MTTF* of the system itself is obtained from equations (7)–(9) as

$$\begin{aligned} T &= \int_0^\infty \sum_{m=k}^{n} (-1)^{m-k} c(k-1, m-1)\, c(m,n) (e^{-\lambda t})^m \, dt, \\ &= \sum_{m=k}^{n} \frac{(-1)^{m-k}}{\lambda m} c(k-1, m-1)\, c(m,n). \end{aligned} \tag{10}$$

We call the dimensionless product (λT) the *normalized life expectancy* of the system, since it is the quotient of the actual life expectancy of the system by the life expectancy of a single component. A simpler expression for *T*, can be obtained from the Markovian state diagram for a *k*-out-of-*n*:G system [53], namely

$$T = \frac{1}{\lambda} \sum_{m=k}^{n} \frac{1}{m}. \tag{11}$$

C. Computational Accuracy

Figure 1 demonstrates our computational experience with formulas (6) and (7). Specifically, Fig. 1 presents the reliability *R*(*k, n, p*) versus *n* for a k-out-of-*n*:G system with k = 20 and a component reliability p = 0.9. In Figs. 1(a) and 1(b), the system reliability *R(k, n, p)* is computed via formulas (6) and (7), respectively, for *n* varying from 20 to 80. While formula (6) maintains its stability at values of *R*(*20, n,* 0.9) approximately equal to 1.0 as *n* grows, formula (7) produces results that grow to extremely high values (up to the order 10^{18}, drastically violating the restriction R ≤ 1.0). As a matter of fact, the two reliability equations start off identically equal or almost equal up to about *n* = 40 before the results produced by formula (7) start to deviate and overflow, as demonstrated in Fig. 1(c). To quantify the comparisons of (6) versus (7), numerical values of the computed reliability are summarized in Table I, for values on *n* between 20 and 80.

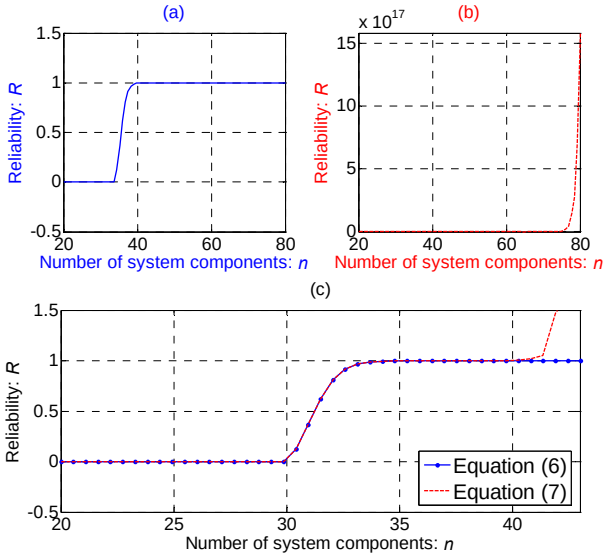


Figure 1. Reliability *R*(20, *n*, 0.9) versus the number of system components *n* for a 20-out-of-n:G system, as computed via: (a) Equation (6) for *n* between 20 and 80, (b) Equation (7) for *n* between 20 and 80, and (c) Equations (6) and (7) overlapped for *n* between 20 and 45.

Similarly, Figure 2 illustrates the computational accuracy of formulas (11) and (10). In Figs. 2(a) and 2(b), the normalized life expectancy (λT) is computed via formulas (11) and (10), respectively, for *n* varying from 20 to 80. Similarly to formula (7), formula (10) goes fast to very high erroneous values as *n* grows, while formula (11) continues to produce accurate values. Figure 2(c) verifies the fact that the results of the two life expectancy formulas stay almost equal up to about *n* = 40 before the results produced by formula (7) start to deviate and overflow. To quantify the comparisons of (11) versus (10), numerical values of the normalized life expectancy are summarized in Table I, for values on *n* between 20 and 80.

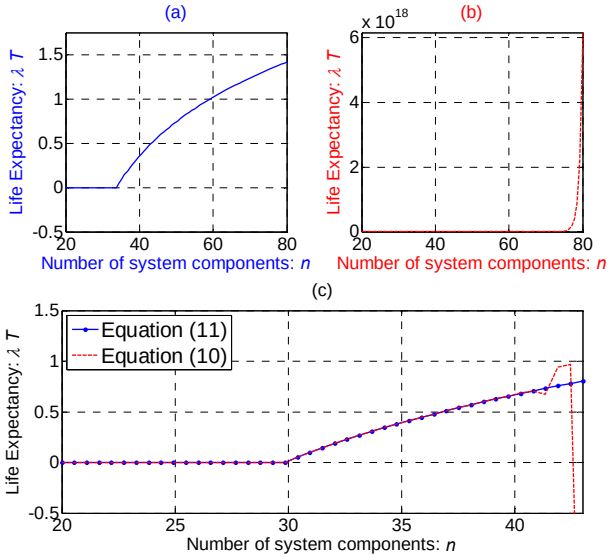


Figure 2. Normalized life expectancy (λT) versus the number of system components n for a 20-out-of-n:G system, as computed via: (a) Equation (11) for n between 20 and 80, (b) Equation (10) for n between 20 and 80, and (c) Equations (11) and (10) overlapped for n between 20 and 45.

TABLE I. NUMERICAL RELIABILITY AND LIFE EXPECANCY VALUES

Value of n	$R(20, n, 0.9)$		λT	
	Formula (6)	*Formula* (7)	*Formula* (11)	*Formula* (10)
20	0.1216	0.1216	0.0500	0.0500
30	0.9999	0.9999	0.4472	0.4472
40	1.0000	1.0518	0.7308	0.6744
50	1.0000	1.5169×10^{4}	0.9515	3.4090×10^{4}
60	1.0000	5.3790×10^{9}	1.1321	8.6123×10^{9}
70	1.0000	1.6780×10^{14}	1.2851	3.7537×10^{14}
80	1.0000	1.5833×10^{18}	1.4177	6.1526×10^{18}

In passing, we need to alert the reader to some smoothing effect exhibited in the small-size drawings of Figs. 1(a) and 2(a). Though these two figures represent numerically stable computations, their initial values at n =20 are indistinguishable from zero though they are not zeroes (albeit somewhat small). According to Table I, the value of $R(20, 20, 0.9)$ is $(0.9)^{20}$ = 0.121576654 or approximately 0.1216, while the value of λT for n = 20 is (1.0/20) = 0.05. For n > 40, the graph of Fig. 1(a) is not distinguishable from 1.0, which is typical for highly- or ultra-highly-reliable systems. Table I is not better than Fig. (1) in this case, since it reports values of 1.0000 for $R(20, n, 0.9)$ and n > 40. Better accuracy or significance is attained for highly- or ultra-highly-reliable systems if we report their unreliabilities in floating-point arithmetic instead of their reliabilities in fixed-point arithmetic. This point is clarified by Table II, which present some k-out-of-n reliabilities in fixed-point arithmetic together with their complements or corresponding unreliabilities in floating-point arithmetic. Certainly a value of $R(20,45,0.9)$ expressed as $(1.0 - 3.6637\times10^{-15})$ is more informative and easier to grasp and comprehend than when expressed as (0.999999999999996).

TABLE II. ACCURATE RELIABILITY AND UNRELIABILITY VALUES

Value of n	*Formula* (6)	
	$R(20, n, 0.9)$	$1.0 - R(20, n, 0.9)$
20	0.121576654590569	8.7842×10^{-1}
25	0.966600055388505	33.3999×10^{-3}
30	0.999910922126177	8.9078×10^{-5}
35	0.999999937506507	6.2493×10^{-8}
40	0.999999999980403	1.9597×10^{-11}
45	0.999999999999996	3.6637×10^{-15}

Despite the fact that formulas (7) and (10) are frequently cited in the reliability literature, and despite their amicable suitability to hand calculations, they become totally worthless from the numerical point of view when dealing with large systems of high component reliabilities. These formulas are highly susceptible to round-off errors and severely suffer from catastrophic cancellations, and therefore they produce the highly erratic plots in Figures 1 (b), and 2(b). In retrospect, the undesirable behavior of formulas (7) and (10) should have been anticipated since they are essentially outcomes of the numerically-notorious *inclusion-exclusion principle* [6]. By contrast, formulas (6) and (11) are purely additive formulas and are really minimally insensitive to round-off errors.

D. Pictorial Representation

The unreliability $U(k, n, p)$ of a k-out-of-n:G system is the probability of at most $(k–1)$ successes in n Bernoulli trials, and hence it equals the *Cumulative Distribution Function (CDF)* $B(k–1, n, p)$ of the binomial distribution. Figure 3 shows a very regular *Mason signal flow graph (SFG)* that illustrates the computation of $B(i, j, p)$ [6, 54, 55]. Note that each diagonal arrow has a transmission equal to p, while each horizontal arrow carries a transmission equal to q = 1.0 – p. There are two types of nodes: (a) Source nodes of known values which are either black or white. A black node has a value of 1.0 while a white node has a value of 0.0, and (b) Non-source nodes drawn as shaded ones, which include (at least) one sink node whose value is the final result sought. Figure 3 is therefore a pictorial representation of the computation of $U(i+1, j, p)$. It can be made to represent the computation of $R(i+1, j, p)$ by either interchanging the colors (and values) of the source nodes, or swapping the symbols p and q [6]. If further, algebraic multiplication and addition are replaced by their logical counterparts (ANDing and ORing), Fig. 3 can also be made to represent the computation of the success function $S(i+1, j, \boldsymbol{X})$ [6], and can then be identified as a Reduced Ordered Binary Decision Diagram (ROBDD), which is well

known to be the state-of-the art data structure for encoding and manipulating switching functions [10, 56, 57]. Figure 3 is also analogous to tank circuits, which are minimal pass-network realizations of symmetric switching functions [58]. Moreover, Fig. 3 has certain similarities and minor dissimilarities with *Pascal's triangle* that reflect the similarities and dissimilarities of the forthcoming recursive equations (15a) and the boundary conditions (15c&d) with their counterparts (2) and (3). Fig. 3 has also good similarities and minor dissimilarities with the signal flow graph that illustrates the computation of the *probability mass function (pmf)* of the generalized binomial distribution [54, 55].

VI. Reliability Analysis for k-out-of-n Systems with non-identical components

There are three general classes of methods for system reliability analysis, namely; (1) the inclusion-exclusion method, (2) the methods of disjoint products, and (3) the Boole-Shannon expansion or equivalently pivoting (pivotal decomposition or factoring) [6]. These classes of methods are applicable (and have been extensively applied) to the reliability analysis of k-out-of-n systems. The *inclusion-exclusion principle* was the readily-available tool from classical probability theory [11], and naturally became the basis of early attempts for evaluating the reliability of k-out-of-n systems with non-identical components (See, e. g., [6, 59]). Though the computational disadvantages of this principle are now well known [6], new methods are still being forwarded that are reproducing its results, sometimes without identifying them as such. For example, the recent direct method in [60] reproduces (in disguise) *the improved inclusion-exclusion formula* (5.29) of [6] as its main result (1). The fact that these two formulas are identical becomes clear if one identifies the tabulated coefficient $b_k^{(m)}$ of [60] as a shifted binomial coefficient c(k–1, m+k–2). Nevertheless, [60] contributes a novel elegant proof via *mathematical induction* for the inclusion-exclusion result.

The study of system reliability has been classically achieved in terms of purely real-algebraic structure functions [8]. An equivalent approach is more *insightful* and *less error-prone* for the methods of disjoint products or the Boole-Shannon expansion. This approach is a logical formulation that utilizes the *isomorphism* between the algebra of events (set algebra) and the bivalent or 2-valued Boolean algebra (switching algebra). In this approach, one expresses the system success as a logical (switching or Boolean) function of the component successes. Next, one moves from the Boolean domain to the probability domain so as to obtain the system reliability as a function of the component reliabilities. This is facilitated by converting the switching (Boolean) expression for the system success into a *probability-ready expression (PRE)*, i.e., into an expression that is directly convertible, *on a one-to-one basis*, to a probability expression. Note that in a PRE: (a) all ORed terms (products) are disjoint, and (b) all ANDed alterms (sums) are statistically independent. The conversion from a PRE to a probability expression is trivially achieved by replacing Boolean variables by their expectations, AND operations by multiplications, and OR operations by additions [61-63]. The most powerful class of algorithms producing PREs are algorithms based on the use of the Boole-Shannon expansion [61-63]. In the following subsections, we describe how the Boole-Shannon expansion is used to derive pertinent recursive relations, leading to a highly-efficient algorithm for k-out-of-n reliability evaluation.

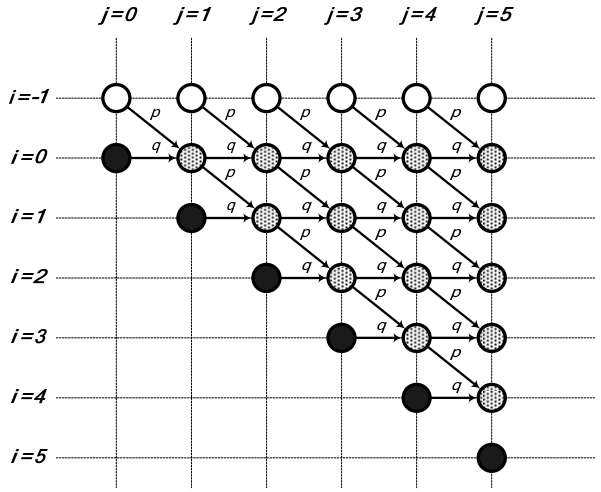


Figure 3. A Mason signal flow graph that illustrates the computation of the CDF B(i, j, p) of the binomial distribution.

A. Recursive Techniques

The Boole-Shannon's expansion [19] for a (totally) symmetric switching function of n variables $\boldsymbol{X}$ about one of these variables, namely, X_m, $1 \le m \le n$, can be stated as follows [3, 6]

$$Sy(A, \boldsymbol{X}) = \overline{X}_m \; Sy(B, \boldsymbol{X}/X_m) \vee X_m \; Sy(C, \boldsymbol{X}/X_m), \tag{12}$$

where the characteristic set A is a subset of $Z_{n+1} = \{0,1, ...,n\}$ and the sets B and C are subsets of $Z_n = \{0, 1, ..., n–1\}$. Let A be given by $A = \{a_0, a_1, ..., a_u\}$, $u \le n$, then, we have

$$B = A \cap Z_n = A \quad \text{if } a_u \neq n$$
$$= A/\{n\} \quad \text{if } a_u = n,$$

$$A_1 = \{a_0 - 1, a_1 - 1, \ldots, a_u - 1\},$$

$$C = A_1 \cap Z_n = A_1 \quad \text{if } a_0 \neq 0$$
$$= A_1/\{-1\} \quad \text{if } a_0 = 0.$$

Now, the success function of a k-out-of-n:G system is the symmetric monotone switching function [3, 6]

$$S(k, n, \boldsymbol{X}) = Sy(\{k, k+1, ..., n\}, \boldsymbol{X}). \tag{13}$$

For $1 \le k \le n$, $S(k, n, \boldsymbol{X})$ can be expanded via Eq. (12) with $A = \{k, k+1,..., n\}$, $B = \{k, k+1,..., n–1\}$ and $C = \{k–1,k,...,n–1\}$, which produces the recursive equation:

$$S(k, n, \boldsymbol{X}) = \overline{X}_m\ S(k, n-1, \boldsymbol{X}/X_m) \vee X_m\ S(k-1, n-1, \boldsymbol{X}/X_m), \quad 1 \le k \le n, \tag{14a}$$

while for the limiting cases k = 0 and k = (n+1), $S(k, n, \boldsymbol{X})$ is given non-recursively by

$$S(k, n, \boldsymbol{X}) = Sy(Z_{n+1}, \boldsymbol{X}) = 1, \quad k = 0, \quad n \ge 0, \tag{14b}$$

$$S(k, n, \boldsymbol{X}) = Sy(\varphi, \boldsymbol{X}) = 0, \quad k = (n+1), n \ge 0, \tag{14c}$$

Since the right hand side of Eq. (14a) is a disjoint sum of products of statistically independent expressions, it is a PRE that is readily convertible, on a one-to-one basis, into a probability expression. Hence, the following recursive relation for the reliability of a k-out-of-n:G system is obtained [3, 6]

$$\begin{aligned} R(k, n, \boldsymbol{p}) &= q_m * R(k, n-1, \boldsymbol{p}/p_m) \\ &\quad + p_m * R(k-1, n-1, \boldsymbol{p}/p_m), \quad 1 \le k \le n, \end{aligned} \tag{15a}$$

$$\begin{aligned} &= R(k, n-1, \boldsymbol{p}/p_m) \\ &\quad + p_m * (R(k-1, n-1, \boldsymbol{p}/p_m) - R(k, n-1, \boldsymbol{p}/p_m)), \\ &\quad 1 \le k \le n \end{aligned} \tag{15b}$$

The boundary conditions (14b) and (14c) also translate from the Boolean to the algebraic domain as

$$R(k, n, \boldsymbol{p}) = 1.0, \quad k = 0, \quad n \ge 0, \tag{15c}$$

$$R(k, n, \boldsymbol{p}) = 0.0, \quad k = (n+1), \quad n \ge 0. \tag{15d}$$

It is possible (though less intuitive) to obtain (15b) directly in the probability domain through the application of the pivoting (pivotal decomposition or factoring) [64] which is simply a version of the well-known total probability theorem [11]. The unreliability $U(k, n, \boldsymbol{p})$ satisfies recursive relations similar to (15a) or (15b) and boundary conditions complementary to those in (15c) and (15d).

B. An Iterative Algorithm Based on Binary Recursion

Based upon the recursive relation (15b) and boundary conditions (15c) and (15d) an efficient non-recursive algorithm for computing $R(k, n, \boldsymbol{p})$ and $U(k, n, \boldsymbol{p})$ has been reported by Rushdi [3, 4, 6]. This algorithm has a nice pictorial interpretation in terms of a SFG generalizing the one in Fig. 3. In fact, if we replace the graph transmissions p and q preceding column j by the subscripted symbols p_j and q_j, respectively, the node (i, j) represents the unreliability $U(i+1, j, \boldsymbol{p})$, *and* if we replace these two graph transmissions by the swapped subscripted symbols q_j and p_j, respectively, the node (i, j) represents the reliability $R(i+1, j, \boldsymbol{p})$.. The algorithm constructs an array of values inclusively bounded in the ij-plane by the four straight lines, i = 1, i = k, i = j, i = (j–n+k), which are the edges of a parallelogram with corners (i, j) at (1, 1), (k, k), (k, n) and (1,n–k+1). The algorithm has three versions depending on the order of traversing or sweeping the aforementioned parallelogram elements, namely:

1- The vertical-sweep version:
Nodes are visited column-wise, starting from the leftmost column (j = 1) and ending at the rightmost column (j = n). Within each column j, the bottom node (i = min(j, k)) is visited first, and then followed by upper nodes till the top node (i = max(1, j–n+k)) is reached.

2- The horizontal-sweep version:
Nodes are visited row-wise, starting from the topmost row (i = 1) and ending at the bottom row (i= k). Within each row i, the algorithm proceeds from left (j = i) to right (j = i+n–k).

3- The diagonal-sweep version:
Nodes are visited diagonal-wise, starting from the leftmost diagonal (j–i = 0) and ending at the rightmost one (j–i = n–k). Within each diagonal, the algorithm proceeds downwards from the top row (i = 1) to the bottom row (i = k).

Our efficient algorithm has a local memory requirement of (k+1) scalars. Its temporal complexity is measured by N = k(n–k+1) multiplications and 2N additions. In the worst case (k ≈ (n+1)/2), the algorithm has a linear spatial complexity of (n+3)/2 and a quadratic temporal complexity of $(n+1)^2/4$. There exist "*dual*" versions of the algorithm that compute the unreliability $U(k, n, \boldsymbol{p})$ instead of the reliability $R(k, n, \boldsymbol{p})$ with no change whatsoever in complexity [3, 6]. The algorithm can be shown to be correct, since when it is given a valid input it produces the right output in a finite amount of time, and also it passes the tests in [65]. To date, this algorithm has the least temporal complexity within the class of algorithms that basically use real multiplications and additions to compute R(k, n, **p**). It is believed [6] to be optimal in the worst case, in the sense that it is unlikely that there is an algorithm in the same class that performs fewer basic operations in the worst case.

The Rushdi algorithm in [3, 4, 6] is the basis for efficient algorithms that compute the reliabilities of more general systems such as the k-to-ℓ-out-of-n system [26, 27], the threshold system [30], the combined k-out-of-n:F, consecutive-k-out-of-n:F, and linear Connected-(r; s)-out-of-(m; n):F System [66], and the multi-state k-out-of-n system [29]. The algorithm was successfully applied in the analysis or design of some practical real-life systems such as furnace systems [49, 50], and static synchronous compensators (STATCOM) used in electric power systems [67]. It is also applicable in the analysis and design of fleets of aircrafts [68] and systems of pervasive computing [69].

The Rushdi algorithm in [3, 4, 6] and its extensions for the k-to-ℓ -out-of-n system [26, 27] and the threshold system [30] have been rediscovered repeatedly in the literature. The concept of a weighted k-out-of-n:G system in [12, 31] is a special case of that of a threshold system in [30]. The recursive relations and the algorithms in [12, 31] are also strongly similar to those in [30]. In another direction, Dutuit and Rauzy [10] paraphrased the Rushdi algorithm into an algorithm that they admitted is "*strongly similar*" to the Rushdi algorithm in [3, 4]. They also paraphrased the extension of the Rushdi algorithm for the k-to-ℓ -out-of-n system [26, 27], and believed that the resulting algorithm is new, obviously unaware of the work in [26, 27].

The above statements should never be understood to belittle the visionary insights in [10], which significantly enhanced the utility of switching (Boolean) algebra in reliability evaluations through the prudent use of the highly efficient and extremely popular ROBDD data structure [56, 57].

The Rushdi algorithm in [3, 4, 6] has an elegant technique of handling two-dimensional recursion, reminiscent of the use of Pascal's triangle in computing combinatorial (binomial) coefficients. This technique is very useful in tackling other problems of two- or multi-dimensional recursion, such as the computation of Stirling's numbers, and the computation of multinomial coefficients and probabilities.

VII. Complexity Issues

The k-out-of-n system has taken a considerably large share of the reliability literature. Virtually all the major techniques of system reliability analysis have been applied to k-out-of-n systems. The outcome is a potpourri of algorithms that have been surveyed in [6], where careful attention has been paid to ensure a uniform treatment of the various algorithms and to point out similarities, differences and interrelations among them. Notable among these algorithms is an algorithm due to Barlow and Heidtmann [2], which uses the coefficients in a *generating-function* expansion to express the probabilities of exactly *m* successes in *n* Bernoulli trials, and then employ an efficient technique to obtain their summation for $k \le m \le n$. This algorithm has a spatial complexity of (k+2) scalars, and a non-symmetric temporal complexity of ((k+1)*(n–k+1)–1) multiplications plus ((2k+1)*(n–k+1)–1) additions. In the worst case ($k \approx (n+1)/2$), these complexities reduce to (n+5)/2 memory cells and ((n+1)*(n+3)/4–1) multiplications, respectively. This means that the Rushdi algorithm [3, 4, 6] discussed herein has a slight trivial advantage over the Barlow-Heidtmann algorithm. Both algorithms are temporally $O(n^2/4)$. They are the best in temporal terms (among algorithms using real operations), in addition to being good space economizers. The similarity between the two algorithms is so strong that they are sometimes mistaken to be the same. To summarize the minor difference between the two algorithms, we note that the Rushdi algorithm is based on *explicit* recursion related to the *Cumulative Distribution Function* (*CDF*) of the generalized binomial distribution. The Barlow-Heidtmann algorithm, however, does not use recursion explicitly, but it has been shown by Rushdi [3, 6] to have *implicit* recursion, which turns out to be related to the *probability mass function (pmf)* of the same distribution [54, 55]. More detailed comparisons between these two algorithms are available in Rushdi [6], and in Kuo and Zuo [12].

In 1995, absolute optimality of the Rushdi algorithm and the Barlow-Heidtmann algorithm was lost to a new algorithm by Belfore [7] which has a temporal complexity $O(n(\log_2 n)^2)$. This algorithm combines the generating-function expansion concept [2, 70] with a recursive application of the *Fast Fourier Transform (FTT)*. The FTT uses complex arithmetic, and involves multiplications by *complex roots of unity*, which are equidistant points on the unit circle in the complex plane. Recursively, the FTT facilitates the computation of the *convolution* of two sequences, and hence the evaluation of the product of two generating functions. It is not desirable to apply the FFT for smaller problems since large overheads are involved [7]. The Belfore algorithm is faster than other algorithms for $n > 4000$ [7]. It is very hard to use (compared to the Rushdi algorithm) in manual computations for small-size systems, and hence does not provide a similar pedagogical insight.

Patel et al. [71] has a serious observation about numerical algorithms, which they emphasize by calling it a "folk theorem." This so called "theorem" states that "*If an algorithm is amenable to 'easy' hand calculation, it is probably a poor method if implemented in the finite floating-point arithmetic of a digital computer*." The "converse of the folk theorem" states that "*Many algorithms that are now considered fairly reliable in the context of finite arithmetic are not amenable to hand calculation*." Notable pertinent examples that fully support the "folk theorem" include formula (7) above for computing the reliability of a k-out-of-n:G system with equal-reliability components, and also formula (10) for computing its normalized life expectancy. Fortunately, the Rushdi algorithm for computing the reliability of a k-out-of-n:G system with non-identical components can serve as a "concrete counterexample" for a stronger version of the folk theorem that lacks the qualifying word "probably," and also for a stronger version of the converse theorem in which the qualifying word "Many" is omitted. The Rushdi algorithm is very nice for hand calculations, and still it is one of the fastest and most reliable and robust methods for digital computations.

VIII. Concluding Remarks

This paper presented a brief tutorial exposition of some recent developments in the evaluation of the reliability of partially-redundant (k-out-of-n) systems. A novel contribution of the paper is that it identified many practical examples in which such systems serve as a useful model. Some formulas for the reliability and life expectancy of these systems were discussed in the case of equal-reliability components. Certain celebrated formulas were shown to be numerically unstable and totally useless in the case of large systems with high-reliability components. The paper also reviewed how the Boole-Shannon expansion (or equivalently, the pivoting or factoring technique) is used to derive pertinent recursive relations, leading to a highly efficient algorithm for k-out-of-n reliability evaluation. This algorithm has a nice interpretation in terms of a regular Mason signal flow graph, which turns out to be (a) a reduced ordered binary decision diagram representing a monotone symmetric switching function, and (b) analogous to the minimal circuit realization of this function. In the worst case, the temporal and spatial complexities of this algorithm were shown to be quadratic and linear, respectively, in the number of system components. The paper listed some extensions and applications of this algorithm and compared it with a few related algorithms. In the following, a few additional remarks are added.

For the parallel system, which is totally or fully redundant, redundancy is always useful in the sense that using several components is always better than using a single component. This is not the case, however, when strict partial redundancy is used. Rushdi and Al-Hindi [46] utilized the Rushdi algorithm in [3, 6] in computing and tabulating values for the lower boundary of the *region of useful redundancy* for k-out-of-n systems, i.e., the region in which system reliability is better than that of a single component. If the components have constant failure rates, their lifetimes are exponentially distributed, and this lower boundary can be used to determine the point that divides the time axis into intervals of short and long missions, respectively [11]. In a different direction, Rushdi and Al-Thubaity [72] modified the Rushdi algorithm in [3, 6] to obtain efficient algorithms for computing the *first-order sensitivity* of k-out-of-n system reliability. These algorithms are useful in computing several criticality or *importance measures* and in evaluating the *instantaneous failure rate* of the system.

An important problem of interest is to study the characteristics of k-out-of-n systems under a set of assumptions other than the one employed throughout this paper. Examples of these include k-out-of-n systems with *statistically dependent components*, *common-cause failures*, *two failure modes*, *repair*, or *spares*. Typically, Markov-chain modeling is appropriate and effective in these cases [5, 11, 14, 73].

While simple *reliability-cost metrics* (such as reliability per cost or life expectancy per cost) can be used to guide the selection of a system from among several systems that are candidates for providing the same performance in a given mission, other more elaborate metrics (such as the cost elasticity of reliability or cost elasticity of life expectancy) have been developed [53] for partially-redundant systems. These metrics can be used to assess the cost-benefit aspect of adding redundancy to a system with the purpose of enhancing its reliability.

Since the tutorial exposition presented herein is somewhat brief, the interested reader is invited to pursue the topic further in the more elaborate surveys available in the literature in the reviews of Rushdi [6], Amari, Zuo, and Dill [14], Kuo and Zuo [12], Dutuit and Rauzy [10], and Chao, Fu, and Koutras [74].

REFERENCES

[1] Boland, P. J., and F. Proschan, The Reliability of k out of n Systems, *The Annals of Probability*, Vol. 11, No. 3, pp. 760-764, 1983.

[2] Barlow, R. E., and K. D. Heidtmann, Computing k-out-of-n System Reliability, *IEEE Transactions on Reliability*, Vol. R-33, No. 4, pp. 322-323, 1984.

[3] Rushdi, A. M., Utilization of Symmetric Switching Functions in the Computation of k-out-of-n System Reliability, *Microelectronics and Reliability*, Vol. 26, No. 5, pp. 973-987, 1986.

[4] Rushdi, A. M. Comment on : An Efficient Non-recursive Algorithm for Computing the Reliability of k-out-of-n Systems, *IEEE Transactions on Reliability*, Vol. 40, No. 1, pp. 60-61, 1991.

[5] Misra, K. B., *Reliability Analysis and Prediction*: *A Methodology Oriented Treatment*, Elsevier Science Publishers, Amsterdam, The Netherlands, 1992.

[6] Rushdi, A. M., *Reliability of k-out-of-n Systems*, Chapter 5 in Misra, K. B. (Editor), *New Trends in System Reliability Evaluation,* Vol. 16, *Fundamental Studies in Engineering,* Elsevier Science Publishers, Amsterdam, The Netherlands, pp. 185-227, 1993.

[7] Belfore, II, L. A., An $O(n(\log_2 n)^2)$ Algorithm for Computing the Reliability of k-out-of-n:G & k-to-1-out-of-n:G Systems, *IEEE Transactions on Reliability*, Vol. 44, No. 1, pp. 132-136, 1995.

[8] Barlow, R. E.., and F. Proschan, *Mathematical Theory of Reliability*, SIAM, New York, NY, USA, 1996.

[9] Ebeling, C. E., An Introduction to Reliability and Maintainability Engineering, McGraw-Hill, New York, NY, USA, 1997.

[10] Dutuit, Y. and A. Rauzy, New Insight into the Assessment of k-out-of-n and Related Systems, *Reliability Engineering and System Safety*, Vol. 72, No. 2, pp. 303-314, 2001.

[11] Trivedi, K. S. *Probability & Statistics with Reliability, Queuing, and Computer Science Applications,* 2nd Ed., Prentice-Hall, Englewood Cliffs, NJ, USA, 2002.

[12] Kuo, W., and M. J. Zuo, *The k-out-of-n System Model*, Chapter 7 in *Optimal Reliability Modelling: Principles and Applications,* Wiley, New York, NY, USA, pp. 231-280, 2003.

[13] Rausand, M., and A. Hoyland, *System Reliability Theory: Models, Statistical Methods, and Applications,* 2nd Ed., Wiley, Hoboken, NJ, USA, 2004.

[14] Amari, S. V., M. J. Zuo, and G. Dill, *O(kn) Algorithms for Analyzing Repairable and Non-repairable k-out-of-n:G Systems*, Chapter 21 in Misra, K. B. (Editor), *Handbook of Performability Engineering,* Springer, London, UK, pp. 309-320, 2008.

[15] Heidtmann, K. D., Bounds on Reliability of a Noncoherent System Using Its Length & Width, *IEEE Transactions on Reliability*, Vol. R-31, No. 5, pp. 424-427, 1982.

[16] Agrawal, A., and R. E. Barlow, A Survey of Network Reliability and Domination Theory, *Operations Research*, Vol. 32, No. 3, pp. 478-492, 1984.

[17] Barbara, D. and H. Garcia-Molina, The Reliability of Voting Mechanisms, *IEEE Transactions on Computers,* Vol. C-36, No. 10, pp. 1197-1208, 1987.

[18] Kumar, A., Hierarchical Quorum Consensus: A New Algorithm for Managing Replicated Data, *IEEE Transactions on Computers*, Vol. 40, No. 9, pp. 996-1004, 1991.

[19] Brown, F. M., *Boolean Reasoning: The Logic of Boolean Equations*, Kluwer Academic Publishers, Boston, MA, USA, 1990 (2nd Ed. by Dover Publications, Mineola, NY, USA, 2003).

[20] Wheeler, R. F., *Rethinking Mathematical Concepts*, Ellis Horwood, Chichester, West Sussex, England, 1981.

[21] Bergman, B., On Reliability Theory and Its Applications (with Discussion), *Scandinavian Journal of Statistics*, Vol. 12, No.1, pp. 1-41, 1985.

[22] Kaufmann, A., D. Grouchko, and R. Cruon, *Mathematical Methods for the Study of the Reliability of Systems*, Academic Press, New York, NY, USA, 1977.

[23] Barlow, R. E., and S. Iyer, Computational Complexity of Systems and the Reliability Polynomial, *Probability in Engineering & Information Science*, Vol. 2, pp. 461-469, 1988.

[24] Billinton, R. and Allan, R. N. *Reliability Evaluation of Engineering Systems: Concepts and Techniques,* 2nd Ed., Springer, New York, NY, USA, 2005.

[25] Papastavridis, S. G., and M. E. Sfakianakis, Optimal-Arrangement & Importance of the Components in a Consecutive-k-out-of-r-from-n:F system, *IEEE Transactions on Reliability*, Vol. 40, No. 3, pp. 277-279, 1991.

[26] Rushdi, A. M., Efficient Computation of k-to-ℓ-out-of-n system reliability, *Reliability Engineering*, Vol. 17, pp. 157 -163, 1987, Erratum : *ibid*, vol. 19, p. 321, 1987.

[27] Rushdi, A. M., and F. M.-A. Dehlawi, Optimal Computation of k-to-ℓ-out-of-n System Reliability, *Microelectronics and Reliability*, Vol. 27, pp. 875-896, 1987, Erratum : *ibid*, Vol. 28, p. 671, 1988.

[28] Upadhyaya, S. J., and H. Pham, Analysis of Noncoherent Systems and an Architecture for the Computation of the System Reliability, *IEEE Transactions on Computers***,** Vol. 42, No. 4, pp 484-493, 1993.

[29] Tian, Z., M. J. Zuo, and R. C. M. Yam, Multi-State k-out-of-n Systems and Their Performance Evaluation, *IIE Transactions*, Vol. 41, No. 1, pp. 32-44, 2009.

[30] Rushdi, A. M., Threshold Systems and Their Reliability, *Microelectronics and Reliability*, vol. 30, No. 2, pp. 299-312, 1990.
[31] Wu, J.-S., and R.-J. Chen, An Algorithm for Computing the Reliability of Weighted-k-out-of-n Systems, *IEEE Transactions on Reliability*, Vol. 43, No. 2, pp. 327-328, 1994.
[32] Rushdi, A. M., A Switching-Algebraic Analysis of Consecutive-k-out-of-n:F systems, *Microelectronics and Reliability*, Vol. 27, pp. 171-174, 1987.
[33] Rushdi, A. M., A Switching-Algebraic Analysis of Circular Consecutive-k-out-of-n:F Systems, *Reliability Engineering and System Safety*, Vol. 21, , pp. 119-127, 1988.
[34] Chang G. J., L. Cui, and F. K. Hwang, Reliabilities for (n, f, k) Systems, *Statistics & Probability Letters*, Vol. 43, pp.237-242, 1999.
[35] Tung, S.S., Combinatorial Analysis in Determining Reliability, *Proceedings of the Annual Reliability and Maintainability Symposium*, pp. 262-266, 1982.
[36] Eryilmaz, S., and T. Aksoy, Reliability of Linear (n, f, k) Systems with Weighted Components, *Journal of Systems Science and Systems Engineering*, Vol. 19, No. 3, pp.277-284, 2010.
[37] Bollinger, R. C., Strict Consecutive-k-out-of-n:F Systems, *IEEE Transactions on Reliability*, Vol. R-34, No. 1, pp. 50-52, 1985.
[38] Rushdi, A. M., Effect of Statistical Dependencies in Strict Consecutive-k-out-of-n:F Systems, *Microelectronics and Reliability*, Vol. 28, No. 2, pp. 309-318, 1988.
[39] Rushdi, A. M., A Conditional Probability Treatment of Strict Consecutive-k-out-of-n:F Systems *Microelectronics and Reliability*, Vol. 29, No. 3, pp. 581-586, 1989.
[40] Rushdi, A. M., S. G. Papastavridis., R. A. Evans, Some Open Questions (and Replies) on: Strict Consecutive-k-out-of-n:F Systems, *IEEE Transactions on Reliability*, Vol. 39, No. 3, pp. 380-381, 1990.
[41] Hwang, F. K., Comment on Strict Consecutive-k-out-of-n:F Systems, *IEEE Transactions on Reliability*, Vol. 40, No. 3, pp. 264, 270, 1991.
[42] Henley, E. J., and H. Kumamoto, *Designing for Reliability and Safety Control*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1985.
[43] Parhami, B., Voting Networks, *IEEE Transactions on Reliability*, Vol. 40, No. 3, pp. 380-394, 1991.
[44] Lin, S., and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications,* 2nd Ed., Pearson Prentice-Hall, Upper Saddle River, NJ, USA, 2004.
[45] Hwang, K., and T.-P. Chang, Combinatorial Reliability Analysis of Multiprocessor Computers, *IEEE Transactions on Reliability*, Vol. R-31, No. 5, pp. 469-473, 1982.
[46] Rushdi, A. M., and K. A. Al-Hindi, A Table for the Lower Boundary of the Region of Useful Redundancy for k-out-of-n Systems, *Microelectronics and Reliability*, Vol. 33, pp. 979-992, 1993.
[47] Rai, S., A. K. Sarje, E. V. Prasad, and A. Kumar, Two Recursive Algorithms for Computing the Reliability of k-out-of-n Systems, *IEEE Transactions on Reliability*, Vol. R-36, No. 2, pp. 261-265, 1987.
[48] Parker, D. S., and C. S. Raghavendra, The Gamma Network, *IEEE Transactions On Computers.*, Vol. C-33, No. 4, pp. 367-372, 1984.
[49] Zuo, M., S. Chiovellib, and J. Huang, Reliability Evaluation of Furnace Systems, *Reliability Engineering and System Safety*, Vol. 65, pp. 283–287, 1999.
[50] Zuo, M.-J., and Y. Wu, Reliability Evaluation of a Furnace System Using the k-out-of-n and the consecutive k-out-of-n Reliability Models, *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics,*, Vol. 4, pp. 3119-3123, 1996.
[51] Shamir, A., How to Share a Secret, *Communications of the Association of Computing Machinery*, Vol. 22, pp. 612–613, 1979.
[52] Smotherman, M., R. M. Geist and K. S. Trivedi, Provably Conservative Approximations to Complex Reliability Models, *IEEE Transactions On Computers*, Vol. C-35, No. 4, pp. 333-338, 1986.
[53] Rushdi, A. M., and A. E. Alsolami, Cost Elasticities of Reliability and MTTF for k-out-of-n Systems, *Journal of Mathematics and Statistics*, Vol. 3, No. 3, pp. 122-128, 2007.
[54] Rushdi, A. M. and Al-Qasimi, A. M., Efficient Computation of the PMF and the CDF of the Generalized Binomial Distribution, *Microelectronics and Reliability*, Vol. 34, pp. 1489-1499, 1994.
[55] Al-Qasimi, A. M., and A. M. Rushdi, A Tutorial on How to Efficiently Calculate and Format Tables of the Binomial Distribution, *Journal of King Abdul-Aziz University: Engineering Sciences,* Vol. 19, No. 1, pp. 3-17, 2008.
[56] Bryant, R., Graph-Based Algorithms for Boolean Function Manipulation, *IEEE Transactions on Computers*, Vol. 35, No. 8, pp. 677-691, 1986.
[57] Rauzy, A., *Binary Decision Diagrams for Reliability Studies*, Chapter 25 in Misra, K. B. (Editor), *Handbook of Performability Engineering,* Springer, London, UK, pp. 381-396, 2008.
[58] Unger, S. H., *The Essence of Logic Circuits*, 2nd Ed., IEEE Press, New York, NY, USA, 1997.
[59] Heidtmann, K. D., Improved Method of Inclusion-Exclusion Applied to k-out-of-n Systems, *IEEE Transactions on Reliability*, Vol. R-31, No. 1, pp. 36-40, 1982.
[60] Arulmozhi, G., Direct Method for Reliability Computation of k-out-of-n Systems, *Applied Mathematics and Computation*, Vol. 143, pp. 421-429, 2003.
[61] Rushdi, A. M., and A. A. Abdul-Ghani, A Comparison Between Reliability Analyses Based Primarily on Disjointness or Statistical Independence, *Microelectronics and Reliability*, Vol. 33, pp. 965-978, 1993.
[62] Rushdi, A. M., and O. M. Ba-Rukab, A Doubly-Stochastic Fault-Tree Assessment of the Probabilities of Security Breaches in Computer Systems, *Proceedings of the Second Saudi Science Conference, Part Four: Computer, Mathematics, and Statistics,* Jeddah, Saudi Arabia, pp. 1-17, 2004.
[63] Rushdi, A. M., and O. M. Ba-Rukab, Fault-Tree Modeling of Computer System Security, *International Journal of Computer Mathematics,* Vol. 82, No. 7, pp. 805-819, 2005.
[64] Page, L. B., and J. E. Perry, Reliability of Directed Networks Using the Factoring Theorem, *IEEE Transactions on Reliability*, Vol. R-38, pp. 556-562, 1989.
[65] Rushdi, A. M., How to Hand-Check a Symbolic Reliability Expression, *IEEE Transactions on Reliability*, Vol. R-32, No. 5, pp. 402-408, 1983.
[66] Zuo, M. J., D. Lin, and Y. Wu, Reliability Evaluation of Combined k-out-of-n:F, Consecutive-k-out-of-n:F, and Linear Connected-(r; s)-out-of-(m; n):F System Structures, *IEEE Transactions on Reliability*, Vol. 49, No. 1, pp. 99-104. 2000.
[67] Lu, Z., and W. Liu, Reliability Evaluation of STATCOM Based on the k-out-of-n: G Model, *2006 International Conference on Power System Technology* (*PowerCon 2006*), pp. 1-6, 2006.
[68] Cochran, J. K., and T. P. Lewis, Computing Small-Fleet Aircraft Availabilities Including Redundancy and Spares, *Computers & Operations Research*, Vol. 29, pp. 529-540, 2002.
[69] Hansen, K. M., and J. Brønsted, *Modeling Service Composition Reliability in Pervasive Computing*, Science Institute, University of Iceland, Reykjavík, Iceland, Technical report RH-02-2010, 2010.
[70] Levitin, G., *Universal Generating Function and Its Applications*. Springer-Verlag, Berlin, Germany, 2005.
[71] Patel, R. V., A. J. Laub, and P. M. Van Doreen. (Editors), *Numerical Linear Algebra Techniques for Systems and Control*, IEEE Press, New York, NY, USA, 1994.
[72] Rushdi, A. M. and A. O. Al-Thubaity, Efficient Computation of the Sensitivity of k-out-of-n System Reliability, *Microelectronics and Reliability*, Vol. 33, pp. 1963-1979, 1993.
[73] Allen, A. O. *Probability, Statistics, and Queueing Theory with Computer Science Applications,* 2nd Ed., Academic Press, San Diego, CA, USA, 1990.
[74] Chao, M.-T., J. C. Fu, and M. V. Koutras, Survey of Reliability Studies of Consecutive-k-out-of-n:F Systems & Related Systems, *IEEE Transactions on Reliability*, Vol. 44, No. 1, pp. 120-127, 1995.

Ali Muhammad Ali Rushdi was born in Port Said, Egypt, on May 24, 1951. He received the B.Sc. degree (Honors) in Electrical Engineering (Electronics and Communications) from Cairo University, Cairo, Egypt, in 1974, and the M.S. and Ph.D. degrees in Electrical Engineering from the University of Illinois at Urbana-Champaign, USA in 1977 and 1980, respectively. He maintained a perfect GPA of 5.0/5.0 throughout his study.

In 1974 he was appointed Demonstrator and Instructor in the Department of Electronics and Electrical Communications Engineering of Cairo University. From 1976 to 1980 he was Research Assistant in the Electrical Engineering Department of the University of Illinois at Urbana-Champaign. Since 1980 he has been with King Abdul-Aziz University in Jeddah, Saudi Arabia, where he is now Professor of Electrical and Computer Engineering as well as Head and Coordinator of the Computer Engineering Group. At King Abdul-Aziz

University he has structured and taught a variety of graduate and undergraduate courses, supervised master theses and senior projects, and contributed significantly to accreditation activities. He served as a member of the Editorial Board of the *IEEE Transactions on Instrumentation and Measurements* (1986-1994), and was a frequent reviewer for the *IEEE Transactions on Reliability* (1983-1998). He is currently a member of the Editorial Board of the *Journal of King Abdul-Aziz University: Engineering Sciences*, and is an Associate Editor of Reliability and Computer Engineering for the *International Magazine on Advances in Computer Science and Telecommunications (IMACST)*. His Research Interests and publications over the past four decades spanned the areas of Electromagnetic Communications Engineering, Computer Engineering, Reliability Engineering, Digital Design, Engineering Education, Neural and Switching Networks, Advanced Mathematics, Boolean Algebra and Logic, Engineering Design, Inferential Thinking, and Problem Solving.

Prof. Rushdi is an initiated member of the Honorary Societies: Eta Kappa Nu and Phi kappa Phi. He is a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE).

Cellular Network Faults and Services Dependency Modeling

Okuthe P. Kogeda and Johnson I. Agbinya

Abstract— **Cellular network services depend on one or more resources and a resource can be used by one or more services. Network faults can affect these resources and hinder them from supporting services that depend on them. It is therefore necessary to react accurately to faults occurring in one or more components that provide such resources to ensure high quality of service provision. This can be achieved by determining the dependencies between different services, dependencies between services and resources and dependencies on the resource level. In this paper, we present the basics, types, benefits, life cycle of service dependency. We also present service dependency models in a dynamically changing environment like cellular networks. The simulation results showing 99.95% dependency are presented in this paper**.

***Index Terms*— Network services, Operations Support System, Network service dependency, Network faults, Dependency types, Simulation, Dependency dynamics, Dependency life cycle, Dependency binding, Availability.**

I. Introduction

Internet explosion, increasing number of services on offer and subscribers has put a lot of pressure on the cellular network service providers. The networks are loaded with different kinds of subscribers having different tastes and needs. This requires that the operation of the network be at its best all the times not only to keep the subscribers happy but also to retain them and attract new ones. This can be achieved with proper maintenance of the network itself.

A. Network Services

Network service is a crucial and very important resource in a cellular network system. For any cellular network service provider to retain and acquire new subscribers, services on offer must be general, cost effective, fair robust, reliable and have high performance connectivity among a large number of communication devices (i.e., computers, wireless terminals, etc), for the highest customer satisfaction.

Manuscript received September 1, 2010. This work was supported in part by the National Research Foundation (NRF), South Africa.

Okuthe P. Kogeda is currently working in Computer Science Department as Senior Lecturer, University of Fort Hare, *Private Bag X1314, Alice 5700, Republic of South Africa. (Email : pkogeda@ufh.ac.za)*

Johnson I. Agbinya, is working at Center for Real-time Information Networks (CRIN), Faculty of Engineering, University of Technology, Sydney, NSW 2007, Australia. (Email: agbinya@eng.uts.edu.au).

B. Definition of Services and Applications

The ITU-T defines telecommunication services as follows [1]: A *service* represents telecommunication capabilities that the customer buys or leases from a service provider. A service is an abstraction of the network-element-oriented or equipment-oriented view. Identical services can be provided by different network elements, and different services can be provided by the same network element. A set of functions and facilities offered to a user by a provider is known as service. One service can serve several consumers and a server is always its execution environment.

An application on the other hand is used as the generic term that represents a set of features, combining communication and document processing, on which end users may perform operations. Applications may depend on working methods, and on allowed processing of documents. Open interchange of process-able documents and co-operative working are examples of applications [1]. An *application* is a program that a user directly interacts with. An application utilizes services and might incorporate modules to fulfill its tasks. The application is not restricted to a special environment to run in.

C. Characteristics of Cellular Network Services

Cellular network services are tricky entities that do not have specific characteristics that apply to all of them. Different cellular network services possess different features that preserve the identity of each cellular network service. However, there are a few features that most cellular network services have in common. These include: services can use one or more media of transmission; most services being offered by cellular service providers are easily programmable and flexible to the needs of customers; most services are easily accessible with cost and legal permission to use them; cellular network services are randomly initiated and executed.

D. Classification of Network Services

A set of applications with similar or common set of characteristics (cf. Section I C) can be classified as a service. Generally cellular network services can be classified into data, voice and multimedia according to ITU-T I.211 [1]. Network services in digital form are called data. Network services in 'vocal chord' form are called voice and are regarded as the oldest cluster of network services. However, the latest type of network services, which are normally composed of pictures, videos, text and/or sound are called multimedia. Multimedia services consume a lot of bandwidth and require powerful devices to process (receive and send) them. Fig. 1 shows classes of network services with examples.

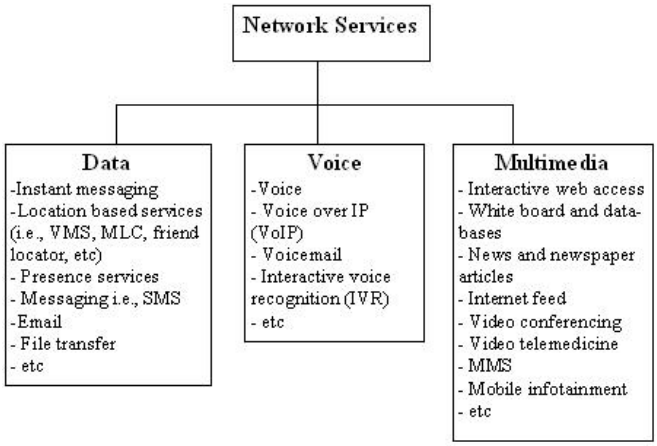


Fig. 1. Classes of Cellular Network Services

E. Service Life-cycle

Network services go through various defined stages, which start with planning, signing (analyse) before one can start the provision of the service, which may encounter errors during this period and such errors must be corrected. The service can be maintained till it requires a major improvement. Then the process starts again from plan, sign, provision and maintenance, hence network service life cycle as shown in Fig. 2 below.

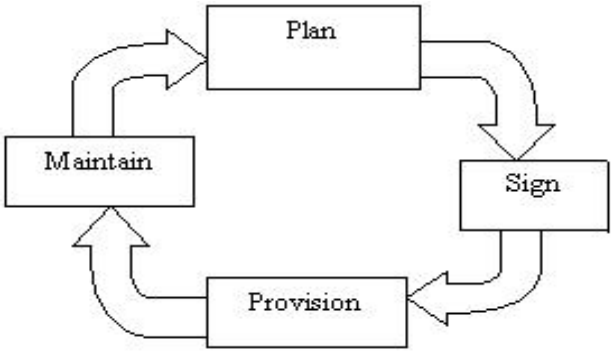


Fig. 2. Cellular Network Service Life-cycle

The stages of service life cycle are:

Plan - This is the first stage and this is where the target sites are identified in market penetration plan with the actual versus planned implementation success being measured. Then the results are reported, which can be according to region, city or sales representatives.

Sign - The rent rates are analyzed by geographic region to establish pricing benchmarks before negotiating site arrangements with landlords and property managers. Then the agreement is signed with track of all information about leases, buildings, facilities and contacts.

Provision - Network services are built and implemented into signed buildings. Construction can be managed and status tracked using building, service and/or geographical location. Then offer new services to increase sales and profitability.

Maintain - Property managers are notified of lease, insurance expiry dates, up to date contact information, site documentation, Service Level Agreement (SLA) details and payment schedule. Then ensure that emergency contacts are up-to-date and readily available to Operations Support System (OSS) group.

F. Network Resource

"A network resource is any physical or virtual component of limited availability within a networked computer system. Every device connected to a computer system is a resource. Every internal system component is a resource. Virtual system resources include files, network connections and memory areas." [2]

Network resources can also be referred to as various parts of the network (hardware and software) which support each other by combining or individually to provide specific functions within the network environment. Network service is created by these functions.

Network resources are network elements that support services. A network resource can be basic or elementary. *A basic* network resource is the smallest element that supports a network service. It is scalar in nature and cannot be split further down. It supports the service with all its parts as a whole. A combination of two or more basic network resources to offer a function to a service is called *elementary* network resource. An elementary network resource cannot be used without any of the basic element parts.

This paper is organised as follows: In Section II, we give a brief overview of Cellular network service dependency and related work. We present types of network service dependencies in Section III. In Section IV, benefits of network service dependency modelling is discussed. Dynamics and network dependency life-cycle is presented in Section V. In Section VI, cellular network service dependency models are presented. In Section VII, simulation results are provided and then we draw conclusion in the subsequent section.

II. Cellular Network Service Dependency

Dependency provides a very interesting relationship between two or more components or services in cellular network environment. The consumer/provider relationship between different entities in a cellular network system is called dependency. When one component requires a service performed by another component in order for it to execute its functions, this relationship between the two components is called a dependency. For example, a voice service depends on a cell where it is located for network signal and the database for billing purposes before a call is initiated. In turn these services depend on availability of power supply. In this case a voice service is the *dependent* and cell and database are the *antecedent*. Consequently, cell and database are the *dependents* and power is the *antecedent*. This relationship is shown in Fig. 3. Services cannot be considered isolated tasks. Services largely depend on other services or sub-services,

lower level network elements, operating systems, physical components and communication infrastructure to be able to function.

Network service dependency has attracted several researchers with different viewpoints. Ensel [3] presents a scalable service dependency. The author used neural networks for dependency detection modeling. Cervantes et al [4] present a mechanism to automate service dependency management in a service-oriented component model. The mechanism they used eliminates complex and error-prone code from component-based applications dynamically.

Gruschke [5] proposed dependency graph for event correlation where dependency was described as a relationship between different entities. A generic approach proposed here could be used for different abstraction levels (i.e., system level, network level, service level). Caswell et al [6] describe dependencies for services with specific reference to Internet Service Providers. They went further by defining five types of dependencies (c.f. Section III). Gupta et al [7] present analysis of temporal relationships of interactions to derive dependencies.

Natu et al [8] proposed a fault diagnosis architecture and algorithm that capture the dynamically changing dependencies in mobile ad-hoc networks. They introduced a temporal correlation algorithm that performs fault diagnosis with dynamically changing dependency information. Bahl et al [9] present localization of the sources of performance problems in large enterprise networks using Inference Graph model. They developed Sherlock system to discover Inference Graphs in the operational enterprise, infer critical attributes, and then leverage the result to automatically detect and localize problems.

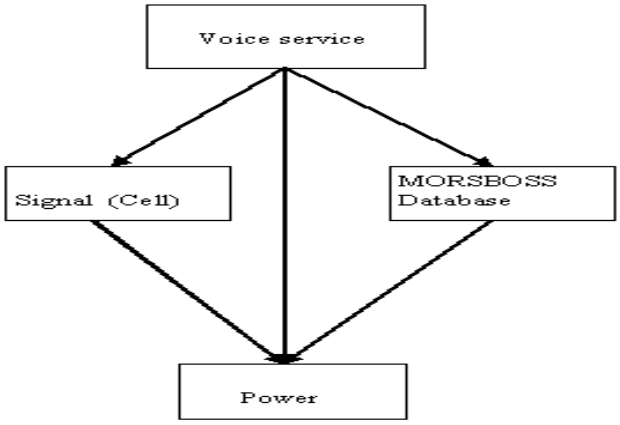


Fig. 3. Examples of (Voice) Service Dependency

However, the approaches presented by various authors as discussed above may not predict which service would be affected by a fault in dynamically changing networks. In this work, an approach of dynamic dependency is used to pre-empt the likely services that a likely fault would affect at a particular time. While some of the approaches above cannot be implemented, the approach adopted in this work can be implemented backed with mathematical support.

III. TYPES OF SERVICE DEPENDENCY

The main types of service dependency include [3], [6], [10]:

Execution dependency – This dependency relates directly to an application server process being executed on a host machine. The performance of an application server process depends on the status of the host machine. The types of application servers that are executed on host machines include web, email, news, DNS, and NFS.

Link dependency – performance of a service depends on the link status. For example, communication between two nodes, *A* and *B*, may solely depend on the link between them *AB*.

Component dependency – in case of a web service that is provided on different front-end servers, which are selected by a round-robin DNS scheduling of performance, depends on the currently selected server. A component dependency occurs in order to ensure scalability and redundancy of a service. ISPs often replicate web, email, and news content across a number of servers. The round robin scheduling balances the load among the servers. The servers are grouped together and assigned a single domain name in the DNS database. When the DNS server receives a request for the domain name, the IP address of one of the servers is acquired in the round-robin scheme.

Inter-service dependency – It occurs when one service accesses another service for its proper operation. This occurs between services, i.e., e-mail service depends on an authentication service and on an NFS service; a web service depends on DNS service to allow the subscriber to connect the web server host using its IP address, and an NFS service is used to access the web content.

Organizational dependency – This dependency occurs when there are different ISP operations personnel (e.g., experts) who are responsible for different services and service components. For example, an ISP may have a first supervisor managing the web service, a second supervisor managing DNS, and a third supervisor managing NFS. Operational responsibilities may also be delegated based upon the geographical location of the service components.

The first three dependencies are grouped and referred to as *resource dependency*. In this case the service being offered depends on the resources (i.e., execution, link, component, and/or another service) available at the time. These resources in turn are affected by the cellular network faults, i.e., faults may degrade, reduce or totally take away the resources available to a service.

IV. BENEFITS OF SERVICE DEPENDENCY

The main benefits of dependency modeling include [3]:

Root cause analysis – it helps to find a common (root) cause of faults detected at different places within the cellular network environment. This can be used on network components reporting error conditions as well as to services, where end users detect problems. The faults that are normally reported to the management systems are descriptions of the

symptoms. Therefore further knowledge about dependencies among the faults is necessary to derive their root cause.

Determination of availability requirements on services. To minimize the time for resolving network faults.

Prediction of the impacts on other services due to management operations. This is of particular interest when a resource goes down (i.e., due to planned downtime for repairs) then it can be determined in real-time which services and customers would be affected.

It can be the basis of scheduling tasks and transactions. The service dependency provides a detailed task structures, which enables better coordination of services. It can be used to recognize service misuse and for intrusion detection.

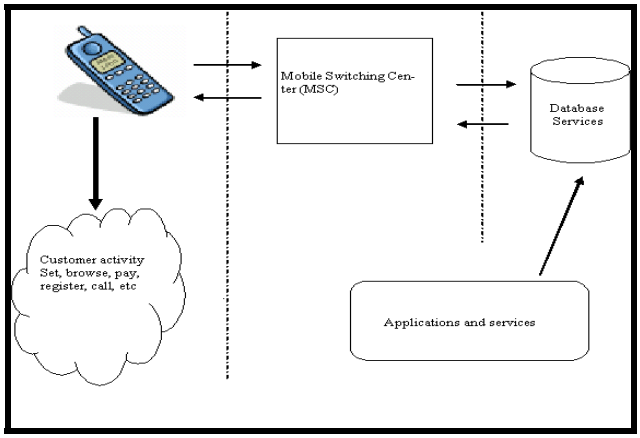


Fig. 4. Model of Network Environment Showing dependency

In order to drive the problem solving process, that is, cellular network faults prediction, a model of cellular network faults, or a concept of services and dependencies between them is required. A cellular network service can be defined as a set of functionalities, which are offered by a cellular network service provider to a customer at a customer provider interface (i.e., mobile handset) with an agreed quality of service. A service can depend on one or more resources and a resource can be used by one or more services. To ensure that high quality of services is provided, it is necessary to react accurately to faults occurring in one or more components that provide such resources. This can be achieved by determining the dependencies between different services, dependencies between services and resources and dependencies on the resource level. It will be better also to bring to clarity what an interface and a component stand for. *A component* is a non-trivial, nearly independent, and replaceable part of a cellular network system that fulfils a clear function in the context of a well-defined architecture. A component conforms to and provides the physical realization of a set of interfaces. An *interface* is a collection of operations that are used to specify a service of a component. It focuses upon the behaviour, not the structure of a given service. Fig. 4 shows a model of cellular network environment showing dependency.

V. DYNAMICS AND DEPENDENCY LIFE-CYCLE

A. Cellular Network Service Dependency Dynamics

Cellular network service dependency changes as variables within the cellular network setup changes. The changes are normally caused by resources (network components, services, power, etc) becoming unavailable due to network faults, resources may migrate, or may be upgraded. In a cellular network, the components and/or managed objects that represent the resources may be many. The change of dependency that may occur as a result of fault in a cellular network is termed as *cellular network dependency dynamics*.

Cellular network services can be modeled as node, communication and precedence constraints between services as directed edge and the model can be expressed as a Directed Acyclic Graph (DAG). Let the service be *S*, $S = \{s_1, s_2, s_3, ..., s_N\}$. A complex cellular network system may offer *N* number of services. A service depends on resource(s) *R*, where $R = \{r_1, r_2, r_3, ..., r_j\}$. A resource can be network link, component, or other services. An edge between two services, S_a and S_b is given by S_{ab}, which expresses the dependent relation between S_a and S_b. Given service S_N the set of parent services is denoted as $pred(S_N)$, and the set of children services is denoted as $succ(S_N)$. A service S_N is called entry service if $|pred(S_N)| = 0$ and an exit service if $|succ(S_N)| = 0$. Therefore *N* services depend on *R* heterogeneous resources. It is essential to map the set of *N* services in the DAG into *R* heterogeneous available resources in order to avoid the faulty components supporting the resources required by the services.

In explaining the cellular network dependency dynamics, the dependency relation between S_a and S_b may change, for example, if S_b malfunctions then S_a (will also fail in normal circumstances) but in this case S_a may use another service say S_k, which offers the same resources for its operation. Also if S_a depends on a particular route (link) R_l then with the failure of R_l, S_a is also expected to fail. But this may not be the case because S_a may use another route to complete the execution.

The system implementation takes into consideration this dependency dynamics with the cellular network system. It therefore means that the dependency models presented in this paper are dynamic in nature and robust. For a system to fail, it means all the alternative dependencies are exhausted. The main causes of dependency dynamics include: cellular network faults, which may cause the cellular network resources to appear and disappear during the system lifetime; deployment of new sub-systems; change of resource availability; re-negotiation of new service level agreements, etc.

However, it is worth noting that most of the dependencies are fairly permanent and only change when there is deemed fault with one of the main antecedent. This is the main interest in studying how cellular network faults may change the dependency and its subsequent effects on the reliability of the cellular network services.

B. Cellular Network Dependency Binding

In exploring network dependency binding, we define the three main variables used. These variables are:

- faults, *F* where $F \rightarrow \{f_1, f_2, f_3, ..., f_i\}$,
- resources, *R*, where $R \rightarrow \{r_1, r_2, r_3, ..., r_j\}$, and
- services, *S*, where $S \rightarrow \{s_1, s_2, s_3, ..., s_N\}$.

The variables relate to each other depicting the relationship between them, which can be one-to-one, one-to-many and many-to-many. The dependency only exists when cardinality between dependent and antecedent is exactly one. The maximum cardinality between the objects is infinity. The relationship between the variables can take either of the following two sets:

(a) The network faults relates to resources directly as follows:
  (i) A set of faults can affect a set of resources in the network, $F \rightarrow R$
  (ii) A set of faults can affect one or a particular resource in the network, $F \rightarrow r_j$
  (iii) One or a particular fault can affect a set of resources in the network, $f_i \rightarrow R$
  (iv) One or a particular fault can affect one or a particular resource in the network, $f_i \rightarrow r_j$

(b) The network services depend on network resources directly as follows:
  (i) A set of services depend on a set of resources in the network, $R \leftarrow S$
  (ii) A set of services depend on one or a particular resource in the network, $r_j \leftarrow S$
  (iii) One or a particular service depends on a set of resources in the network, $R \leftarrow s_N$
  (iv) One or a particular service depends on one or a particular resource in the network, $r_j \leftarrow s_N$

A binding which can be static or dynamic would occur with the knowledge of cardinality. A *static binding* is where the dependency bindings cannot change at run time and the dependent service is guaranteed to be present the entire time the resource is available, whereas *dynamic binding* is where the dependency bindings can change at run time and service availability cannot be guaranteed. Network services would be affected differently by different types of bindings as summarized in Table I:

Table I: Different Types of Dependency Binding

Binding Type	**Semantics of the dependency type**
One-to-One, static	A service is bound to one resource, any change invalidates the service.
One-to-One, dynamic	A service is bound to one resource; changes do not invalidate the service as long as it can be bound to another resource.
One-to-Many, static	A service is bound to at least one resource, any change invalidates the service.
One-to-Many, dynamic	A service is bound to at least one resource; changes do not invalidate the service as long as the binding count is not zero.
Many-to-Many, static	A set of services are bound to a set of available resources at the time of binding, changes invalidates the services.
Many-to-Many, dynamic	A set of services are bound to a set of all available resources at the time of binding, as resources become available/unavailable they are bound/unbound to/from the services, the services never becomes invalid.

C. Cellular Network Service Dependency Life-Cycle

Cellular network environment is very dynamic in nature and so the dependency evolving through five phases, referred to as *dependency lifecycle*. The phases include:

1) **Initiate**: This is the initial phase where the dependency is initiated when service consumption is signaled. The initial parameter values are received at this stage. For example, when you initiate a call, first the signal is acquired to have connection to the MSC, and then database connection is initiated to establish whether you have enough units to continue with the service consumption.
2) **Acquire**: existing services, resources, and common (known) faults are acquired by the dependency for mapping purposes at this stage. New and old dependencies are ascertained mainly for consumption purposes, i.e., after service, S_a signaled service, S_k for dependency and received positive answer, then it acquires the resources in readiness for the dependency mapping to complete the service consumption through S_{ak}.
3) **Start Map**: This stage triggers the start of dependency mapping.

4) **Map**: new resources may be added to the dependency pool during this phase. Existing resources and dependency parameters may be removed or updated to ensure the dependency dynamics are maintained. Dependency mapping can be affected by these changes, and so must be resolved continuously for a robust cellular network system.
5) **Stop**: The dependency is terminated at this stage.

The dependency life cycle continues by initiating another dependency. The semantics of the dependency are implemented in the system, which correlates the services to faults. The service dependency life cycle is shown in Fig. 5.

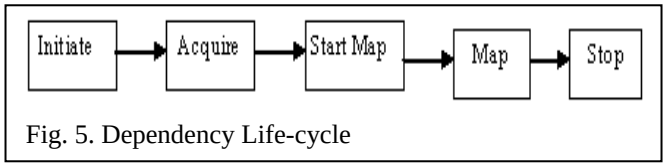


Fig. 5. Dependency Life-cycle

VI. CELLULAR NETWORK DEPENDENCY MODELS

A cellular network environment can be logically modeled as layers of resources (i.e., services, applications and other software and hardware components) that cooperate to deliver an end-to-end service. Services or components in one layer depend on functions provided by components in a lower-supporting layer. Failures occurring in one layer affect the functioning of dependent components in another layer. The dynamics of service dependency are considered for cellular network faults prediction purposes, because significant changes in the overall system behaviour are detected through emerging or disappearing dependencies. In addition to the discussion in Section I *F* above, network resources are important components in service dependency modeling.

Network elements (resources) support services through Service Access Points (SAP) and port accesses. The services dependency is modeled keeping in mind the current changes happening to the network environment to proactively detect faults before they impact end users. A ontology is implemented in the system which facilitates the mapping of faults and services to list faults that are likely to cause services failure. Table II below summarizes the list of faults, resources and services correspondingly.

A. Network service availability models

One of the main aims of this work is to develop a reliable service based Operations Support System (OSS) where services would be available to users whenever they want to consume them. Availability of network services depends on availability of network resources to support them to carry out their functions. For example, an end-to-end service availability would depend on availability of service source, network, link, and availability of the destination device. Network service availability is a combined availability of the network parts (elements) supporting the service(s).

Table II: Summary of faults, resources, and services correspondingly

Faults	**Resources**	**Common Services**
Multiplexer	Link (lines), electrical power, Multiplexer adaptor, External Bus Interface (EBI) cable, conversion kit, port module, etc	Voice, VoIP, Videoconferencing, etc
Power	Generator, Electrical power sources, Electrical switches, transmission lines, etc	SMS, MMS, Voice, VoIP, Email, etc. Virtually all services are affected.
Transmission	Link, cables, multiplexer, network name resolution, ISDN switches, ISDN lines, Gateways, etc	Affect real-time services Affects services in a serial connection May delay services such as SMSs, etc
Cell	Link, Electrical power, cables, multiplexer, etc	VoIP, SMS, MMS, Email, Internet, Video conferencing, etc
Time Out	Link, RAM, multiplexer, etc	SMS, MMS, Email, VoIP, Video conferencing, etc
Run Time Error	RAM, Link, Switches, etc	VoIP, voice, Video conferencing, etc
Out of Range	Signals, wireless access point, Internet, etc	Voice, VoIP, email, SMS, MMS, etc

The combined availability is a product of the availability of all the network parts involved. This can be defined as:

$$SA = S(s) * N(s) * L(s) * Sw(s) * D(s) \qquad (1)$$

Where SA – Service Availability

$S(s)$ – Availability of service source

$N(s)$ – Availability of network

$L(s)$ – Availability of link

$Sw(s)$ – Availability of software

$D(s)$ – Availability of destination device

Equation (1) also means that the combined availability of the network is always lower than the availability of its individual components (resources). It is important to note that when network is available, the services being offered will also be available and vice-versa. Therefore, network availability directly impacts on service availability. Simply put,

$$NA = RA = SA \quad where \quad p(F) = 0 \qquad (2)$$

Where NA – Network Availability

RA – Resource availability

SA – Service availability

$p(F) = 0$ - is a probability of fault occurrence is 0 indicating fault-free network

The network availability at time, t for network service, s may be defined in terms of several parameters that includes network reliability $R(t,s)$, network maintainability $M(t)$ and fault effects $F(t)$ where,

$$NA(t,s) = R(t,s) + F(t,s) * M(t,s) \quad (3)$$

Where $F(t,s) = 1 - R(t,s)$ (4)

However, the network reliability depends upon the reliability of many components that make up the network; i.e., network link, power, software, switches, and services. These set of components (resources) can be represented by $R \to (r_1, r_2, r_3, ..., r_j)$. Rewriting Equation (4) to know faults effects at time *t*, for service *s*, is

$$F(t,s) = 1 - (R(t,s))^j \quad (5)$$

A given set of resources consisting of *R* members can be constructed with $R_\pi (\pi = 1,2,3,..., j)$ homogenous sub-populations.

$$\sum_{\pi=1}^{j} R_\pi = R \quad (6)$$

A homogenous sub-population R_π is defined by the verifiable assumption that its members exhibit the same probabilistic decision behaviour. However, these set of resources are affected by faults. Network faults are errors that occur frequently within the network elements impairing their operations. They may render the network elements unusable or partially working thereby diminishing partly or wholly the resources ability to carry out its functions depending on the faults impact.

A set of network faults $F \to (f_1, f_2, f_3, ..., f_i)$, can affect the network resources, R:

$$\alpha : F \to R \quad (7)$$

Where the domain α is the set *F*, the target of α is the set *R*

The range or image of α, written rng α, is

$$rng\alpha = \{r \in R \mid (f, r) \in \alpha \ for \quad some \quad f \in F\}$$

$$= \{r \in R \mid r = \alpha(f) \ for \quad some \quad f \in F\} \quad (8)$$

Therefore the function has its range of resources as the target of network faults given by $rng\alpha = R$; that is every $r \in R$ is of the form $r = \alpha(f)$ for some $r \in R$. Equivalently for any $r \in R$, the equation $r = \alpha(x)$ has a solution $x \in R$. The affects on network resources are transferred to network services with the function:

$$w : R \to S \quad (9)$$

The composition of α and w is the function

$$\alpha \circ w : F \to S \quad (10)$$

Equation (10) is defined by

$$(\alpha \circ w)(f) = \alpha(w(f)) \ for \quad all \quad f \in F \quad (11)$$

VII. Simulation Results

In this Section, we use a case study of network voice service with a number of assumptions made during computation. Software availability was assumed to be 100%, other network faults were not considered except power fault, etc. Source availability, network availability, link availability, software availability, and destination device availability are 0.9958, 0.2343, 0.7737, 1.0, and 0.9958 respectively. We used Equation 1 to compute Service Availability (SA) and obtained 17.982%. Network availability (NA) was computed using Equation 3 and a value of 70.62% was obtained.

However, according to Equation (2), service availability is supposed to be equals to network availability. This is not the case here and it can be attributed to assumptions made, network fault's impact and other factors which are beyond the scope of this work.

Network faults effects on network services at time *t* is given by Equation (5). The value of 99.95% shows that network faults directly affect network services. The margin of 0.05% can be attributed to noise.

The utility of network service (in this case voice) improves with the reduction of network fault (in this case power) occurrence. Network faults occurrence [11, 12, 13] and network service utility matched at the 100th iteration. This point is called acceptance point as shown in Fig. 6.

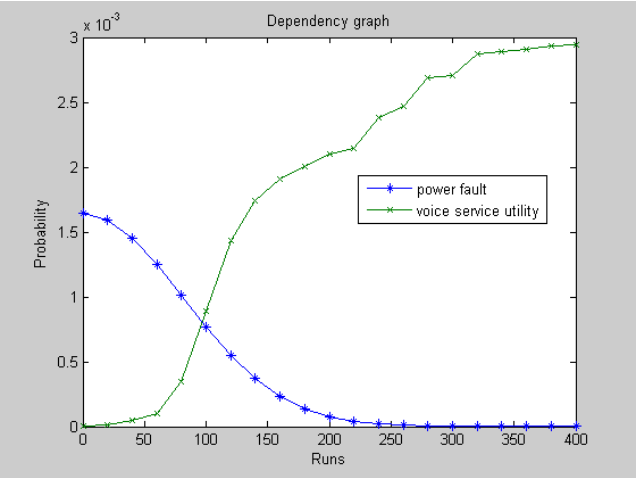


Fig. 6. Network service vs Network faults dependency

VIII. CONCLUSION

The basics of network services delving on characteristics, classification and life cycle are provided in this paper. In our previous publication, we provided the classification and models of network faults prediction using Mobile Intelligent Agents (MIA) [11], [12], [13], which are used to derive the dependency presented in the paper. The basics, types and benefits of network service dependency were presented. The dynamics and dependency life cycle were presented. Network service dependency models are presented with simulation results showing network services dependent on network faults. The value of 99.95% dependability of network services on network faults. The paper gave an insight on the relationship between network faults and network services.

REFERENCES

[1] ITU-T Recommendation I.211, "B-ISDN Services Aspects," Geneva, Switzerland, 1993.

[2] S.V. Kartalopoulos, "A Global Multi-Satellite Network for Multi-Media and PCS Service with Fault and Disaster Avoidance Characteristics, ICC (2), pp.694-698, 1997.

[3] Christian Ensel, "A Scalable Approach to Automated Service Dependency Modeling in Heterogeneous Environments", Proc. 5th International Enterprise Distributed Object Computing Conference, pp.128, September 04-07, 2001.

[4] Humberto Cervantes and Richard S. Hall, "Autonomous Adaptation to Dynamic Availability Using a Service-Oriented Component Model", ICSE, pp.614-623, 2004.

[5] B. Gruschke, "Integrated event management: Event correlation using dependency graphs" Proc. 9th IFIP/IEEE International Workshop on Distributed Systems: Operations & Management (DSOM 98), pp.130-141, October 1998.

[6] D. Caswell and S. Ramanathan, "Using service models for management of Internet Services", In HP Technical Report HPL-1999-43, HP Laboratories, Palo Alto, California, USA, March 1999.

[7] M. Gupta, A. Neogi, M. Agarwal, and G. Kar, "Discovering Dynamic Dependencies in Enterprise Environments for Problem Determination" Lecture Notes in Computer Science, Vol. 2867, Springer Berlin, pp.125-166, 2003.

[8] Maitreya Natu and Adarshpal S. Sethi, "Using temporal correlation for fault localization in dynamically changing networks", International Journal of Network Management, Volume 18 , Issue 4 (August 2008), Pages: 301-314, ISSN:1099-1190.

[9] Paramvir Bahl, Ranveer Chandra, Albert Greenberg, Srikanth Kandula, David A. Maltz and Ming Zhang, "Towards highly reliable enterprise network services via inference of multi-level dependencies" Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications, Kyoto, Japan, Pages: 13 - 24, ISBN:978-1-59593-713-1.

[10] Andreas Hanemann, David Schmitz, Martin Sailer, "A Framework for Failure Impact Analysis and Recovery with Respect to Service Level Agreements", Proc. of IEEE International Conference on Services Computing, Vol.2, pp.49-58, 2005.

[11] Okuthe P. Kogeda, Johnson I Agbinya, and Christian W. Omlin, "A Probabilistic Approach To Faults Prediction in Cellular Networks", In the Proceedings of the 5th International Conference on Networking (ICN2006), Mauritius, April 23 - 28, 2006, ISBN: 0-7695-2570-9.

[12] Okuthe P. Kogeda and Johnson I. Agbinya, "Proactive Cellular Network Faults Prediction Through Mobile Intelligent Agent Technology", In the Proceedings of the 2nd International Conference on Wireless Broadband and Ultra Wideband Communications (AusWireless 2007), Sydney, Australia, 27-30 August 2007. ISBN: 978-0-7695-2846-5.

[13] Okuthe P. Kogeda and Simbarashe Nyika, "Automating Cellular Network Faults Prediction Using Mobile Intelligent Agents", In the proceedings of IEEE 5th International joint conference on INC, IMS and IDC, August 25-27 2009, Soul, South Korea, pp. 1804 -1810, ISBN: 978-0-7695-3769-6.

Dr. Okuthe P. Kogeda was born in Kisumu, Kenya. He obtained a doctorate degree in Computer Science from University of the Western Cape in Cape Town, South Africa in 2009. He obtained Master of Computer Application from Dr. Babasaheb Ambedkar Marathwada University in Aurangabad, India in 1999. His main areas of interest include: Biologically inspired modelling & simulation, Bayesian networks, Cellular networks & OSS, Mobile Intelligent Agents and Data mining. He is currently a Senior Lecturer in the Computer Science Department at University of Fort Hare, South Africa. He was Lecturer in the Computer Science Department at University of the Western Cape in Cape Town, South Africa from 2004 to 2009. He was a Lecturer at University of Nairobi in Nairobi, Kenya from 1999 to 2000. He has published over 15 internationally refereed conference and journal papers and author of a book Chapter in Cellular Networks book.

Dr Johnson I Agbinya holds a doctorate degree from La Trobe University in Melbourne, Australia. He is an Adjunct Professor of Computer Science at the University of the Western Cape, Cape Town, South Africa and a Senior Lecturer at the University of Technology Sydney. He was a Principal Engineer at Vodafone Australia from 2000 to 2003 and a Senior Research Scientist at CSIRO from 1993 to 2000. His areas of research interests are personal area networks, networks on mobile platforms and in uncovered areas, ultra wideband communication, wireless communication networks, mobile network design, wireless access methods, air interface, planning, mobility and moving wireless networks. He also has extensive experience in DSP, signal processing and biometrics. He has published over 80 internationally refereed conference and journal publications, numerous confidentially research reports (CSIRO) and author of a couple of internetworking books.

A Mathematical Model of DNA Replication

Ahmad A. Rushdi

***Abstract*—DNA replication is a fundamental process for cell proliferation in humans and other living organisms. It involves transfer of genetic information from the original DNA molecule into two copies. Understanding this process is of great importance towards unveiling the underlying mechanisms that control genetic information transfer between generations. In this interdisciplinary work, we address this process by modeling it as a data communication channel, denoted: the genetic replication channel. A novel formula for the capacity of this channel is derived, and an example of a symmetric replication channel is studied. Rigorous deduction of the channel flow capacity in DNA replication secures more accurate understanding of the behavior of different DNA segments from various organisms, and opens for new horizons of engineering applications in genetics and bioinformatics.**

***Index Terms*—Communication Channels, Communication Systems, DNA Replication, Mathematical Modeling.**

I. Introduction

ONE of the most important questions which arises in the design process of an information transmission link or processing system is: what is the capacity of this link or system? In other words, in a given time frame, how much information can this link or system carry or process efficiently? In specific, this question is of great importance in modern digital communication systems and has been addressed by several researchers over the past few decades. In fact, a mathematical model that tackled this question successfully was first proposed in 1948 by C. E. Shannon [1]. In his work, Shannon introduced an operational framework to establish a well defined model for wireless communication channels. This generic model starts with an information source that produces a data stream composed of elements picked from a fixed alphabet. The most simple stream is the binary stream composed of bits that belong to the mathematical group $\mathbb{F} = \{0, 1\}$, which is the basis for modern digital communication systems. The data stream then passes through a transmitter that subsequently modulates it and encodes it to form the transmitted signal. The transmitted signal, in turn, propagates through a communication channel until it is delivered to the target receiver. The modulation and coding schemes are to be chosen carefully to secure the transmitted signal's propagation capability and overcome the channel distortion effects. Many noise sources can impact the transmitted signal including additive white Gaussian noise (AWGN) and multipath fading. Upon receiving the transmitted signal, mixed with noise, the receiver is responsible for several procedures including noise cancellation, decoding, and demodulation. Basically, the receiver's function is to reverse every process the original data stream had to go through, either intentionally by the transmitter, or unintentionally due to the channel conditions. Finally, the receiver delivers the recovered stream to the information destination which is the intended target by the data transaction process. Figure 1 summarizes the aforementioned steps.

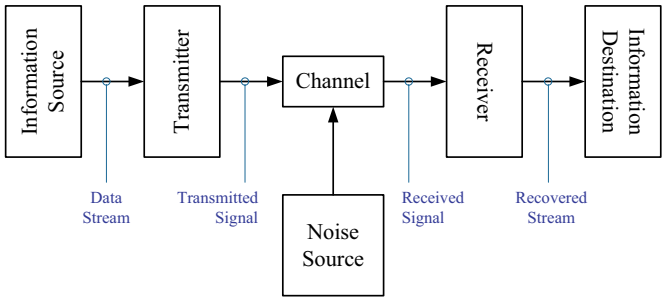


Fig. 1. Block diagram of a basic communication system.

Shannon has also introduced a number of new concepts through his proposed model that we define briefly as follows. The first concept is the entropy; which is a measure of the amount of uncertainty regarding the value of a random variable. For transmitted information message, entropy would be an indicator of how uncertain the receiver was about the contents of that message, or alternatively, how much information was carried by that message. Hence, entropy is typically measured in units of information (e.g. bits). The second concept is the mutual information; which is a measure of the mutual dependence between two random variables, measured in units of information (e.g. bits). For the context of information communications, the mutual information between a transmitted signal and a received one measures how much knowing what the received signal is reduces our uncertainty about the transmitted version. The last and probably most crucial concept in Shannon's work is the channel capacity; defined to be the tightest upper bound on the amount of information that can be reliably transferred through the channel, and is typically measured in units of information per unit time (e.g. bits/sec).

From an application point of view, Shannon's model first came in service to represent communication channels. However, it was easily extended to other communication related procedures and systems including cryptography, coding, information security, and encryption [2]. Furthermore, it was applied to the analysis of non-electrical communication systems such as human communications [3]–[9] and even animal communications [10]. Due to the simplicity

Manuscript received August 31, 2010.
Accepted November 02, 2010.
A. A. Rushdi can be reached at P.O. Box 5542, Santa Clara CA, 95056 USA, (e-mail: aarushdi@ieee.org).

1857-7202/1008011

and mathematical rigorousness of this model, it grew vigorously into new areas that are not directly related to data communications, including the fields of finance and investment [11]–[14], gambling [15]–[18], seismic exploration [19], the study of black holes [20], and most recently, Genetics. On the last front, many signal processing algorithms were proposed to break down and model complex genetic interactions. These contributions paved the road for new interdisciplinary scientific fields such as bioinformatics, molecular information theory, and genomic signal processing [21]–[25].

The Deoxyribonucleic acid (DNA) molecules are simply long-term memory storage cells that contain all genetic information and replication instructions needed for human proliferation [26]. Finding the amount of information embedded or encoded in DNA molecules, the coding system used, and the natural mechanism of data replication and transmission into new generations, are all interesting problems from a data communication point of view. Several researchers have shown interest in unveiling the genetic code [27], [28] in order to address such problems and consequently find out how the complex genetic system is structured.

In this work, we study the application of Shannon's model to the DNA replication process, which is the process of copying a DNA chain of nucleotides into two new chains. This process is the basis for biological inheritance for all living organisms. In order to tackle this problem, we start by changing the way of conceiving DNA from a "chain of molecules" into a "sequence of information elements" that belong to the nucleotide bases set, or the so-called genetic alphabet $\mathbb{F} = \{A, C, G, T\}$. With this perception in mind, the original DNA copy is considered the data source and the two copies are two targeted data destinations. This process is not error-free as might seem obvious, and therefore can be modeled as a noisy channel using Shannon's model.

The rest of this paper is organized as follows. In section II, we derive the necessary mathematical and statistical definitions comprised by Shannon's model. For illustration, we further provide a simple example for a binary symmetric channel. Section III, on the other hand, covers the DNA structure, and its replication process. Then, we study the replication process after characterizing it as a communication model in section IV, which concludes with a simple symmetric replication channel analysis. Finally, our concluding remarks are located in section V. For notation consistency over the paper, symbols for matrices and vectors are shown in bold letters, while scalar variables are shown in italic letters.

II. Shannon Model for Communication Channels

As introduced in section I, Shannon model starts with a data source that produces information messages to be sent to one or more destinations. The sent message is typically composed of blocks that belong to a specific alphabet that is known to the receiver. The transmitter applies different procedures to secure a safe trip for the message through noisy channel conditions. These procedures include amplitude, phase, or frequency modulation and source/channel coding. The receiver collects the received signal and reverses the transmitter's known effects as well as the channel's unknown effects, in order to recover the original information message. In the following discussions, we refer to the communication system model of figure 2 which is a simplified version of figure 1, where the transmitted signal is denoted X and the received copy is denoted Y.

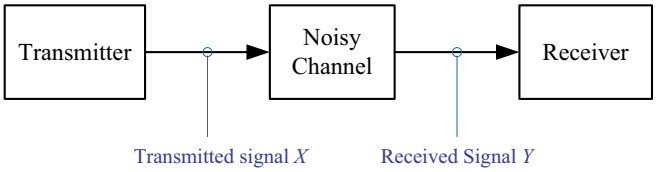


Fig. 2. A simplified version of the block diagram in figure 1.

A. Channel Capacity

In order to quantitatively analyze transmission through a channel, Shannon [1] also introduced an important measure of the amount of information embedded in a message that could be transmitted reliably through this channel: the channel capacity. The "amount of information" in this definition is a measure of uncertainty about the values carried by the message. Based on this concept, a message would be very informative if we have high levels of uncertainty about its contents. In contrast, if we can predict a message with a sufficient level of certainty, then it does not add much information.

B. A Framework to Find the Channel Capacity

In this section, we build a mathematical framework for deducting the channel capacity of a communication channel. Obviously, the amount of uncertainty regarding a message received after going through a communication channel depends on the nature of the original message and the channel effects. Hence, we first define probability matrices representing the transmitter, the channel, and the receiver. Consider a general alphabet $\mathbb{F}$ of N elements. If $\mathbb{F} = \{s_0, s_1, ..., s_N\}$, then we can define the three following matrices [29]–[32]:

- **The source probability vector $\mathbf{P}_s$**, of dimensions $N \times 1$. This vector given in (1) defines the occurrence probability distribution of the elements of $\mathbb{F}$, where $p(s_i) \equiv p(X = s_i)$.

$$\mathbf{P}_s = \begin{bmatrix} p(s_0) \\ \vdots \\ p(s_{N-1}) \end{bmatrix} \tag{1}$$

- **The channel transition matrix $\mathbf{P}_t$**, of dimensions $N \times N$. This matrix defines the transition conditional probability distribution over the channel. An element in this matrix

$p(s_j|s_i) \equiv p(Y = s_j|X = s_i)$ represents the conditional probability that an alphabet element with the row index is received if the element with the column index was transmitted, as in (2).

$$\mathbf{P}_t = \begin{bmatrix} p(s_0|s_0) & \cdots & p(s_0|s_{N-1}) \\ \vdots & \ddots & \vdots \\ p(s_{N-1}|s_0) & \cdots & p(s_{N-1}|s_{N-1}) \end{bmatrix} \quad (2)$$

- **The reveiver probability vector** $\mathbf{P}_r$, of dimensions $N \times 1$. This vector defines the probability distribution of receiving different elements of the alphabet as given by (3), where $p(s_j) \equiv p(Y = s_j)$.

$$\mathbf{P}_r = \begin{bmatrix} p(s_0) \\ \vdots \\ p(s_{N-1}) \end{bmatrix} \quad (3)$$

Figure 3 shows a directed graph combining $\mathbf{P}_s$, $\mathbf{P}_t$, and $\mathbf{P}_r$ in the form of a signal flow graph (SFG). This SFG consists of nodes and branches, where:

- A node denoted $p(X = s_i)$ indicates the probability of transmitting data element $s_i, i = 0, 1, \ldots, N-1$.
- A branch denoted $p(Y = s_j|X = s_i)$ indicates the probability of receiving data element s_j given that data element s_i was transmitted $\forall\ i, j = 1, 2, \ldots, N-1$.
- A node denoted $p(Y = s_j)$ indicates the probability of receiving data element $s_j, j = 0, 1, \ldots, N-1$.

The SFG of Figure 3 is a graphical expression of the total probability theorem. It is usually named a "channel diagram". Using the source probability vector and the channel transition matrix, we further define the joint probability distribution of the two random variables X and Y as in (4).

$$\begin{aligned} p(X = s_i, Y = s_j) &= p(Y = s_j|X = s_i)p(X = s_i) \\ &\equiv p(s_j|s_i)p(s_i) \end{aligned} \quad (4)$$

Based on the probability definitions above, we define a few important quantities pertaining to Shannon model, leading to the definition of the channel capacity:

- **Entropy**; $H(Y)$, of a random variable Y, is the statistical expected value of the information content stored in Y, or alternatively is a measure of the amount of uncertainty regarding the value of Y as stated in (5):

$$H(Y) = E\{I(Y)\} = -\sum_{j=1}^{N} p(y_j) \log_2 p(y_j) \quad (5)$$

where $I(Y)$ is a random variable recording the information content of Y, and the set of values $\{y_j; j = 1, 2, \ldots, N\}$ are the possible values for the random variable Y and $\{p(y_j); j = 1, 2, \ldots, N\}$ are their corresponding probabilities.

- **Conditional entropy**; $H(Y|X)$, is the entropy of the random variable Y upon learning the value of another random variable X, given by (6):

$$H(Y|X) = -\sum_{i=1}^{N} \sum_{j=1}^{N} p(x_i, y_j) \log_2 p(y_j|x_i) \quad (6)$$

where the set of values $\{x_i; i = 1, 2, \ldots, N\}$ are the possible values for the random variable X and $\{p(x_i); i = 1, 2, \ldots, N\}$ are their corresponding probabilities. Intuitively, if X and Y are independent random variables, then $H(Y|X) = H(Y)$ since there is no added knowledge upon gaining knowledge of the value X.

- **Mutual information**; $I(X, Y)$, between the two random variables X, and Y, is a measure of the impact of knowing Y on the knowledge of X. This measure is simply the difference between the uncertainty about the value of X and the remaining uncertainty upon acquiring knowledge about the value of Y as in (7)

$$\begin{aligned} I(X, Y) &= I(Y, X) \\ &= H(Y) - H(Y|X) \end{aligned} \quad (7)$$

- **Channel capacity**; C, is the max value of the mutual information between the transmitted signal X and the recieved one Y, over the source probabilities $p(X)$, as in (8)

$$C = \max_{p(X)} I(X, Y) = \max_{p(X)} [H(Y) - H(Y|X)] \quad (8)$$

The operational meaning of C is the maximum data rate that can be transferred over the channel reliably. Maximization here takes place over the source probability distribution.

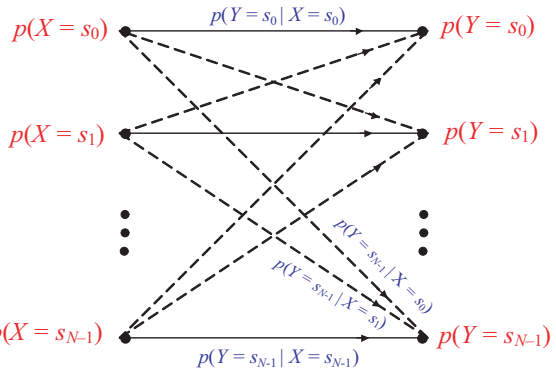


Fig. 3. A signal flow graph representation of a communication channel.

C. A Binary Symmetric Communication Channel

For illustration, we derive the capacity of a simple binary symmetric channel, with only two information elements. That is $X \in \mathbb{F} = \{0, 1\}$, where the two elements are equiprobable. The symmetricity characteristic of the channel indicates equal error probabilities for either element. This error probability is a key factor in the computation of the channel capacity, and is denoted the crossover error probability: ρ. The source

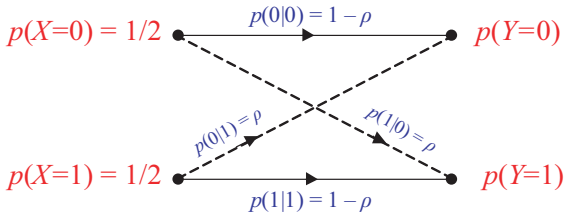


Fig. 4. An SFG represntation of a binary symmetric channel.

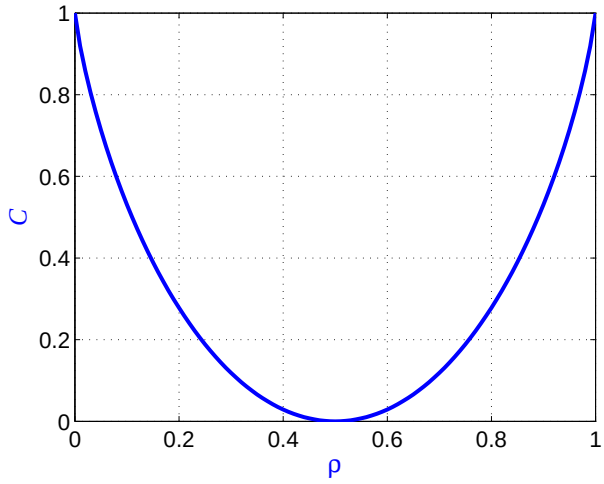


Fig. 5. Channel capacity of a symmetric binary communication channel as a function of the crossover probability ρ.

probability vector and the channel transition matrix can be simplified into (9) and are summarized by the signal flow graph in Figure 4.

$$\mathbf{P}_s = \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix}, \quad \mathbf{P}_t = \begin{bmatrix} 1-\rho & \rho \\ \rho & 1-\rho \end{bmatrix} \tag{9}$$

Using (8), the capacity of the binary symmetric channel can be manipulated into (10) which is a closed form expression

$$C = 1 + \rho \log_2 \rho + (1-\rho)\log_2(1-\rho) \tag{10}$$

The derivation of (10) is given in Appendix A. Figure 5 shows the capacity of the binary symmetric channel as a function of the crossover probability ρ. The value of ρ apparently controls the channel transition matrix, which in turn controls the capacity level. It could be seen from Figure 5 that the capacity function has two maxima ($\rho = 0, 1$) and one minimum ($\rho = \frac{1}{2}$). The formulas in (11) show numeric values of the probability transition matrix $\mathbf{P}_t$ at the maxima and minimum extremes.

$$\begin{aligned} \mathbf{P}_t|_{\rho=0} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{P}_t|_{\rho=1} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \\ \mathbf{P}_t|_{\rho=\frac{1}{2}} &= \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \end{aligned} \tag{11}$$

In light of Shannon's model, (11) could be explained as follows. At $\rho = 0$, no transmission errors occur and the uncertainty vanishes. Therefore, C reaches its maximum value which is equal to unity, and the channel transition matrix reduces into an indentity matrix of rank 2 ($\mathbf{P}_t = \mathbf{I}_2$). The same happens when $\rho = 1$, since an always erroneous transmission could be addressed and fixed by reversing every received data bit to recover the correct transmitted signal. When $\rho = 1/2$, uncertainty is at its maximum, and the capacity goes to a minimum of 0.

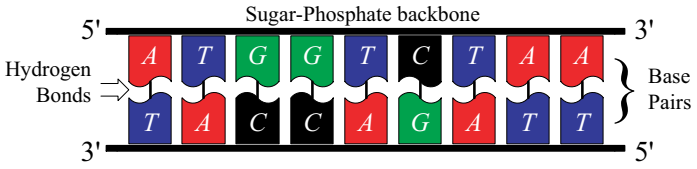


Fig. 6. A DNA double helix composed of two complimentary strands.

III. The Deoxyribonucleic Acid

The deoxyribonucleic acid (DNA) is a long polymer of nucleotides that encodes the sequence of the amino acid residues in proteins using the genetic code [33]–[35]. It is a large, double-stranded, helical molecule that contains genetic instructions for growth, development, and replication after being coded into proteins. A single DNA strand is represented as a sequence of letters that belong to the alphabet $\mathbb{F} = \{A, C, G, T\}$. The letters are the standard symbols given to the four nucleotides in the DNA, namely the two *purines*: *Adenine* (A) and *Guanine* (G) and the two *pyrimidines*: *Thymine* (T) and *Cytosine* (C). A typical DNA double helix is shown in Figure 6. The $5'$ and $3'$ are labels for the two ends of the strand. Each strand reads from the $5'$ end to the $3'$ end.

A. DNA Replication Process

DNA replication is a fundamental process for the inheritance and proliferation of all living organisms. It is a core stage in the central Dogma of molcular biology which refers to the flow of genetic information in biological systems. As shown in Figure 7, genetic information flows from DNA to messenger RNA (ribonucleic acid) to proteins. DNA carries the encoded genetic information for most species. In order to use this information to produce proteins, DNA is first replicated into two copies. Each copy gets converted to a messenger RNA through a process called transcription. Translation is the last stage applied to the information carried by the mRNA in order to construct a specific protein (or polypeptide) that performs a specific function in the cell. The enzyme; DNA polymerase is the major player in the process of DNA replication. It is responsible for recognizing one strand of the parent DNA double helix as a template strand. It then makes a compliment

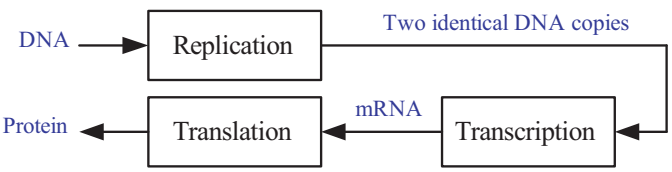


Fig. 7. Central dogma of molecular biology.

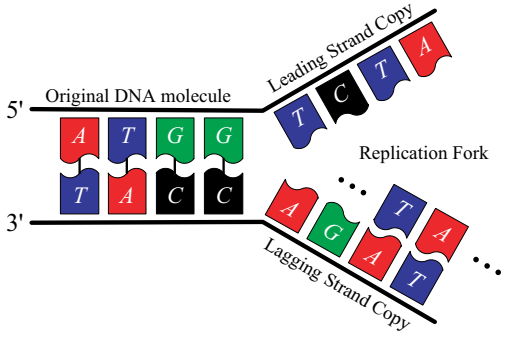


Fig. 8. The DNA replication process.

copy of that template. To do this, DNA polymerase starts by reading the nucleotides on a template strand, finds the complementary nucleotides, adds nucleotides in the $5'$ to $3'$ direction, and finally links those nucleotides into a new DNA strand which will ultimately form a new double helix with the parental template strand.

B. Error Detection and Correction in the Replication Process

The new DNA molecule is supposed to be an identical copy of the original molecule. However, events can occur before and during the process of replication that can lead to deletions, insertions, substitutions and mismatched base pairs. Upon adding the complementary base to the new DNA strand, DNA polymerase proofreads the output to detect and potentially fix errors before moving on to the next base on the template. Once an error is detected, the damaged molecule could be repaired using one of the following natural approaches: mismatch repair, base-excision repair and nucleotide-excision repair. In the following section, we consider the replication process a basic data transmission process. A study of the error correction mechanisms is outside the scope of this work.

IV. DNA Replication Process Modeled as a Communication Channel

Several genetic processes have been addressed and modeled using mathematical models by the bioinformatics research community [36]–[39]. In this section, we analyze the DNA replication process after modeling it as a communication channel. This process involves steps that could be easily mapped to the communication channel model of section II.

A. DNA Replication Channel

As covered in section III, the DNA replication process starts with the original DNA sequence, which is mapped to the transmitted signal X in the communication channel model. The collective mechanisms of the DNA polymerase are mapped to the distortion effects resulting from going through the channel. Finally, writing copies of the original sequence is mapped to the reception stage, and each DNA copy is mapped to the received signal Y.

Interestingly enough, several advanced blocks in the communication system model seem to appear in the DNA replication model, including error detection and correction mechanisms as mentioned in section III. In fact, the DNA nucleotide alphabet could be perceived as a modulated constellation, analogous to the quadrature phase shift keying (QPSK) modulation scheme as illustrated in Figure 9. Finally, copying the original sequence into two identical chains is analogous to multiple-input multiple-output (MIMO) communication systems [40], [41] where more than one antenna could be used in the transmitter or the receiver for bit error rate (BER) reduction purposes. In our case, the replication channel is a 1×2 MIMO communication system.

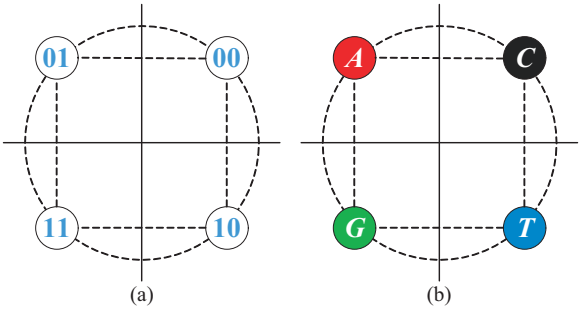


Fig. 9. Analogy between QPSK and DNA encoding systems: (a)QPSK signal constellation, (b) DNA nucleotide constellation

B. Capacity of the Genetic Replication Channel

To completely define the DNA replication channel model, we need to find its capacity. Following similar steps to those in section II, we start by establishing expressions for the source probability vector $\mathbf{P}_s$ and the channel transition matrix $\mathbf{P}_t$. Substituting the DNA alphabet $\mathbb{F} = \{A, C, G, T\}$ into (1) and (2), we get:

$$\mathbf{P}_s = \begin{bmatrix} p(A) \\ p(C) \\ p(G) \\ p(T) \end{bmatrix}, \tag{12}$$

$$\mathbf{P}_t = \begin{bmatrix} p(A|A) & p(A|C) & p(A|G) & p(A|T) \\ p(C|A) & p(C|C) & p(C|G) & p(C|T) \\ p(G|A) & p(G|C) & p(G|G) & p(G|T) \\ p(T|A) & p(T|C) & p(T|G) & p(T|T) \end{bmatrix} \tag{13}$$

Using the general graph of figure 3, we can deduce a signal flow graph to represent the source and channel probabilities of (13) as shown in Figure 10. Now, we can define the entropy of a replicated DNA copy. If the new copy is denoted Y, then $H(Y)$ is a measure of the amount of uncertainty regarding the value of Y which indicates the accuracy of the replication process and is given by (14):

$$H(Y) = -\sum_{y \in \mathbb{F}=\{A,C,G,T\}} p(y) \log_2 p(y) \tag{14}$$

where the set of values $\{y; y \in \mathbb{F}\}$ are the possible values for the random variable Y and $\{p(y)\}$ are their corresponding probabilities. Acquiring knowledge about the original DNA

molecule X makes it possible to find the conditional entropy of Y given X as follows:

$$H(Y|X) = -\sum_{y\in\mathbb{F}}\sum_{x\in\mathbb{F}} p(x,y)\log_2 p(y|x) \tag{15}$$

where the set of values $\{x; x \in \mathbb{F}\}$ in (15) are the possible values for the random variable X and $\{p(x)\}$ are their corresponding probabilities. The mutual information between X and Y could therefore be stated as $H(Y)-H(Y|X)$ and hence a general expression for the capacity of the DNA replication channel could be deduced using (8). To put this conclusion into perspective, we investigate the case of a symmetric channel where the crossover error probability is the same for each DNA base and the bases are equiprobable.

C. A Symmetric DNA Replication Channel

In the following discussion, we consider a symmetric DNA replication channel. The nucleotide random variable X takes values from the genetic alphabet $\mathbb{F} = \{A, C, G, T\}$, where the four elements are equiprobable. The symmetricity characteristic of the channel implies the same error probability for each DNA base, denoted ρ. Furthermore, the value of ρ is split equally towards the other three bases. That is the probability of error from one base to any of the other three bases is $\rho/3$. The source probability vector and the channel transition matrix of this channel can be manipulated into (16) given the previous assumptions.

$$\begin{aligned}\mathbf{P}_s &= \begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix}, \\ \mathbf{P}_t &= \begin{bmatrix} 1-\rho & \rho/3 & \rho/3 & \rho/3 \\ \rho/3 & 1-\rho & \rho/3 & \rho/3 \\ \rho/3 & \rho/3 & 1-\rho & \rho/3 \\ \rho/3 & \rho/3 & \rho/3 & 1-\rho \end{bmatrix}\end{aligned} \tag{16}$$

The capacity of this system is therefore given by (17) which is derived in Appendix B.

$$C = 2 + \rho\log_2\frac{\rho}{3} + (1-\rho)\log_2(1-\rho) \tag{17}$$

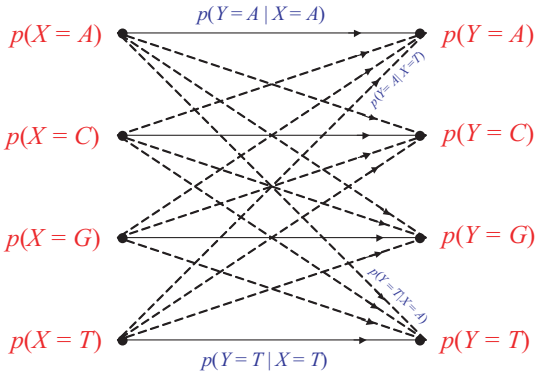


Fig. 10. A signal flow graph representation of a symmetric DNA replication channel.

Figure 11 shows the capacity of the DNA replication channel of (17) as a function of the crossover probability ρ. Similar to the binary symmetric channel case, the value of the crossover error probability ρ apparently controls the channel transition matrix, which in turn controls the capacity level. It could be seen from Figure 11 that the capacity function has two maxima ($\rho = 0, 1$) and one minimum ($\rho = \frac{3}{4}$). In (18), numeric values of the probability transition matrix $\mathbf{P}_t$ are deduced at the maxima and minimum extremes.

$$\begin{aligned}\mathbf{P}_t|_{\rho=0} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ \mathbf{P}_t|_{\rho=1} &= \begin{bmatrix} 0 & 1/3 & 1/3 & 1/3 \\ 1/3 & 0 & 1/3 & 1/3 \\ 1/3 & 1/3 & 0 & 1/3 \\ 1/3 & 1/3 & 1/3 & 0 \end{bmatrix}, \\ \mathbf{P}_t|_{\rho=\frac{3}{4}} &= \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix}\end{aligned} \tag{18}$$

At $\rho = 0$, no replication error occurs and the uncertainty vanishes. Therefore, C reaches its global maximum value which is equal to 2, and $\mathbf{P}_t = \mathbf{I}_4$. At $\rho = 1$, the system approaches an always erroneous transmission case and the capacity goes only to a local maximum value of $2 + \log_2\frac{1}{3}$. Note that in the replication channel case, $\rho = 1$ does not specify which one of the other three nucleotides was supposed to be copied. Therefore, we can not easily fix the error as was the case for the binary symmetric channel with only two data elements. Finally, at $\rho = 3/4$, all transition probabilities go to $1/4$, including erroneous and correct transmission. Uncertainty is at its maximum in this case, and the capacity goes to a minimum of 0.

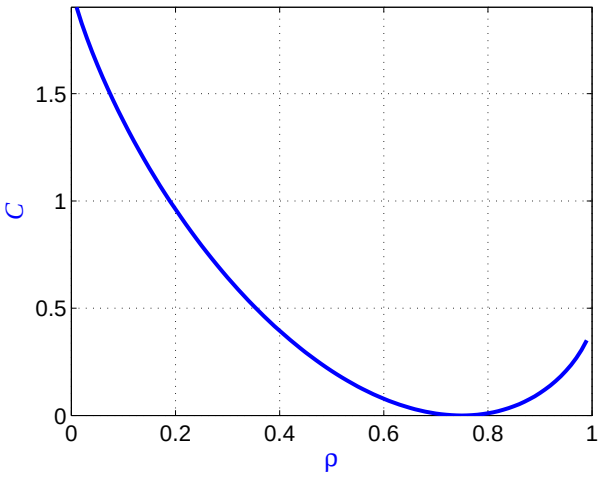


Fig. 11. Channel capacity of a symmetric genetic replication channel as a function of the crossover probability ρ.

V. Concluding Remarks

In this work, we have introduced a simple model for the analysis of the DNA replication process as a communication channel based on Shannon's model for communication channels. We have derived a systematic methodology to find the capacity of this channel and found its value for the case of a symmetric channel. Several remarks are to be observed at this point. First, replication channels are not necessarily symmetric and are different for various DNA segments over different organisms [42]–[44].

Second, using the proposed mathematical framework, we can distinguish between different molecules in terms of their capacity deviation from the symmetric capacity case, or alternatively, the deviation of the channel transition matrix $\mathbf{P}_t$ from the identity matrix. This approach could help solve classical molecular biology problems such as the identification of the protein coding regions or the exon-intron classification.

Third, modern pattern recognition algorithms such as neural networks, fuzzy logic, and genetic algorithms, could be used to solve the classification problem of DNA segments into categories based on how close their transition matrices are to the symmetric case or even to other values of $\mathbf{P}_t$. The neural network algorithm for example is composed of a learning phase where known data are used to "train" the classifier, followed by a testing phase where DNA segments under test are classified into categories based on how close they are to the chosen reference value of $\mathbf{P}_t$, which is used as prior information for the training or learning phases.

Finally, it is important to point out the low computation cost of this algorithm. Computationally efficient techniques are essential when large size DNA databases need to be processed.

Appendix A
Proof of Equation (10)

This proof could be found in most references on information theory (e.g. [31]). It is reproduced here for convenience in order to facilitate the comparison with its peer proof in Appendix B.

Using the source probability vector and the channel transition matrix given by (9), the conditional entropy of Y given X for the binary symmetric channel could be found to be

$$\begin{aligned}
H(Y|X) &= -\sum_{i=1}^{N}\sum_{j=1}^{N} p(x_i,y_j)\log_2 p(y_j|x_i) \\
&= -\sum_{i=1}^{N}\sum_{j=1}^{N} p(y_j|x_i)p(x_i)\log_2 p(y_j|x_i) \\
&= -p(0)\{p(0|0)\log_2 p(0|0)+p(1|0)\log_2 p(1|0)\} \\
&- p(1)\{p(0|1)\log_2 p(0|1)+p(1|1)\log_2 p(1|1)\} \\
&= -(p(0)+p(1))(\rho\log_2\rho+(1-\rho)\log_2(1-\rho))
\end{aligned}$$

Consequently, the mutual information is

$$\begin{aligned}
I(X,Y) &= H(Y)-H(Y|X) \\
&= H(Y)+(p(0)+p(1)) \\
&\star\ (\rho\log_2\rho+(1-\rho)\log_2(1-\rho)) \\
&\leq 1+(p(0)+p(1)) \\
&\star\ (\rho\log_2\rho+(1-\rho)\log_2(1-\rho))
\end{aligned}$$

Therefore, the channel capacity could be manipulated into

$$\begin{aligned}
C &= \max_{p(X)} I(X,Y) \\
&= 1+\rho\log_2\rho+(1-\rho)\log_2(1-\rho)
\end{aligned}$$

where the max capacity could be achieved at the optimal solution point of the maximization problem could be easily found to be: $p(0)=p(1)=\frac{1}{2}$. ■

Appendix B
Proof of Equation (17)

Using the source probability vector and the channel transition matrix given by (9), the conditional entropy of Y given X for the binary symmetric channel could be found to be

$$\begin{aligned}
H(Y|X) &= -\sum_{x\in\mathbb{F}}\sum_{y\in\mathbb{F}} p(x,y)\log_2 p(y|x) \\
&= -\sum_{x\in\mathbb{F}}\sum_{y\in\mathbb{F}} p(y|x)p(x)\log_2 p(y|x) \\
&= -[p(A)\{p(A|A)\log_2 p(A|A) \\
&+ p(C|A)\log_2 p(C|A)+p(G|A)\log_2 p(G|A) \\
&+ p(T|A)\log_2 p(T|A)\} \\
&+ p(C)\{p(A|C)\log_2 p(A|C) \\
&+ p(C|C)\log_2 p(C|C)+p(G|C)\log_2 p(G|C) \\
&+ p(T|C)\log_2 p(T|C)\} \\
&+ p(G)\{p(A|G)\log_2 p(A|G) \\
&+ p(C|G)\log_2 p(C|G)+p(G|G)\log_2 p(G|G) \\
&+ p(T|G)\log_2 p(T|G)\} \\
&+ p(T)\{p(A|T)\log_2 p(A|T) \\
&+ p(C|T)\log_2 p(C|T)+p(G|T)\log_2 p(G|T) \\
&+ p(T|T)\log_2 p(T|T)\}] \\
&= -(p(A)+p(C)+p(G)+p(T)) \\
&\star\ (3\frac{\rho}{3}\log_2\frac{\rho}{3}+(1-\rho)\log_2(1-\rho))
\end{aligned}$$

Consequently, the mutual information between X and Y is

$$\begin{aligned}
I(X,Y) &= H(Y)-H(Y|X) \\
&= H(Y)+(p(A)+p(C)+p(G)+p(T)) \\
&\star\ (\rho\log_2\frac{\rho}{3}+(1-\rho)\log_2(1-\rho)) \\
&\leq 2+(p(A)+p(C)+p(G)+p(T)) \\
&\star\ (\rho\log_2\frac{\rho}{3}+(1-\rho)\log_2(1-\rho))
\end{aligned}$$

Therefore, the channel capacity could be manipulated into

$$\begin{aligned}
C &= \max_{p(X)} I(X,Y) \\
&= 2+\rho\log_2\frac{\rho}{3}+(1-\rho)\log_2(1-\rho)
\end{aligned}$$

where the max capacity could be achieved at the optimal solution point of the maximization problem found at: $p(A) = p(C) = p(G) = p(T) = \frac{1}{4}$. ■

REFERENCES

[1] C.E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 1948.

[2] D. Stinson, *Cryptography Theory and Practice*, CRC Press, 2005.

[3] E. Katz, "The twostep flow of communication," *Public Opinion Quarterly*, vol. 21, pp. 61–78, 1957.

[4] C. R. Berger and S. H. Chaffee, "On bridging the communication gap," *Human Communication Research*, vol. 15, no. 2, pp. 311–318, 1988.

[5] C. R. Berger and S. H. Chaffee, "Communication theory as a field," *Communication Theory*, vol. 9, pp. 119–161, 1999.

[6] A. Chapanis, "Men, machines, and models," *American Psychologist*, vol. 16, pp. 113–131, 1961.

[7] K. Deutsch, "On communication models in the social sciences," *Public Opinion Quarterly*, vol. 16, pp. 356–380, 1952.

[8] D. C. Barnlund, *Interpersonal Communication: Survey and Studies*, Boston Houghton Mifflin, 1968.

[9] G. Gerbner, "Toward a general model of communication," *Audio-Visual Communication Review*, vol. 4, pp. 171–199, 1956.

[10] Z. Reznikova, "Dialog with black box: using information theory to study animal language behaviour," *Acta Ethologica*, vol. 10, no. 1, pp. 1–12, April 2007.

[11] E. Maasoumi, "A compendium to information theory in economics and econometrics," *Econometric Reviews*, vol. 12, pp. 137–181, 1993.

[12] H. H. Permuter Y. H. Kim and T. Weissman, "Directed information and causal estimation in continuous time," in *Proceedings of the IEEE International Symposium on Information Theory ISIT*, 2009, pp. 819–823.

[13] T. M. Cover, "Universal portfolios," *Mathematical Finance*, vol. 1, pp. 1–29, 1991.

[14] T. M. Cover and E. Ordentlich, "Universal portfolios with side information," *IEEE Transactions on Information Theory*, vol. 42, pp. 348–363, 1996.

[15] R. Bell and T. M. Cover, "Game-theoretic optimal portfolios," *Management Science*, vol. 34, pp. 724–733, 1988.

[16] H. H. Permuter Y. H. Kim and T. Weissman, "On directed information and gambling," in *Proceedings of the IEEE International Symposium on Information Theory ISIT*, 2008, pp. 1403–1407.

[17] T. M. Cover and R. King, "A convergent gambling estimate of the entropy of english," *IEEE Transactions on Information Theory*, vol. 24, no. 4, pp. 413–421, 1978.

[18] L. Breiman, "Optimal gambling systems for favourable games," in *Berkeley Symposium on Mathematical Statistics and Probability*, 1961, pp. 65–78.

[19] P. Haggerty, "Seismic signal processinga new millennium perspective," *Geophysics*, vol. 66, no. 1, pp. 18–20, 2001.

[20] S. Chandrasekhar, *The Mathematical Theory of Black Holes*, Oxford University Press, 1998.

[21] J. H. Moore, "Bases, bits and disease: a mathematical theory of human genetics," *European Journal of Human Genetics*, vol. 16, pp. 143–144, 2008.

[22] A. Figureau, "Information theory and the genetic code," *Origins of Life and Evolution of the Biosphere*, vol. 17, pp. 439–449, 2006.

[23] R. Sanchez and R. Grau, "A genetic code boolean structure. ii. the genetic information system as a boolean information system," *Bulletin of Mathematical Biology*, vol. 67, pp. 1017–1029, 2005.

[24] R. Hilleke D. Hutchison T. Alvager, G. Graham and J. Westgard, "A generalized information function applied to the genetic code," *Biosystems*, vol. 24, pp. 239–244, 1990.

[25] J. Hagenauer C. Andreoli T. Meitinger Z. Dawy, B. Goebel and J. C. Mueller, "Gene mapping and marker clustering using shannon's mutual information," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 3, pp. 47–56, 2006.

[26] F. H. C. Crick J. D. Watson, "Molecular structure of nucleic acids," *Nature*, vol. 171, pp. 737–738, 1953.

[27] B. Hayes, "The invention of the genetic code," *American Scientist*, vol. 86, 1998.

[28] B. R. Frieden R. A. Gatenby, "Information theory in living systems, methods, applications, and challenges," *Bulletin of Mathematical Biology*, vol. 69, pp. 635–657, 2007.

[29] A. Y. Khinchin, *Mathematical Foundations of Information Theory*, Dover Publications, 1957.

[30] J. R. Pierce, *An Introduction to Information Theory*, Dover Publications, 1980.

[31] J. A. Thomas T. M. Cover, *Elements of Information Theory*, Wiley-Interscience, 2006.

[32] S. Kullback, *Information Theory and Statistics*, Dover Publications, 1997.

[33] A. Berry J. D. Watson, *DNA: The Secret of Life*, Alfred A. Knopf, 2003.

[34] L. A. Allison, *Fundamental Molecular Biology*, Wiley-Blackwell, 2007.

[35] R. Weaver, *Molecular Biology*, McGraw-Hill Science/Engineering/Math, 2007.

[36] D. Posada, *Bioinformatics for DNA Sequence Analysis*, Humana Press, 2009.

[37] B. F. F. Ouellette A. D. Baxevanis, *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins*, Wiley-Interscience, 2004.

[38] G. R. Grant W. J. Ewens, *Statistical Methods in Bioinformatics: An Introduction*, Springer, 2004.

[39] D. W. Mount, *Bioinformatics: Sequence and Genome Analysis*, Cold Spring Harbor Laboratory Press, 2004.

[40] P. Stoica E. G. Larsson, *Space-Time Block Coding for Wireless Communications*, Cambridge University Press, 2004.

[41] D. Gore A. Paulraj, R. Nabar, *Introduction to Space-Time Wireless Communications*, Cambridge University Press, 2008.

[42] D. Charlesworth J. W. Drake, B. Charlesworth and J. F. Crow, "Rates of spontaneous mutation," *Genetics*, vol. 148, pp. 1667–1686, 1998.

[43] S. L. Crowell N. W. Nachman, "Estimate of the mutation rate per nucleotide in humans," *Genetics*, vol. 156, pp. 297–304, 2000.

[44] D. N. Cooper and M. Krawczak, *Human Gene Mutation*, Bios Scientific Publishers, Oxford, 1993.

Ahmad A. Rushdi received the B.Sc. and M.Sc. degrees in Electronics and Electrical Communications Engineering from Cairo University, Cairo, Egypt, in 2002 and 2004 respectively, and the Ph.D. degree in Electrical and Computer Engineering from the University of Califiornia, Davis in 2008. Since 2008, he has been with Cisco Systems, Inc., San Jose CA. His current work with Cisco Systems involves signal and power integrity for hardware design.

Dr. Rushdi is the recipient of several teaching and research awards, and an active member within the engineering and scientific research communities. His research interests include statistical and multirate signal processing with applications in wireless communications and bioinformatics.

Mapping Network Protocols to Layers of the OSI Model

Götz Peter Gallert

***Abstract*–The 7-layer OSI architecture is the most important model for teaching computer networking. However, the various computer networking teaching sources contain much disagreement, if not outright argument, about which OSI layer describes the respective protocols used in today's networks. If the academic claim is true that the OSI model completely describes computer networking, a consensus should be possible on where any network protocol is located in this model. Such consensus has not been achieved – Is there something wrong with the procedure of mapping network protocols to layers of the OSI model?**

This paper attempts to shed light on the issue by claiming that there are indeed three different sets of criteria for deciding which OSI layer governs what protocol. It describes them, shows how their amalgamation leads to consistency problems, and offers an outline of what OSI layer placement of common protocols they suggest if they are being followed consistently. It further considers the distinction between network protocols and network management protocols, refuting its ability to solve the problem. It concludes by recommending to consistently stick to one set of placement criteria.

***Index terms*–layering, network modeling, OSI, TCP/IP**

1 Introduction

The International Standards Organisation's (ISO) Open Systems Interconnection (OSI) model today is the single most important model for teaching computer networking. It is both universal and clearly defined, two properties that, academically, give it a clear advantage over the more scetchily laid down and protocol-specific TCP/IP counterpart.

It structures the process of information exchange over computer networks and, in its role as a model of the real process, simpifies discourse about design, implementation and deployment of complex internetworking solutions.

Manuscript received October 4, 2010.

G P Gallert is with the Polytechnic of Namibia, Private Bag 13388, Windhoek, Namibia (phone: +264-61-207-2268, fax: +264-61-2072051, email: pgallert@polytechnic.edu.na).

The seven layers of the OSI model describe the tasks to be solved to build a functional internetwork in a modular way [14]. Much similar, although not OSI-based, is the practical implementation of computer networking today: Several protocols together provide the functionality of "the network", each being responsible for a restricted number of tasks.

Consequently, if ISO's OSI reference framework really is *a model* of computer networks it should be able to describe the interdependence of prevailing protocols within its seven layers. But while there is agreement that every network protocol fits *somewhere* into the OSI model it is sometimes not clear exactly where it fits.

The reason for the difficulty populating the OSI model with today's common-use network protocols emanates mainly from the situation that most of them are part of, or related to, the protocol stack of TCP/IP. From the standards formulation in the respective Request for Comment (RfC) it is sometimes made explicit which TCP/IP layer is responsible for a given protocol [2] [3], but layering *with respect to the OSI model* often remains unclear.

Furthermore, IETF protocol designers are least concerned about allocation to OSI layers and indeed do not intend to follow the principles of OSI layering, i.e. the complete independence of lower layers from upper layers and the concept of the sole reason for the existence of layer $n-1$ being support for layer n, c.f. [9].

Possibly as a result of this uncertainty, computer networking text books seldom explicitly thematize this issue. Instead, the claim that a certain protocol is on a particular layer of either model is often hidden in the general structure of the text (the table of contents) and does not receive further coverage [cf. e.g. 12]. Various sites on the World Wide Web however do discuss the topic. As those sources (this paper uses [13] and [1] as two examples of thousands of pages in this area) are much easier available than printed material, the necessity arises to discuss the merits of their claims because tertiary education in computer networking is heavily affected by web sites explaining "how it really works".

1857-7202/1010001

1.1 Some contested protocols

Wikipedia[1] discussion pages[2] on the subject area provide some insight into the thoughts of contributors with different professional and educational background, all of whom claim to have some authority on the matter. These discussions [13] reveal the following main points of disagreement:

1. Whether routing protocols should be on OSI layer 3 because they facilitate end-to-end `IP` delivery, or on other layers depending on what functionalities they use,

2. Whether protocols should be placed on the highest OSI layer they contribute to, or on the lowest (e.g. in the cases of `DHCP` and `HTTP`),

3. Whether the exercise of OSI-layering protocols that have been designed independent of OSI has any merit at all.

Also on more formal platforms authors disagree about certain protocols. A Cisco white paper on TCP/IP technology places Address Resolution Protocol (`ARP`) on OSI layer 2 [10, p.2], Forouzan on layer 3 [6, p.43]. Even its location in the TCP/IP model seems not to be entirely obvious; the Cisco CCNA Discovery curriculum (version 4.0, module 2, chapter 7.2.1.2) stacks it between Internet and Link layer.[3]

1.2 Relationship of the TCP/IP and OSI Reference Models

As with the layer placement of particular protocols, the relationship between TCP/IP and OSI reference models is not often made explicit in standard networking texts. Few of them go as far as suggesting certain TCP/IP protocols *to belong to* a particular OSI layer as in [11, chapter 1]. Most do, however, offer a model comparison that is often depicted as in figure 1: This graphic suggests protocol location to be a surjective function from OSI onto the TCP/IP model. It particularly:

- restricts the placement of Link-layer TCP/IP protocols to OSI layers 1 and 2,
- maps Transport-layer TCP/IP protocols to OSI layer 4, and Internet-layer TCP/IP protocols to OSI layer 3, and
- restricts the placement of Application-layer TCP/IP protocols to OSI layers 5 to 7.

[1] `http://en.wikipedia.org`

[2] It should be emphasized that the reference here is to *the discussion pages* of the articles, typically batches of comments that as a matter of netiquêtte are not extensively edited. This is in contrast to *the articles themselves* that might change to better or worse within a matter of minutes.

[3] For consistency in this paper the OSI layers 2 and 3 are called "Data Link" and "Network" layer while the TCP/IP layers 1 and 2 are called "Link" and "Internet" layer, respectively.

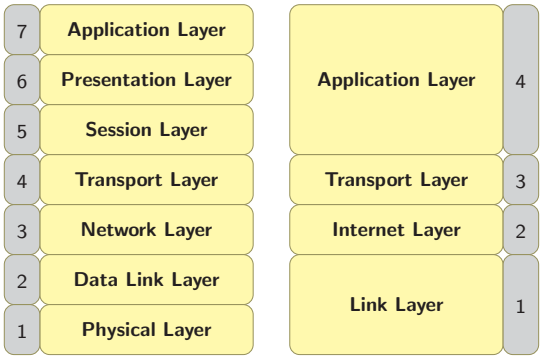


Fig. 1: A juxtaposition of the OSI model (left) and the TCP/IP model (right) as suggested e.g. in [6] and [12]

This relationship can be criticised. While a detailed reasoning in this regard is beyond the scope of this article [see 8, for a thorough discussion], one example shall be given to pinpoint the problem: For the OSI model, the responsibility of the Network Layer lies in the provision of end-to-end delivery of datagrams, in contrast to next-hop delivery which is the responsibility of the Data Link layer [14, p.430]. On the other hand, the TCP/IP model sees its Internet layer as the virtual network interface [7, p.8], something that clearly includes next-hop delivery. (The disagreement about `ARP` is based on this difference).

The relation between the two reference models is therefore not a function at all, and figure 1 is an oversimplification.

1.3 Layering Considered Harmful?

RfC 3439 [9] contains a section titled "Layering Considered Harmful", a paragraph sometimes quoted to scetch the seemingly ultimate impossibility to decide on OSI layers for common protocols [1] [13].

Claims like these stem from a misunderstanding about the purpose of RfC 3439: It uses a completely different definition of layering and does not say anything at all about network models as such. It particularly does not reference RfC 1122 nor RfC 1123, the two documents defining the TCP/IP layers. What has indeed been considered harmful by the authors is the over-diversification of the protocol landscape.

To a certain degree RfC 3439 repudiates the necessity to design IEEE protocols in a way reconcilable with the OSI framework: Artificial separation of responsibilities by layers – OSI or TCP/IP – is to be avoided if it does nothing to improve implementation. This observation ought to be kept in mind when performing any TCP/IP-to-OSI mapping but it should never be used as a critique of the IEEE standardisation process. It is the obligation of the scientists to prove that a model is a useful abstraction of the real-life process; to argue the other way round is unrealistic.

1.4 Network Protocols vs Network Management Protocols

To avoid some of the issues regarding layer mapping it has been suggested to distinguish network protocols from network management protocols [13] or *core* protocols from *supporting* protocols, [6, p.43], arguing that the management protocols do not carry user data and are therefore to be regarded as necessary overhead to "real" networking communication. The rationale behind this suggestion might have been that it is foremost the management protocols that seem to evade OSI classification, and not so much the protocols that actually transfer user data.

With this distinction one can apply different strategies of OSI-layering, one for the "genuine" and one for the management protocols, e.g. by basing the classification of "normal" protocols on functionality but placing the management protocols into the layer they actually manage. However, difficulties seem to begin, rather than end, at this classification:

1. The distinction itself is not easy to do: Whether `TCP` transmits user data or not depends on its current payload: if it is an `HTTP` request it does, if it is a `BGP` update it doesn't.

2. The term "user data" lacks satisfactory precision: Is there any user data transmitted by Samba? Does a user-initiated `traceroute` constitute user data or not?

3. The distinction fails for multi-layer protocol stacks like VoIP or ATM, as they will typically contain both management and user data protocols.

4. The layering instructions for management protocols is just as opaque as before: for instance, which layer is managed by `DHCP`? By `NTP`?

In conclusion, the following picture develops: Applying a narrow definition of "user data", the genuine protocols in `IP` networks are simply the core protocols from the TCP/IP protocol stack: `TCP`, `UDP`, `IP`, and some clear cases on link and application layer. All other protocols are management protocols. The instruction to sort them into layers "by what they manage" therefore simply reverts to the layering instruction that will below be called the *conceptual approach*, to matching the protocol's and the layer's responsibilities.

2 Three General Approaches

Investigating the reasons given for particular placements of protocols into layers of the OSI model, at least three major classes of arguments seem to prevail:

1. Functional approach: What data does this protocol send? Conclude the placement of the protocol from the payload of its data units.

2. Operational approach: On what functionality does this protocol rely? Place the protocol dependent on the services it uses.

3. Conceptual approach: What is the responsibility of this protocol? Decide the placement of the protocol by determining its contribution to network connectivity.

Of course these different approaches can lead to different results, and mixing them to populate one and the same network model leads to the described difficulties. But just as mixing the three strategies yields undesired, even haphazard, results, the effect of consistently using one method for layering might equally surprise. The following sections describe the three notions in detail.

2.1 The Functional Approach

The functional approach makes the assertion that for any PDU, if the payload is on layer n of the TCP/IP model, the PDU itself belongs to layer $n-1$.

> *"From the IETF perspective, the payload defines layering."* [13]
>
> Howard C. Berkowitz, IETF engineer and author

The appeal of this principle is that it directly follows the TCP/IP design approach with regards to encapsulation by arguing e.g that any PDU that has a `TCP` segment as payload must be on the Internet layer.

For the core TCP/IP protocols this is entirely correct: The payload of any transport layer segment is application data, the payload of a packet is a segment, the payload of a frame is a packet. But two questions remain unanswered by the functional approach:

1. How can this principle be applied to layers *of the OSI model*? As outlined in section 1.2 the mapping between OSI and TCP/IP is not entirely trivial.

2. What should be done with protocols that do not encapsulate any upper-layer PDU (the *management protocols* as characterised in section 1.4)?

There is no satisfying answer to question 1. If the network is operated by the TCP/IP protocol stack, peer-to-peer information that would belong to OSI layers 1, 5, and 6 is included in the encapsulation process on layers 2 and 7, respectively. As such the layers under the functional approach would be *defined* by the encapsulation process. This happens only four times, and OSI layers 1, 5, and 6 would necessarily remain empty.

There are two possible solutions to the problem in question 2. One is to place all management protocols on the highest layer because they do not fulfill the requirements to be placed anywhere else. This mapping would lead to a layer mapping as shown in figure 2.

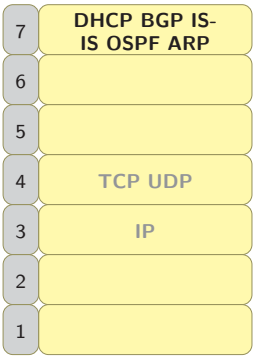


Fig. 2: OSI placement under the functional approach

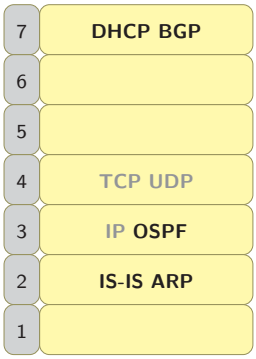


Fig. 3: OSI placement under the operational approach

Another attempt would be to investigate those management protocols separately and not to decide their placement at this stage.

Both these possible solutions have disadvantages. The first possibility leads to a counter-intuitive protocol placement where most functionalities are concentrated on OSI layer 7, the second one reduces the entire approach to a solution dependent on other approaches to answer "the interesting" questions. Indeed, answer 2 would decide layer placement only for all trivial cases.

2.2 The Operational Approach

There are several versions of this approach. In its most general form, the operational approach makes the assertion that no protocol x may be placed below a protocol y, if x relies on y to operate. More restrictive characterisations are:

1. If a protocol uses functionality on layers

   $$n_1, n_2, \ldots, n_m \in N$$

   it belongs to layer $k = \max\{N\}$.

2. If a protocol uses functionality on layers

   $$n_1, n_2, \ldots, n_m \in N$$

   it belongs to a layer $k > \max\{N\}$.

This is the opposite of the functional approach – it concentrates not on the payload but the encapsulation of a PDU. The operational approach enforces a layer mapping as outlined in figure 3. The appeal of this approach is that it formalises the constitutional principle of the OSI model: that the sole *raison d'être* of the lower layers be to support the higher layers [as outlined in 14, p.430]. Furthermore it avoids the potential ambiguity of placing protocols that span multiple layers of either model.

Unilateral use of the operational approach leads to the situation that functionally similar protocols might be scattered all over the place. This becomes particularly visible in the case of routing protocols which all have more or less the same functionality but differ widely in implementation, and therefore in their usage of network functionality.

There is a further problem with this strategy: To decide whether a certain protocol relies on another one is not entirely straight-forward. For instance, both `DHCP` and `BGP` are assigned well-known ports. Still `DHCP` does not rely on `IP` and `UDP` functionality in the same way that `BGP` relies on `IP` and `TCP` because `DHCP` often is implemented within the same broadcast domain.

2.3 The Conceptual Approach

The conceptual approach makes the assertion that each protocol belongs to the OSI layer that matches its responsibility.

The appeal of this approach is that it best maps the design intentions of the OSI model: to assemble a framework of responsibilities that together define and accomplish a general, common functionality. It also fits best into the model character of OSI as it offers some explanation behind layer placement as opposed to just a technical criterion. Furthermore it does not reqire the additional bypass of refering to TCP/IP layering first, and OSI layering second. It thus is not susceptible to the problems outlined in section 1.2.

This approach is the hardest to follow as the high-level purpose of networking protocols is rarely well documented, and if it is it often happens years after the protocols have been standardized.[4] Furthermore, as protocol designers are not bound by the OSI layer definitions, not every protocol might strictly fit one layer – as RfC 1925 humorously puts it:

> *"It is always possible to aglutenate* [sic] *multiple separate problems into a single complex interdependent solution."* [4]
>
> R. Callon, RfC 1925, Fundamental Networking Truth #5

[4] For the Internet protocol suite see e.g [5] which was published 15 years after the first protocol proposals.

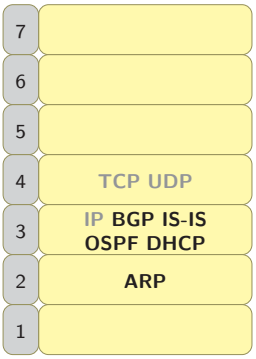


Fig. 4: OSI placement under the conceptual approach

The conceptual aproach leads to certain uncomfortable situations with protocols that have responsibilities on multiple layers. DHCP for instance provides IP addresses (OSI layer 3), gateway information (a layer 2 OSI function), and domain information (layer 7).

While this is in itself not a problem – many protocols are actually protocol stacks and cover multiple layers – it might still seem strange that certain layers are completely left out. This defies the OSI layering principles as laid out in [14]; DHCP certainly has no responsibility whatsoever on OSI layers 4 to 6, and DHCP is clearly a singular protocol and not a stack.

It must be reiterated that the IETF refutes these principles (cf. section 1.3). The perceived conceptual weakness therefore is inherent to the IETF protocols, not to their mapping to the OSI model. Still the question remains where to place protocols whose functionalities span multiple OSI layers, an exhaustive discussion of which is considered beyond the scope of this article.

The rigorous attempt to solve this challenge might be to put them to every layer they contribute to in terms of functionality, because the solution that may seem obvious and desirable: to place protocols on their *main* layer of contribution, will necessarily introduce new ambiguity.

The pragmatic attempt might be to place such protocols into a layer of choice depending on the main focus of the point to be made. OSI layering is, as stated before, mainly an academic exercise. If for example the aim is to explain how Data Link Layer connectivity is "lifted" to the Network Layer with automatic address allocation it makes sense to place DHCP on OSI layer 3 (as done in figure 4). If on the other hand the automation of network configuration tasks is focused on, DHCP acts much like an application providing the necessary parameters, and could be placed on OSI layer 7.

2.4 Evaluation and Conclusion

The functional approach is either not an independent solution to the problem, or it decides only trivial cases and enforces a distinction into network protocols and management protocols (cf. section 1.4), a strategy that in turn invokes the conceptual approach (cf. 2.3). The functional approach therefore must be rejected as a strategy to map TCP/IP protocols to layers of the OSI model.

What remains are the two independent strategies, the operational approach (layer to reliance) and the conceptual approach (layer to responsibility). Although the assumption underlying current OSI layering practice – that the TCP/IP layer predetermines the OSI layer – lacks foundation, both have their merits and lead to an acceptable layer population that avoids ambiguity in the variable of interest.

Problems arise when both methods are combined: To argue that BGP must be above OSI layer 4 for its reliance on TCP (operational approach) and then to place it on layer 7 because it has no responsibilities on layers 5 and 6 (conceptual approach) lacks rigor because the conceptual approach would place any routing protocol on OSI layer 3, not 7, in the first place. Alternatively, to place ARP and DHCP both on layer 3 because of ARP's implementation over the link layer (operational approach) and DHCP's main role (provide IP addresses, conceptual approach) is methodologically inferior.

The operational approach leads to uncommon and counter-intuitive protocol placement (cf. 2.2). The conceptual approach differs slightly from common practice and is the hardest to apply because its criterion is harder to formalise. Using this approach, placement of every networking protocol on one singular OSI layer is hardly possible, given the design goals of IETF [9]. It has two advantages, though: it is the closest to the definition of the OSI model, and the one generally applicable throughout the protocol landscape.

From a teaching perspective a method that underscores the model properties (simplification, structuring, and explanation) of OSI could be suited best, and the conceptual approach could therefore be prefered over other attempts. Other standpoints might of course merit a different choice.

References

[1] 4engr. *Internet Protocol Suite.* 4engr. All Your Engineering Information, August 2009. http://www.4engr.com/dictionary/catalog/3693/index.html. Last accessed 25 August 2009.

[2] R. Braden, editor. *RfC 1123. Requirements for Internet Hosts – Application and Support.* IEEE RfC, 1989. Definition of Application Layer of the TCP/IP model.

[3] R. Braden, editor. *RfC1122. Requirements for Internet Hosts – Communication Layers.* IEEE RfC, 1989. Definition of Link Layer, Internet Layer and Transport Layer of the TCP/IP model.

[4] R. Callon. *RfC 1925. The Twelve Networking Truths.* IEEE RfC, 1996. April 1st RfC.

[5] David D Clark. The Design Philosophy of the DARPA Internet Protocols. *Proceedings of SIGCOMM. Computer Communication Review*, 18:106–114, August 1988. Re-published 1995.

[6] Behrouz A Forouzan. *Data Communications and Networking*. McGraw Hill, 4th edition, 2007.

[7] Lydia Parziale et al. *TCP/IP Tutorial and Technical Overview*. IBM RedBook series. IBM International Technical Support Organisation, New York City, 8th edition, 2006.

[8] D M Piscitello and A L Chapin. *Open Systems Networking: TCP/IP and OSI*. Addison-Wesley, 1993.

[9] D. Meyer R. Bush. *Some Internet Architectural Guidelines and Philosophy*. IEEE RfC, 2002. Section 3 L̈ayering Considered Harmful:̈

[10] Cisco Systems. *TCP/IP Technology*. Cisco Systems Technology White Paper. Cisco Systems, 2005. `http://www.cisco.com/en/US/docs/internetworking/technology/handbook/ito_doc.html`. Last accessed 14 August 2009.

[11] Cisco Systems. *Internet Technologies Handbook*. 2008. `http://www.cisco.com/en/US/docs/internetworking/technology/handbook/ito_doc.html`. Last accessed 7 July 2009. (Series of chapters without author information; not published as entire book).

[12] Andrew S. Tanenbaum. *Computer Networks*. Pearson Education International, 4 edition, 2003.

[13] Wikipedia Talk. *Wikipedia Talk: OSI model*. English Wikipedia, August 2009. `http://en.wikipedia.org/wiki/Talk:OSI_model`. Last accessed 25 August 2009.

[14] Hubert Zimmermann. OSI Reference Model – The ISO Model of Architecture for Open Systems Interconnection. *IEEE Transactions on Communications*, COM-28:pp. 425 – 432, 1980. Definition of the OSI Model.

Peter Gallert (born 1971) holds a *Magister* degree in logic, theory of science, communication studies and media science from University of Leipzig, Germany.

He has been working as Senior Lecturer: Information Technology at Polytechnic of Namibia in Windhoek since his emigration to Namibia in 2000. He is currently working on a quality of service implementation for meshed overlay networks in cooperation with the University of Cape Town.

Fast Geographic Internet Mapping System for Location-Based Services

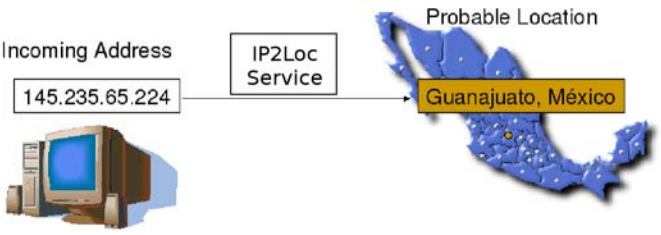
Edmar Mota-García[†] and Rogelio Hasimoto-Beltran[‡]

***Abstract*— We propose an Internet (IP) mapping system for discovering network topology and geographic location of hosts (routers/end-nodes) based on four main steps: a) domain name based mapping, b) host IP clustering, c) clustering refinement via host delay location, and d) IP distance similarity using IP differences. Unlike previous schemes that use large databases and intensive route probing systems over different Internet Service Providers (ISPs), our scheme makes use of small but reliable database tables spread across the region of interest (no probing is needed over different IPSs), Border Gateway Protocol (BGP) tables, and scalable traceroute information (that includes delay and connection information) for a fast and effective IP to location mapping (no ISP probing is needed). We concentrate on providing fast node location information for Location-Base Services (LBS) to support multimedia delivery architecture design (previous schemes focus on accurate geographic location of nodes). As a study case we analyzed the router-level topology of the Internet in México and built an IP address to location mapping table. The order of magnitude of the complete analysis over the region of interest is in the order of minutes to hours, whereas previous schemes order of magnitude can go from days to weeks. Our approach can be successfully applied anywhere in the world with the corresponding initial database table.**

***Index Terms*— Geographic Mapping, Traceroute, One-way Transit Time, Content Delivery Network.**

I. Introduction

BUILDING an IP address to location mapping (IP2loc) service is an interesting problem that has received a lot of attention during the last years. The aim of an IP2loc service is to find the probable geographic location of a node or router in a predefined area of interest based on its IP only, as depicted in Figure 1. Depending on the complexity of the scheme, probing duration and size/density of the initial database table, the certainty of the assigned location can be as high as hundreds of meters around the real node location. Our aim is practical scenarios of Content Delivery Networks (CDN) [14],



where a fast and efficient IP2Loc service is more important than location precision. This type of service could enable an interesting class of location-aware services in media delivery applications for Internet hosts [1] (we will use host and routers interchangeably throughout the rest of the paper). Under this panorama, the closest server can be in charge of serving clients requests for improving delivery delays; additionally, servers can use more than one path (known as path diversity) to deliver information to clients for a more effective use of bandwidth as shown in [2]. Location-aware applications can track network properties more easily and use them as Quality of Service (QoS) measure for the next data delivery.

Fig. 1. IP address to geographic location.

Discovering network topology and geographic location of routers and end-nodes (extreme nodes in a communication process) is not an easy task. Internet Service Providers (ISP) keep their router-level topology confidential, and when trying to infer it by software tools such as traceroute, routers may not reply to ICMP packages or may reply with different misnamed domain names [15]. At the end, only an approximation of the real IP topology can be obtained.

Current schemes in the literature use tremendous amount of time and probes to infer the IP topology in detail (which can take days to weeks to converge to the specified topology), in addition to require large databases with sometimes unreliable mapping list (applied to a particular ISP at a time). This huge effort in providing a high detailed mapping could be useless due to the unceasing growth and dynamic nature of the network (host mapping could be degraded up to a 70% in a period of a month) [3]. Our goal in designing a new Internet router level mapping scheme is to obtain an acceptable level of topological detail for CDNs with less effort and resources (in minutes to hours), and without depending on third-party generated information (ISP probing). We prove that a small and independent of any ISP database of homogeneously spread nodes over the region of interest provides the same

Manuscript received October 20, 2010. This work was supported in part by the State Council of Science and Technology of Guanajuato-México (CONCYTEG) under grant number 07-02-K662-044-A01.

[‡]R. Hasimoto-Beltran is with the Department of Computer Science at the Center for Research in Mathematics (CIMAT), Guanajuato, México (corresponding author e-mail: hasimoto@cimat.mx).

[†]Edmar Mota-García is with the National Institute of Public Education, in Sinaloa, México (edmar@cimat.mx).

backbone than the one obtained by intensive probing over multiple ISPs. The level of detail or resolution in our scheme is flexible depending on the initial input data density. To validate our proposed scheme, we present a study-case to infer the Mexican Internet topology and compare our results against one of the state of the art tools named RocketFuel [5].

The paper is organized as follows. In the next section we discuss current methods for Internet mapping. Section 3 and 4 describe the proposed approach and results respectively. Future work and concluding remarks are presented in section 5.

II. Related Work

Several research efforts have been performed to infer the Internet routing topology (backbone) and geographic location of routers through different approaches. The Mercator tool [4] is used to create maps of Internet hosts without considering their geographic location. Mercator uses traceroute like probes to obtain host connectivity and iteratively improves the mapping process (address-to-location) using heuristics for the exploration of the IP space called informed random address probing. This probing scheme is aimed at minimizing the number of probes sent to the network using a subset of the address space. In [1], authors present an IP2Geo scheme for determining the geographic location of Internet hosts. IP2Geo consists of three techniques: GeoTrack to infer location from DNS names, GeoPing for location interpolation using ping delay measurements, and GeoCluster to create host clusters based on BGP tables and an initial IP2loc list. They report a 30% localization error (compared to the correct ISP topology).

Rocketfuel [5] was developed to map the topology inside the Internet Service Provider (ISP). It is composed of a series of techniques similar to those found in IP2Geo tool [1], which interact each other to build the topology for a given ISP. One problem with RocketFuel is that the level of complexity grows with the size of the ISP. Each ISP has to be probed and analyzed separately, so that the time to completely map a region of interest grows with the number of ISP present in the area. In [6], authors explore the use of Rocketfuel for path diversity discovery, proposing minor changes to maximize the probability to correctly infer the inner topology of the targeted ISP. Despite of the great level of detail obtained by Rocketfuel, the modified scheme presents higher rate of false and missed links. Their conclusion is that, increasing the level of detail or increasing the probing time does not necessarily change the minimal average error (reported to be 15% of the address space probed).

Using delay and topology information can be helpful in providing better geographic location as proved in [7], the Vivaldi protocol [8]), and more recently in [9, 10]. In [9], authors claim that the utilization of delay-based geolocation is preferable since eliminates the need to solely rely on data that could be incomplete, inaccurate or misleading like DNS registers. Furthermore, they report that a simple ping-like probing (like the ones returned by traceroute output) is sufficient to be used as a host geolocator.

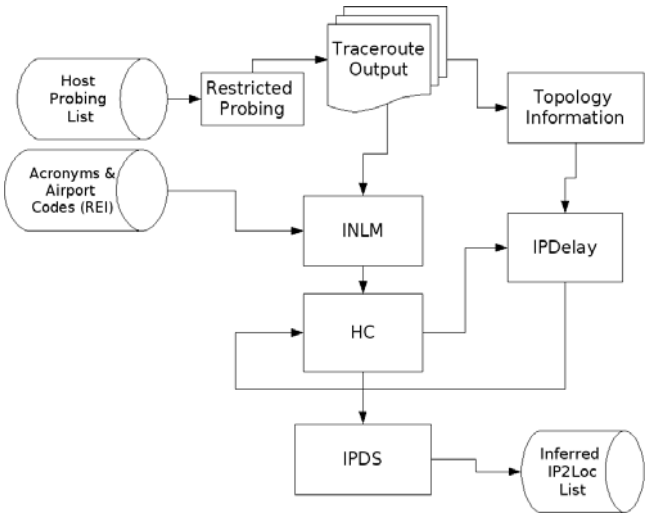


Fig. 2. Diagram of the proposed scheme.

In this work, we propose a new system for IP address geographic location, where an acceptable level of topological detail can be accomplished using a simpler scheme. Our proposed scheme do not probe any ISP in particular, instead it uses a small and reliable database homogeneously spread across the region of interest (initial IP-to-location list), along with restricted probing and hosts delay from different landmarks nodes to discern from hosts having the same name or address prefix, but located in far apart regions (which makes our scheme robust against misnamed DNS [15]). Our aim is to provide a rapid node geolocation system in a particular region of interest for real-time content delivery services (video, audio, web content, etc.). The outcome of our system includes the connection topology in the region plus an IP2Loc list for use with newly incoming clients.

III. Proposed Scheme

Our approach attempts to infer the connection topology and geographic location of hosts with a specified level of detail (depending on the initial database node density), by combining 4 main algorithms (Figure 2):

a) Initial Name-Location Mapping (INLM). It makes use of domain name information in order to obtain an initial guess on the location of the hosts. This algorithm is affected by DNS misnaming, which accur in the 0.5% of the IP addresses [15].

b) Host clustering (HC). A clustering algorithm is applied to the initial localization of hosts to find groups of IP prefixes that share a common location. For this step, a predefined IP2loc prefix list is used to refine the host localization process in previous phase (a).

c) Topology and Delay Location (IPDelay). We use the topology and delay information returned by traceroute to infer the location of unclassified hosts in the initial phases (a and b). A new host clustering phase takes place to refine the localization process. This algorithm makes the scheme robust to DNS misnaming [15].

d) IP distance similarity (IPDS). As a final step in our approach we use a simple IP number distance based metric to locate isolated and not recognized addresses.

These four algorithms work together to provide an IP2Loc mapping and to construct a connection topology of a particular region in the Internet. An additional level of detail can be provided by augmenting the information provided in the probing phase (algorithm a). A detail explanation of each algorithm is described next.

3.1 Initial Name-Location mapping (INLM)

In order to minimize the probing phase in our scheme, we propose the use of a small Host Probing List (HPL) dispersed over the region of interest. The HPL (provided by the Institute of Educational Technology of the Department of Public Education and Culture of the state of Sinaloa, México–IET-DPEC), consists of web portals of local governments, educational institutes, companies, etc. for which their physical locations and IP addresses are known. In general, the HPL takes one host per city in the region of interest, but is flexible enough to allow finer (coarser) resolution to the system by adding (eliminating) elements to the list. The HPL is complemented with selected hosts called landmarks. Landmarks help in the probing process by gathering delay and connection information (from different perspectives) when a host does not fit in any particular IP group. A restriction on the selection of the landmarks is that they must not be behind a firewall, in order to receive ICMP and/or UDP packets.

The process starts by running traceroute or tracert from each landmark to each host in the HPL to obtain router-level connection and domain name information. This is what we call an Initial Restricted Probing phase, which differs from other schemes that use of random probing instead [4].

The domain name information of intermediate routers is compared with the ISPs naming convention in the area of interest [1, 5]. Unfortunately, this information is not publicly available, but it is possible to infer some patterns after analyzing the current structure of the ISPs domain names. Specific extensions or codes used by the DNS may be helpful to separate those hosts located in the analyzed country from the rest (for example mx for México, se for Sweden, jp for Japan and so on), and within the country itself at a finer resolution (at state and/or city resolution). We found that the domain naming convention used by the main ISPs in México includes city and state acronyms in their clients, routers and backbone routers in most of the cases. We additionally created a database containing airport codes, strings bb and core for backbone routers, and other synonyms that give a glimpse of the nature of the host in their domain name.

The INLM algorithm filters out any private IP address denoting an erroneous behavior of the routers, as for example 10.xx.xx.xx prefixes. This is the minimal information from which any name-location mapping process can be carried out ([4, 5]), but not necessarily sufficient. Some hosts cannot be located or their advertised location in the DNS registers could be wrong [see Zhang]. The next algorithms were devised to refine and correct the initial DNS-based mapping using a verified IP address to location list.

3.2 Host Clustering (HC)

```
    HC Algorithm
1   APList = (Aps from BGP)
2   CP1 = threshold for first phase
3   CP2 = threshold for second phase
4   IP2LocList = pairs (IP,Loc) list. Verified.
5   AP2LocList = MainBlock(APList,IP2LocList,CP1)
6   IP2LocList += (IP,Loc) pairs form INLM
7   AP2LocList = MainBlock(APList, IP2LocList,CP2)
8   for each (AP,Loc,n) in AP2LocList
9         for each IP not localized
10              if IP in AP && n >= CP2
11                    IP.Loc = AP.Loc
12              endif
13        endfor
14  endfor
15  return IP2LocList


16  MainBlock (APList, IP2LocList,n)
17  n is the cluster threshold
18  foreach IP in IP2LocList
19        AP = AP in APList | IP in AP
20        add (IP,Loc,AP) to IPAPLocList
21  endfor
22  cAP = first AP in IPAPLocList
23  foreach element e(IP,Loc,AP) in IPAPLocList
24        if (AP in e(IP,Loc,AP) == cAP)
25              add (IP,Loc) to SameAPList
26        else
27              if |SameAPList| > n
28                    curLoc = first Loc in e;
29                    last_e =  first element;
30                    foreach element e(IP,Loc) in SameAPList
31                          if Loc in e(IP,Loc) != curLoc
32                                newIP = (IP in e – IP in last_e)/2
33                                newAP = AP(first IP in cAP,newIP)
34                                add (newAP,curLoc) to AP2LocList
35                          endif
36                    endfor
37              else
38                    add (AP,Loc,n) to AP2LocList
39              endif
40        endif
41        cAP = AP
42  endfor
43  end MainBlock
```

Fig. 3. Pseudocode of the HC algorithm.

Once we have mapped the set of intermediate routers by using their domain name only, the next step is a clustering process based on a verified host IP list to further refine this first approximation in the case of errors. The HC refinement is important to further partition those groups of routers with the same domain name but physically far apart, or with different domain name (belonging to another location) but located in the same region. HC uses a verified address-location list, named vIP2Loc list, which is the HPL list with additional end-nodes (clients) IP addresses and location information (it is a superset of HPL containing a verified mapping of IP addresses to locations).

The clustering algorithm takes as input the Address Prefix (AP) information contained in the BGP tables and the vIP2Loc list to map intermediate routers (obtained by INLM) to locations. An AP is an aggregate of IP addresses that belong to an autonomous system, and is recognized by a starting IP address and a number denoting the range of numbers from it. For example the address prefix 4.0.0.0/16 denotes the 65536 IP numbers starting from 4.0.0.0 and ending on 4.0.255.255. The knowledge of APs enables us to identify clusters; which with high probability constitute geographic clusters, as observed in [1]. Different than current schemes, we additionally partition each AP (since APs can be widespread over the region of interest) for a more detailed host-location analysis as much as the gathered information permits.

The pseudocode of the HC algorithm is presented in Figure 3. It includes a subroutine called MainBlock responsible for partitioning and locating the AP blocks (main task of HC). MainBlock is divided into two steps. In the first step (lines 18-21), an Address Prefix to Location list is created by simply analysing the IP addresses in the IP2Loc list using a Consensus Parameter *CP* (third argument of MainBlock). *CP* represents the minimum cardinality (in terms of IPs probed) an AP must have to be partitioned. This value is proportional to the total number of IPs to be processed. In the second step (starting at line 23), every single AP found in step one is partitioned (if greater than its corresponding CP value) into *m* regions with different estimated locations (that is, IP addresses with the same AP can be located in different regions). The borderlines of the *m* partitions are obtained by taking the average between those IPs producing the minimum magnitude distance between the regions. For example, consider a two-region AP with corresponding values *region1* ={ xxx.xxx.xxx.7, xxx.xxx.xxx.14} and *region2* = {xxx.xxx.xxx.20, xxx.xxx.xxx.24, xxx.xxx.xxx.28}; the borderline between these regions becomes xxx.xxx.xxx.17. A node with IP address xxx.xxx.xxx.15 is set to *region1*.

If a single IP address is the only reference for a given AP then the location of this address is directly assigned to the AP. MainBlock returns an AP to location list (AP2LocList), which represents a list of clusters of IPs that may share a common location.

Initially the HC algorithm uses the verified IP to location list (vIP2loc) in the first call to MainBlock to do an initial selection of APs, and then uses vIP2Loc plus the INML output list in the second call to make the partitions. Each call uses different *CP* values *CP1* and *CP2* for the initial processing of the vIP2loc list and for the processing of the rest of the IPs probed respectively. Given that our confidence in the vIP2loc list is strong (because is a verified mapping list), the value of *CP1* must be small in order to maximize the number of segments in the AP. We propose any positive value of *CP1* in the range *[2, CP2 = 5]*. If the cardinality of the resulted AP in the second call to MainBlock (after partitioning) is ≥*CP2*, then we have enough number of hosts which agree on the same location, so it is more likely that all the addresses in the AP share a common location.

The final step in the HC algorithm (lines 8-14) is the mapping of unlocated IPs in the AP2Loc list by using the new partitions of the APs. The location given by the HC algorithm is not final, i.e., it can be refined by using delay values as shown in the next section.

3.3 Topology and Delay Location (IPDelay)

For those hosts whose location is still unknown after the INLM and HC steps, we gather additional host information such as Round-Trip Time (RTT) and host connectivity (topology) from each landmark. Each unlocated IP address is mapped to the closest location reported by the landmarks, in terms of their RTT and topological information. Mathematically, this can be expressed in the following way. Let $G = (V,E)$ be a graph where V is the set of nodes v_i and E is the set of edges $e_{ij} \in E$ connecting the nodes v_i and v_j. This graph G is constructed using a new topology connection from traceroute paths reported from each landmark.

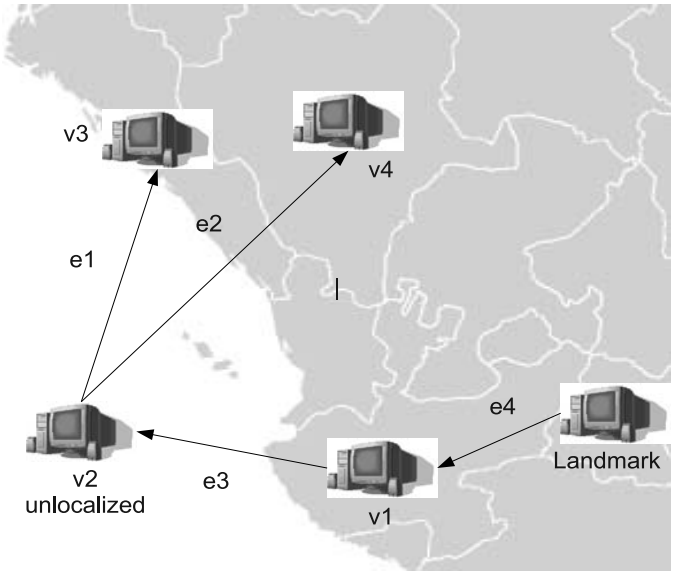


Fig. 4. Representation of the IPDelay scheme. The unlocated node v_2 is assigned to the localization of the closest node with respect to the minimum temporal distance between e_1, e_2 and e_3 edges.

Each edge $e_{ij} \in E$ has a weight $w(i,j)$ associated with the mean RTT reported by traceroute measurements. Let $F \subseteq E$ be the subset of edges connecting a fixed given unlocated node v_i. Then,

$$v_i.Loc = v_k.Loc$$

with

$$e_{ik} = \min_{w(i,j)} \{(e_{ij} \cup e_{ji}) \in F\}$$

If we assume the relationship between delay and distance as proportional (at some degree), then a previously not located node v_i has more probability to be located near or in the region represented by the node v_j. An example of IPDelay is shown in Figure 4, where v_2 represents the unlocated node after steps INML and HC. After the landmark probing, v_2 is assigned to that location with the minimum edge magnitude e; if the arrows are proportional to the magnitude, v_2 will be assigned to the same location of node v_1). Of course, the RTT values between v_2 to v_1-v_3 are affected by the amount of traffic present on each link (the probing phase must avoid peak traffic hours).

Other schemes in the literature using delay information [1, 4, and 10] make complete correspondence between delay and geographic distance, assigning geographic location between nodes by road map information. The proposed scheme does not imply any correspondence in this sense, so it does not infer non-useful information to users (i.e.,"20.1Km from B city", instead we conclude "in the zone of B city"). After the host localization delay refinement, HC is again executed in order to refine the APs partitions and location of the IPs. This is needed because new located IPs are aggregated to the IP2Loc list. HC and IPDelay can be iterated until all IPs' location remains unmodified.

3.3 IP Distance Similarity Location (IPDS)

An IP node serving as a router may have an alias or may share two different interfaces. This can be detected using a simple similarity algorithm. In our scheme, an IP address is said to be similar to another if the difference between them is less than 5 units. The addresses 200.52.176.30 and 200.52.176.35 had a similarity index of 5 (the difference between these two IPs is 5 address units); therefore the probability to be in the same geographic location (even in the same building) is high. The similarity value of 5 is chosen following [1] and [6], where they report that there are a high number of virtual router addresses with a mean distance of 3 to 4 units. Another reason is that, in general ISPs assign numerically close IPs to the access points or gateways.

We make a single pass of this simple IPDS algorithm to the entire IP2Loc list in order to locate those unclassified routers from previous steps.

IV. DATA SET AND RESULTS

4.1 Data Set

The approach described in previous section is applied for the discovering the Mexican IP topology. Four traceroute landmarks and a 64-node HPL table with known IP address to geographic location mapping were used in the defined region of interest (at least one node was chosen to be present in each Mexican state). As previously stated, the probing landmarks do not necessarily need to be spread all over the target region. In our case, we located two landmarks in the same city.

The IP addresses in the HPL table were chosen from the IET-DPEC database system to cover the entire region of interest. The IET-DPEC database provides information (up to the year 2006) of public schools enrolled in national educational projects. The information is organized by school name, address, teachers information, number of students enrolled in the project and host information (IP address and domain name). From the verified IP list, 38 IPs (60%) were used in the mapping process (to form the vIP2loc list in the HC scheme), and the rest of the IPs were used in the verification process (IPDelay scheme). We also use publicly available Mexican airport codes and well-known state and city acronyms information for the initial phase (INLM step).

For the BGP tables we use the snapshot from Routeviews, which is publicly available [13]. Routeviews is an initiative to provide the research community with weekly or monthly snapshots of the complete BGP tables shared by AP frontier routers. The BGP tables are filtered out to keep only the available and valid prefixes.

4.2 Results

The proposed system was implemented in C/C++ programming language, on a 750Mhz Pentium III class processor with 256MB in RAM running Linux 2.6.18-53.1.13.el5 operating system. The execution time of the experiment was divided into 2 parts: a) the time spent on the Restricted Probing (traceroute probing and BGP tables parsing, described in section 3.1), and the time needed for the geographical location of the nodes found during the probing phase (first part). The first part of the execution time lasted 2 hrs for our chosen level of detail, and can be executed once every 15-30 days. The second part is much faster; it takes up to 10 sec to geolocate the complete set of nodes (2 sec on a 2.4Ghz Xeon processor PC with 1GB of RAM). Once the topology has been defined, it takes the order of milliseconds (1-10) to geolocate a new input node. With this fast response time (1-10 ms), a Content Delivery Network (CDN) can select in real-time the closest server to the client's geographic location to be in charge of delivering the requested multimedia content.

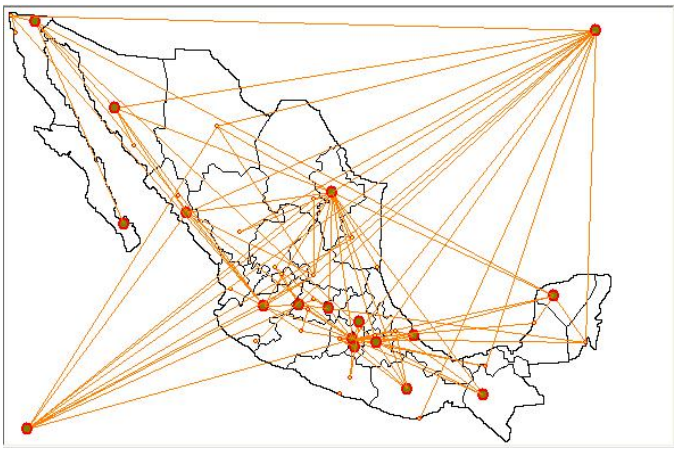


Fig. 5. IP topology for the Mexican Territory.

Figure 5, shows the resulting topology of the proposed scheme for the Mexican territory. The big dots correspond to inferred backbone nodes, while the small dots correspond to cities with an assigned node(s). The upper right dot corresponds to the IP addresses located outside the country, and the lower left dot corresponds to unlocated IP addresses after processing the entire dataset. For the 4 landmark nodes and the 64 end-nodes used in the test, we found 228 different routes, 589 total routers (after removing duplicates and private addresses), from which 430 routers are in the Mexican territory, and 118 are located outside the country, but used for regional traffic. We found 139 routers (out of the 589) advertised as backbones. 1 summarizes the results of the mapping process. The error percentage, which represents the fraction of addresses incorrectly located, is less than 1%. The unlocated hosts dropped ~10% when using the complete set of algorithms, which is quite good giving that we have such a small set of addresses. For the mapping of new hosts, the error percentage is around 30% with 4 landmarks (same percentage than more complex schemes), and decreases as low as 20% when 5 landmarks are used. The error percentage for geolocating new hosts may appear to be high at first sight, but in fact it is not. We consider an error, when a host A is reported to be in city or state X when actually it is in city or state Y (for X and Y contiguous cities or states), even though the real geographic position of node A is too close of the reported location (we do not set a geographical radius for which our mapping report must fall in or out, in order to be considered right or wrong respectively). For the kind of

applications our scheme is aimed to (efficient delivery of multimedia content in Content Delivery Networks--CDN), the localization error among contiguous states is not important since they may be only few seconds apart in delay units. We obtained an average distance error between wrong located nodes <15ms, which is minuscule.

The topology drawn by our scheme was compared with the one obtained in [11]. Our scheme correctly found the backbone location. For a final and more critical evaluation, we ran the publicly available reverse tree routine called Scriptroute based on Rocketfuel [12], in order to compare the location of the connections between hosts in México and hosts in the United States. Rocketfuel is an ISP topology mapping engine, which takes into account around 10 ISPs, in Europe, Australia, and the United States using around 300 traceroute web servers. A database is then constructed of over 50 thousand IP addresses representing 45 thousand routers in 537 POPs connected by 80 thousand links. Our scheme, with only 38 nodes in the area of interest correctly located the backbone connection points reported by Scriptroute in about 2 hrs without going through any ISP analysis. A valuable asset of our scheme, in addition to its fast response and accuracy is its independency of any ISP topological analysis in the region of interest. We would like to point out as well, that despite of the small database use in our experiment, our error is equivalent to Scriptroute using thousand of hosts.

V. CONCLUSIONS

This paper is a first attempt to provide fast, simple, effective, and ISP independent scheme for solving the IP to location problem over the public Internet. We proved that with a simple scheme and a small input dataset (38 distinct IP addresses) it is possible to get a very good approximation of the Internet topology in a specified region of interest, for the mapping of new hosts. Our scheme is independent of any ISP topological analysis, and its mapping accuracy is scalable according to the number of nodes in the initial database table.

REFERENCES

[1] V. N. Padmanabhan, L. Subramanian, "An investigation of geographic mapping techniques for internet hosts," in SIGCOMM '01: Proceedings of the 2001 Conference on Applications, technologies, architectures, and protocols for computer communications. ACM Press, New York, NY, USA, pp. 173–185.

[2] J. Apostolopoulos, "Reliable video communication over lossy packet networks using multiple state encoding and path diversity," in Proceedings of the Visual Communication and Image Processing, VCIP 2001, pp. 392–409.

[3] Inc., M. (January 2008), *Geoip*. Available: www.maxmind.com.

[4] R. Govindan, H. Tangmunarunki, "Heuristics for internet map discovery," in IEEE INFOCOM 2000.

[5] N. Spring, Mahajan, R., D. Wetherall, T. Anderson, "Measuring ISP topologies with rocketfuel," in IEEE/ACM Transactions on Networking 2004, 12 (1), pp.2–16.

[6] R. Teixeira, K. Marzullo, S. Savage, G. M. Voelker, "In search of path diversity in isp networks," in Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement IMC '2003. ACM Press, New York, NY, USA, pp. 313–318.

[7] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, L. Zhang, "Idmaps: a global internet host distance estimation service," in IEEE/ACM Transactions on Networking, 2001 9 (5), 525–540.

[8] F. Dabek, R. Cox, F. Kaashoek, R. Morris "Vivaldi: a decentralized network coordinate system," in SIGCOMM '04: Proceedings of the 2004 Conference on Applications, technologies, architectures, and protocols for Computer Communications. ACM Press, New York, NY, USA, pp. 15–26.

[9] E. Katz-Bassett, J.P. John, A. Krishnamurthy, D. Wetherall, T. Anderson, Y. Chawathe, "Towards ip geolocation using delay and topology measurements," in IMC '06: Proceedings of the 6th ACM SIGCOMM on Internet measurement. ACM Press, New York, NY, USA, 2006, pp. 71–84.

[10] B. Gueye, A. Ziviani, M. Crovella, S. Fdida, "Constraint-based geolocation of internet hosts," in IEEE/ACM Transactions on Networking, 14 (6), 2006, pp.1219–1232.

[11] J. Tomasson,, W. Foster, L. Press, "The diffusion of the internet in México," Tech. Report., in Latin American Network Information Center, University of Texas, 2002.

[12] D. Meyer (2004), Routeviews project. Available: http://www.routeviews.org.

[13] University of Washington (2005), "Rocketfuel maps and data". Avalilable: http://www.cs.washington.edu/research/networking/rocketfuel.

[14] A-M. Khan-Pathan, Utility-Oriented Internetworking of Content Delivery Networks. *Department of Computer Science and Software Engineering*, The University of Melbourne, Australia. Ph.D Thesis, 2009. http://www.cloudbus.org/students/MukaddimPhDThesis2009.pdf

[15] M. Zhang, Y. Ruan, V. Pia, J. Rexford, How DNS Misnaming Distorts Internet Topology Mapping. *Proceedings of the annual conference on USENIX '06 Annual Technical Conference*, 2006.

TABLE I. Percentage results obtained by the application of the different algorithms. Basic implies the use of INLM and HC phases only.

	Basic (%)	Basic + IPDelay (%)	Basic + IPDS (%)	Full Scheme (%)
Located In Region	65.42	65.57	70.4	71.65
Located Out of Region	16.97	18.32	21.65	21.5
Unlocated	17.6	16.04	7.94	6.85
Wrongly Located	0.5	0.5	0.6	0.7

M.C. Edmar Mota-Garcia received his bachelor degree in Civil Engineering from the Department of Engineering-Autonomous University of Sinaloa, México in 1999 and the M.S. in computer science and industrial mathematics from the Center for Research in Mathematics (CIMAT) in Guanajuato, México in 2002. He is Ph.D. candidate from CIMAT.

He joined the Department of Education Technology of the Education and Culture Ministry of Sinaloa, México (SEPyC), where he leads the applications of Robotics in Middle Education Programs and the Local State wide Internet platform of Digital Abilities for Everyone (HDT). He is part of Sinaloa State Council for Research and Technology (CECyT), which provides master degree courses on Multimedia, Technology in Education and Problem Solving.

Rogelio Hasimoto-Beltran received his M.S. in Computer Science from the Scientific Research Center and Higher Education of Ensenada (CICESE), México, in 1990, and Ph.D. from the University of Delaware, USA, in 2001. After his Ph.D., he spent two years as a Senior Software Engineer in the Streaming Group at Akamai Technologies in San Mateo, CA, USA. Since 2003 he serves as a Research Fellow in the Department of Computer Science at the Center for Research in Mathematics (CIMAT), México, conducting research on image/video processing and compression, robust multimedia transmission, biometric systems and multimedia encryption based on chaos. During August/2009-July/2010 he was Visiting Associate Professor in the Department of electrical and Computer Engineering at University of Illinois in Chicago (UIC). Dr. Hasimoto is member of the IEEE association.

On Fixed Point Theorems in Fuzzy Metric Spaces

Rajesh Kumar Mishra[†], Sanjay Choudhary[‡]

***Abstract*: This paper presents some common fixed point theorems for occasionally weakly compatible mappings in fuzzy metric spaces under various conditions.**

***Index Terms*: Occasionally weakly compatible mappings, fuzzy metric space.**

I. INTRODUCTION

Fuzzy set was defined by Zadeh [24]. Kramosil and Michalek [12] introduced fuzzy metric space, George and Veermani [4] modified the notion of fuzzy metric spaces with the help of continuous t-norms. Many researchers have obtained common fixed point theorems for mappings satisfying different types of commutativity conditions. Vasuki [23] proved fixed point theorems for R weakly commutating mappings. Pant [16, 17,18] introduced the new concept reciprocally continuous mappings and established some common fixed point theorems. Balasubramaniam et al.[2], have shown that Rhoades [20] open problem on the existence of contractive definition which generates a fixed point but does not force the mappings to be continuous at the fixed point, posses an affirmative answer. Pant and Jha [18] obtained some analogous results proved by Balasubramaniam et al. Recent literature on fixed point in fuzzy metric space can be viewed in [7, 14, 22]. This paper presents some common fixed point theorems for more general commutative condition i.e. occasionally weakly compatible mappings in fuzzy metric space. Before giving the results, some preliminary definitions are given below.

Definition 1.1 [24] A fuzzy set A in X is a function with domain X and values in [0, 1]

Definition 1.2 [21] A binary operation * : [0, 1] × [0, 1] → [0, 1] is a continuous t-norms if * is satisfying conditions:

(i) * is an commutative and associative;

(ii) * is continuous;

(iii) a * 1 = a for all a ∈ [0, 1];

(iv) a * b ≤ c * d whenever a ≤ c and b ≤ d, and a, b, c, d ∈ [0, 1].

Definition 1.3 [4] A 3-tuple (X,M, *) is said to be a fuzzy metric space if X is an arbitrary set, * is a continuous t-norm and M is a fuzzy set on $X^2 \times [0, \infty]$ satisfying the following conditions, for all x, y, z ∈ X, such that t is in [0, ∞]

(f1) M(x, y, t) > 0;

(f2) M(x, y, t) = 1 if and only if x = y

(f3) M(x, y, t) = M(y, x, t);

(f4) M(x, y, t) *M(y, z, s) ≤M(x, z, t + s);

(f5) M(x, y, *) : (0,∞) → (0, 1] is continuous.

Then M is called a fuzzy metric on X. Then

M(x, y, t) denotes the degree of nearness between x and y with respect to t.

Definition 1.4 [4]: Let (X,M, *) be a fuzzy metric space. Then

(a) a sequence $\{x_n\}$ in X is said to converges to x in X if for each ε > 0 and each t > 0, there exists $n_0 \in N$ such that M(xn, x, t) > 1 −ε for all $n \geq n_0$.

(b) a sequence $\{x_n\}$ in X is said to be Cauchy if for each ε > 0 and each t > 0,there exists $n_0 \in N$ such that $M(x_n, x_m, t) > 1 - \varepsilon$, ∀ n, m ≥ n_0.

(c) A fuzzy metric space in which every Cauchy sequence is convergent is said to be complete.

Definition 1.5 [23] A pair of self-mappings (f, g) of a fuzzy metric space (X,M, *) is said to be

(i) weakly commuting if

M(fgx, gfx, t) ≥ M(fx, gx, t) ∀ x ∈ X & t > 0.

(ii) R-weakly commuting if there exists some R > 0 such that M(fgx, gfx,t) ≥ M(fx, gx, t/R) ∀x ∈ X and t > 0.

Definition 1.6 [11] Two self mappings f and g of a fuzzy metric space (X,M, *) are called compatible if

$\lim_{n\to\infty} M(fgx_n, gfx_n, t) = 1$

whenever $\{x_n\}$ is a sequence in X such that

$\lim_{n\to\infty} fx_n = \lim_{n\to\infty} g x_n = x$ for some x in X.

Definition 1.7 [5]: Two self maps f and g of a fuzzy metric space (X,M, *) are called reciprocally continuous on X if

$\lim_{n\to\infty} fgx_n = fx$ and $\lim_{n\to\infty} gfx_n = gx$

whenever $\{x_n\}$ is a sequence in X such that

$\lim_{n\to\infty} fx_n = \lim_{n\to\infty} gx_n = x$ for some x in X.

Definition 1.8 Let X be a set, f, g selfmaps of X. A point x in X is called a coincidence point of f and g iff fx = gx. We shall call w = fx = gx a point of coincidence of f and g.

Definition 1.9 [11] A pair of maps S and T is called weakly compatible pair if they commute at coincidence points.

Definition 1.10 Two self maps f and g of a set X are occasionally weakly compatible (owc) iff there is a point x

Manuscript received October 31, 2010.

[‡]Rajesh Kumar Mishra is with the Bhabha Engineering Research Institute, Bhopal, India. (corresponding author e-mail: ptrajeshmishra@gmail.com).

[†]Sanjay Choudhary is with the Govt. Narmada P.G.College, Hoshangabad, Madhya Pradesh, India.

in X which is a coincidence point of f and g at which f and g commute.

A. Al-Thagafi and Naseer Shahzad [1] have shown that occasionally weakly is weakly compatible but converse is not true.

Example 1.11 [1] Let R be the usual metric space. Define S, T : R → R by Sx = 3x and Tx = x^2 for all x ∈ R. Then Sx = Tx for x = 0, 3 but ST0 = TS0, and ST3 ≠ TS3. S and T are occasionally weakly compatible self maps but not weakly compatible.

Lemma 1.12 [10] Let X be a set, f, g owc self maps of X. If f and g have a unique point of coincidence, w = fx = gx, then w is the unique common fixed point of f and g.

Main Results

Theorem 2.1 Let (X,M, *) be a complete fuzzy metric space and let A,B, S and T be self-mappings of X. Let the pairs {A, S} and {B, T} be owc. If there exists q ∈ (0,1) such that

M(Ax,By,qt)≥min{M(Sx,Ty,t), M(Sx,Ax,t), M(By,Ty,t), [M(Ax,Ty,t)+M(By,Sx,t)]/2}……..……(1)

for all x, y ∈ X and for all t > 0, then there exists a unique point w ∈ X

such that Aw = Sw = w and a unique point z ∈ X such that Bz = Tz = z. Moreover, z = w, so that there is a unique common fixed point of A,B, S and T.

Proof 2.1: Let the pairs {A, S} and {B, T} be owc, so there are points x, y ∈ X

such that Ax = Sx and By = Ty. We claim that Ax = By. If not, by inequality (1)

M(Ax,By, qt) ≥ min{M(Sx, Ty, t), M(Sx, Ax, t),M(By, Ty, t), [M(Ax, Ty, t)+M(By, Sx, t)]/2}
= min{M(Ax,By,t),M(Ax,Ax,t), M(By,By,t), ,[M(Ax,By, t)+M(By, Ax, t)]/2}
= M(Ax,By, t).

Therefore Ax = By, i.e. Ax = Sx = By = Ty.

Suppose that there is a another point z such that Az = Sz then by (1) we have Az = Sz = By = Ty, so Ax = Az and w = Ax = Sx is the unique point of coincidence of A and S.

By Lemma 1.12: w is the only common fixed point of A and S. Similarly there is a unique point z ∈X such that z = Bz = Tz. Assume that w ≠ z. We have

M(w,z, qt) =
M(Aw,Bz,qt)≥ min{M(Sw,Tz,t),M(Sw,Az,t), M(Bz,Tz,t), [M(Aw,Tz,t)+M(Bz,Sw, t)]/2}
=min{M(w, z, t),M(w, z, t), M(z, z, t), [M(w, z, t)+M(z,w, t)]/2}
= M(w, z, t)

Therefore we have z = w by Lemma 1.12 and z is a common fixed point of A,B, S and T. The uniqueness of the fixed point holds from (1).

Theorem 2.2 let (X,M, *) be a complete fuzzy metric space and let A,B, S and T be self-mappings of X. Let the pairs {A, S} and {B, T} be owc. If there exists q ∈ (0, 1) such that

M(Ax,By,qt) ≥ Ψ(min{M(Sx,Ty,t),M(Sx,Ax,t), M(By, Ty, t), [M(Ax,Ty,t)+M(By,Sx,t)/2}…….(2)

for all x, y ∈ X and Ψ : [0, 1] →[0, 1] such that Ψ (t) > t for all 0 < t < 1, then there exists a unique common fixed point of A, B, S and T.

Proof 2.2: The proof follows from Theorem 2.1.

Theorem 2.3 Let (X,M, *) be a complete fuzzy metric space and let A,B, S and T be self-mappings of X. Let the pairs {A, S} and {B, T} be owc. If If there exists q ∈ (0, 1) such that

M(Ax,By,qt)≥Ψ{ M(Sx,Ty,t), M(Sx,Ax,t), M(By,Ty,t), [M(Ax,Ty,t)+M(By,Sx,t)]/2 , M(By,Sx, t)}……(3)

for all x, y ∈ X and Ψ: $[0, 1]^5$ → [0, 1] such that Ψ (t, 1, 1, t, t) > t for all 0 < t < 1, then there exists a unique common fixed point of A,B, S and T.

Proof: Let the pairs {A, S} and {B, T} are owc, there are points x, y ∈ X such that Ax = Sx and By = Ty. We claim that Ax = By. By inequality (3) we have

M(Ax,By,qt) ≥ Ψ{M(Sx,Ty,t), M(Sx,Ax,t), M(By,Ty,t), [M(Ax,Ty,t)+M(By,Sx,t)]/2 , M(By,Sx, t)}
= Ψ(M(Ax,By,t), M(Ax,Ax,t), M(By,By,t), [M(Ax,By,t)+M(By, Ax, t)]/2, M(By, Ax, t)}
= Ψ{(M(Ax, y,t),1,1,M(Ax,By, t)}
> M(Ax,By, t).

a contradiction, therefore Ax = By, i.e. Ax = Sx = By = Ty. Suppose that there is a another point z such that Az = Sz then by (3) we have Az = Sz = = Ty, so Ax = Az and w = Ax = Tx is the unique point of coincidence of A and T. By Lemma 1.12 w is a unique common fixed point of A and S. Similarly there is a unique point z ∈ X such that z = Bz = Tz. Thus z is a common fixed point of A, B, S and T. The uniqueness of the fixed point holds from (3).

REFERENCES

[1] A. Al-Thagafi and Naseer Shahzad, Generalized I- Non expansive Selfmaps And Invariant Approximations, Acta Mathematica Sinica, English Series May, 2008, Vol. 24, No. 5, pp. 867876.

[2] P. Balasubramaniam, S. Muralisankar, R.P. Pant, "Common fixed points of four mappings in a fuzzy metric space", J. Fuzzy Math. 10(2) (2002),379-384.

[3] Y.J. Cho, H.K. Pathak, S.M. Kang, J.S. Jung, "Common fixed points of compatible maps of type (A) on fuzzy metric spaces", Fuzzy Sets and Systems 93 (1998), 99-111.

[4] A. George, P. Veeramani, "On some results in fuzzy metric spaces", Fuzzy Sets and Systems, 64 (1994), 395-399.

[5] M.Grabiec,"Fixed points in fuzzy metric spaces",Fuzzy Sets and Systems 27(1988), 385-389.

[6]O.Hadzic, "Common fixed point theorems for families of mapping in complete metric space", Math. Japon. 29 (1984), 127-134.

[7] Mohd. Imdad and Javid Ali, "Some common fixed point theorems in fuzzy metric spaces", Mathematical Communications 11(2006), 153-163

[8] G. Jungck, "Compatible mappings and common fixed points (2)", Internat. J. Math. Math. Sci. (1988), 285-288.

[9] G. Jungck and B. E. Rhoades, "Fixed Point for Set Valued functions without Continuity", Indian J. Pure Appl. Math., 29(3), (1998), pp.771-779.

[11] G. Jungck and B. E. Rhoades, "Fixed Point Theorems for Occasionally Weakly compatible Mappings", Fixed Point Theory, Volume 7, No. 2,2006, 287-296.

[11] G. Jungck and B. E. Rhoades, "Fixed Point Theorems for Occasionally Weakly compatible Mappings", Erratum, Fixed Point Theory, Volume 9,No. 1, 2008, 383-384.

[12] O. Kramosil and J.Michalek, "Fuzzy metric and statistical metric spaces",Kybernetika, 11 (1975), 326-334.

[14] S. Kutukcu, ”A fixed point theorem for contraction type mappings in Menger spaces”, Am.J.Appl.Sci.4(6) (2007),371-373.
[15] Servet Kutukcu, Sushil Sharma1 and Hanifi Tokgoz, ”A Fixed Point Theorem in Fuzzy Metric Spaces”, Int. Journal of Math. Analysis, Vol. 1,2007, no. 18, 861 - 872.
[16] S.N. Mishra, ”Common fixed points of compatible mappings in PM spaces”,Math. Japon. 36 (1991), 283-289.
[17] R.P. Pant, ”Common fixed points of four mappings”, Bull. Cal. Math.Soc. 90 (1998), 281-286.
[18]R.P. Pant, ”Common fixed point theorems for contractive maps”, J. Math.Anal. Appl. 226 (1998), 251-258.
[21] R.P. Pant, K. Jha, ”A remark on common fixed points of four mappings in a fuzzy metric space”, J. Fuzzy Math. 12(2) (2004), 433-437.
[22] H. K.Pathak and Prachi Singh, ”Common Fixed Point Theorem for Weakly Compatible Mapping ”, International Mathematical Forum, 2,2007, no. 57, 2831 - 2839.
[23] B.E. Rhoades, ”Contractive definitions and continuity”, Contemporary Math. 72 (1988), 233-245.
[24] B. Schweizer and A. Sklar, ”Statistical metric spaces”, Pacific J. Math.10(1960), 313-334.
[25] Seong Hoon Cho, ”On common fixed point in fuzzy metric space”, Int.Math. Forum, 1, 2006, 10, 471-479.
[26] R. Vasuki, Common fixed points for R-weakly commuting maps in fuzzy metric spaces, Indian J. Pure Appl. Math. 30 (1999), 419-423.
[27] L.A. Zadeh, Fuzzy sets, Inform and Control 8 (1965), 338-353.

Rajesh Kumar Mishra, Assist. Prof & Head of Department of Engineering Mathematics, Bhabha Engineering Research Institute, Bhopal. He did his graduation in Mathematics from Barkatullah University, Bhopal and Post Graduation from same university in 1986. He also did M.Phil from Madurai Kamraj University in year 2007. His field of interest is Non linear analysis and Topology.

Dr. Sanjay Choudhary, Head of Department of Mathematics, Govt. Narmada P.G.College, Hoshangabad, Madhya Pradesh, India. He did Post graduation in year 1984 from Barkatullah University, BHOPAL (MP) India. He completed his research work in field of Non linear analysis and awarded Ph.D from Barkatullah University, BHOPAL (MP) India in year 2003.

International Magazine on Advances in Computer Science and Telecommunications

Published by the Association for Computer Science and Telecommunications AKOMNAT TEL DOO
IMACST 1 (1) 1-47 (2010) ISSN 1857-7202
www.imacst.com