Radonjic, A., 2018. (Perfect) Integer Codes Correcting Single Errors. IEEE Communications Letters 22, 17–20. https://doi.org/10.1109/LCOMM.2017.2757465

(Perfect) Integer Codes Correcting Single Errors

Aleksandar Radonjic[1]

Abstract: This letter presents a class of integer codes capable of correcting single errors. Unlike Hamming codes, the presented codes are constructed with the help of a computer. Among all codes of length up to 4096 bits, a computer search has found four perfect codes: (15, 10), (63, 56), (1023, 1012), and (4095, 4082). In addition, it is shown that, for practical data lengths up to 4096 bits, the proposed codes require only one check bit more compared to Hamming codes.

Keywords: Integer codes, error correction, single errors, perfect codes.

I. Introduction

Codes correcting single errors always drew attention of coding theoreticians. The reason for this mostly lies in the fact that the first single error correcting (SEC) codes (Hamming codes) were also perfect [1]. Thus, it was very challenging to construct codes with rate better than $R = (2^u - u - 1)/(2^u - 1)$, where $u \geq 3$.

However, even after 67 years, researchers have not found codes with lower redundancy than the Hamming ones (see [2], [3] and references therein). Moreover, almost all proposed codes were either more complicated than the Hamming codes or required more check-bits. Among the rare exceptions were the Varshamov-Tenengolt's (VT) codes constructed in the mid 1960s [4]. Although these codes did not have the ability to correct symmetric errors, they were easier to encode/decode than the Hamming codes. Thanks to this feature, the VT codes were generalized by several authors [5]–[7], including Vinck and Morita [8]. In their paper, these authors developed the concept of coding over the ring Z_m of integers modulo m. The key idea of this concept was to construct codes capable of correcting single errors of specific types (single peak-shifts, single cross errors, single square errors, etc.). Such codes were subsequently suggested for use in almost all applications at the physical layer except those related to channel coding (see [9] and references therein).

Motivated by this fact, this letter presents a class of integer codes capable of correcting single errors within a codeword. The presented codes belong to the family of integer error control codes (IECCs), which means that they share many similarities with the codes proposed in [10]–[13]. However, unlike these codes, the presented codes are very efficient in terms of redundancy. In addition, for some rates, they are also perfect. Both these features make them unique among IECCs, and consequently, the most interesting from a theoretical point of view.

[1] Institute of Technical Sciences, Serbian Academy of Sciences and Arts, Beograd 11000, Serbia (e-mail: sasa_radonjic@yahoo.com).

II. Integer SEC Codes

A. Codes Construction

As already mentioned, the concept of integer codes is developed by Vinck and Morita in the late 1990s. According to their definition, an integer code $C(d, \omega)$ is defined by

$$C(d,\ \omega) = \left\{ v \in Z_m^n : \sum_{i=1}^{n} \omega_i \cdot v_i \equiv d \ (\mathrm{mod}\ m) \right\} \qquad (1)$$

where $v = (v_1, v_2, \ldots, v_n) \in Z_m^n$ is the codeword vector, $\omega = (\omega_1, \omega_2, \ldots, \omega_n) \in Z_m^n$ is a fixed-weight coefficient vector and $d \in Z_m$ is a fixed integer. So, n is the length of the code and m is the size of the code alphabet. The concept of IECCs is not so general. This can be seen from the following definition.

Definition 1: Let $Z_{2^b-1} = \{0, 1, \ldots, 2^b - 2\}$ be the ring of integers modulo $2^b - 1$ and let $B_i = \sum_{n=0}^{b-1} a_n \cdot 2^n$ be the integer representation of a b-bit byte, where $a_n \in \{0, 1\}$ and $1 \leq i \leq k$. Then, the code C (b, k, c), defined as

$$C(b, k, c) = \left\{ x \in Z_{2^b-1}^{k+1} : \sum_{i=1}^{k} C_i \cdot B_i \equiv C_B \ (\mathrm{mod}\ 2^b - 1) \right\} \quad (2)$$

is an (kb + b, kb) IECC, where $x = (B_1, B_2, \ldots, B_k, C_B) \in Z_{2^b-1}^{k+1}$ is the codeword vector, $c = (C_1, C_2, \ldots, C_k) \in Z_{2^b-1}^{k}$ is a fixed-weight coefficient vector and $C_B \in Z_{2^b-1}$ is a fixed integer.

To understand this definition, suppose that a codeword $x = (B_1, B_2, \ldots, B_k, C_B) \in Z_{2^b-1}^{k+1}$ is sent through a noisy channel. Then, the received vector can be written in the form $y = (B_1, B_2, \ldots, B_k, C_B) = (B_1 \pm e_1, B_2 \pm e_2, \ldots, B_k \pm e_k, C_B \pm e_{k+1}) \in Z_{2^b-1}^{k+1}$, where $e = (e_1, e_2, \ldots, e_k, e_{k+1}) \in Z_{2^b-1}^{k+1}$ is the error vector. To identify this vector it is necessary to choose the coefficients $C_i \in Z_{2^b-1} \setminus \{0, 1\}$ in such a way that the syndrome S

$$S = \sum_{i=1}^{k} C_i \cdot B_i - C_B (\mathrm{mod}\ 2^b - 1) = \sum_{i=1}^{k+1} \pm e_i \cdot C_i (\mathrm{mod}\ 2^b - 1) \quad (3)$$

is unique, where $C_{k+1} = -1$. Bearing this in mind, we can state the following definition and theorem.

Definition 2: The set of syndromes corresponding to single errors is defined as

$$\xi_{b,k} = \left\{ \bigcup_{i=1}^{k+1} \left(\pm\ 2^r \cdot C_i \right) \ (\mathrm{mod}\, 2^b - 1) : 0 \leq r \leq b - 1 \right\} \quad (4)$$

Theorem 1: The codes defined by (2) can correct all single errors iff there exists k different coefficients $C_i \in Z_{2^b-1} \setminus \{0, 1\}$ such that

$$\left| \xi_{b,k} \right| = 2 \cdot b \cdot (k + 1),$$

where $\left| \xi_{b,k} \right|$ denotes the cardinality of $\xi_{b,k}$.

Proof: Observe that the set $\xi_{b,k}$ can be expressed as

$$\xi_{b,k} = \bigcup_{i=1}^{k+1} s_i$$

where

$$\begin{aligned} s_1 &= \left\{ \left(\pm\, 2^r \cdot C_1\right) \pmod{2^b - 1} : 0 \le r \le b-1 \right\} \\ &\vdots \\ s_k &= \left\{ \left(\pm\, 2^r \cdot C_k\right) \pmod{2^b - 1} : 0 \le r \le b-1 \right\} \\ s_{k+1} &= \left\{ \left(\mp\, 2^r\right) \pmod{2^b - 1} : 0 \le r \le b-1 \right\} \end{aligned}$$

From this it is easy to see that the syndromes caused by single errors will be nonzero and mutually different iff there exists k different coefficients $C_i \in Z_{2^b-1} \setminus \{0, 1\}$ such that

$$\begin{aligned} s_1 \cap \cdots \cap s_k \cap s_{k+1} &= \emptyset \\ |s_1| = \cdots = |s_k| &= |s_{k+1}| . \end{aligned}$$

In that case, the set $\xi_{b,k}$ will have

$$\begin{aligned} \left|\xi_{b,k}\right| &= |s_1| + \cdots + |s_k| + |s_{k+1}| \\ &= |s_{k+1}| \cdot (k+1) = 2 \cdot b \cdot (k+1) \end{aligned}$$

nonzero elements. □

Based on the above theorem, we can establish a condition for existence of a perfect code.

Theorem 2: An (kb + b, kb) integer SEC code is perfect iff for some b > 0 and k > 0 it holds that

$$k = \frac{2^{b-1} - b - 1}{b}.$$

Proof: From coding theory we know that a code is called perfect if it uses all available nonzero syndromes. In the case of IECCs, the number of available nonzero syndromes is equal to $2^b - 2$. Combining this with Theorem 1, we get the equality

$$\left|\xi_{b,k}\right| = 2 \cdot b \cdot (k+1) = 2^b - 2$$

wherefrom it follows that

$$k = \frac{2^{b-1} - b - 1}{b}.$$ □

Remark: From Theorem 2 we see that perfect integer SEC codes have a rate of $R = (2^{b-1} - b - 1)/(2^{b-1} - 1)$.

A computer search has shown that for smaller values of b the condition of Theorem 2 is not only necessary, but also sufficient. Namely, for each "perfect" value of b less than or equal to 13 ($b = 5, 7, 11$ and 13) there is exactly k coefficients $C_i \in Z_{2^b-1} \setminus \{0, 1\}$ such that $\left|\xi_{b,k}\right| = 2^b - 2$. These coefficients, listed in Table 1, allow us to construct four perfect codes: (15, 10), (63, 56), (1023, 1012) and (4095, 4082).

As far as "non-perfect" values of b are concerned ($b = 6$, 8, 9, 10 and 12), the number of coefficients $C_i \in Z_{2^b-1} \setminus \{0, 1\}$ varies between $\lfloor (2^{b-1} - b - 1)/b \rfloor - 5$ and $\lfloor (2^{b-1} - b - 1)/b \rfloor$ (Table 1). This means that, even in these cases, we can construct codes requiring only one check bit more compared to the Hamming ones. This result, obviosly, is quite better than that obtained by modifying Fletcher's code [14], [15] (Table 2) (the codes from [15] are also defined over the ring of integers modulo $2^b - 1$).

TABLE I

COEFFICIENTS FOR INTEGER SEC CODES WITH PARAMETERS $5 \le b \le 13$

$b = 5$											
3	5										
$b = 6$											
3	5	11									
$b = 7$											
3	5	7	9	11	13	19	21				
$b = 8$											
3	5	7	9	11	13	19	21	23	25	27	37
43											
$b = 9$											
3	5	7	9	11	13	15	17	19	21	23	25
27	29	35	37	39	41	43	45	51	53	55	75
77	83	85									
$b = 10$											
3	5	7	9	11	13	15	17	19	21	23	25
27	29	35	37	39	41	43	45	47	49	51	53
55	57	59	69	71	73	75	77	83	85	87	89
91	101	103	105	107	109	147	149	171	173	179	
$b = 11$											
3	5	7	9	11	13	15	17	19	21	23	25
27	29	31	33	35	37	39	41	43	45	47	49
51	53	55	57	59	61	67	69	71	73	75	77
79	81	83	85	87	89	91	93	99	101	103	105
107	109	111	113	115	117	119	137	139	141	147	149
151	153	155	157	163	165	167	169	171	173	179	181
183	185	199	201	203	205	211	213	215	217	219	293
299	301	307	309	331	333	339	341				
$b = 12$											
3	5	7	9	11	13	15	17	19	21	23	25
27	29	31	33	35	37	39	41	43	45	47	49
51	53	55	57	59	61	67	69	71	73	75	77
79	81	83	85	87	89	91	93	95	97	99	101
103	105	107	109	111	113	115	117	119	121	123	133
135	137	139	141	143	145	147	149	151	153	155	157
163	165	167	169	171	173	175	177	179	181	183	185
187	197	199	201	203	205	207	209	211	213	215	217
219	221	227	229	231	233	235	237	239	275	277	279
281	283	285	291	293	295	297	299	301	307	309	311
313	327	329	331	333	339	341	343	345	347	349	355
357	359	361	363	365	371	397	403	405	407	409	411
421	423	425	427	429	435	437	439	587	589	595	597
603	613	619	661	683	685	691	717				
$b = 13$											
3	5	7	9	11	13	15	17	19	21	23	25
27	29	31	33	35	37	39	41	43	45	47	49
51	53	55	57	59	61	63	65	67	69	71	73
75	77	79	81	83	85	87	89	91	93	95	97
99	101	103	105	107	109	111	113	115	117	119	121
123	125	131	133	135	137	139	141	143	145	147	149
151	153	155	157	159	161	163	165	167	169	171	173
175	177	179	181	183	185	187	189	195	197	199	201
203	205	207	209	211	213	215	217	219	221	223	225
227	229	231	233	235	237	239	241	243	245	247	265
267	269	271	273	275	277	279	281	283	285	291	293
295	297	299	301	303	305	307	309	311	313	315	317
323	325	327	329	331	333	335	337	339	341	343	345
347	349	355	357	359	361	363	365	367	369	371	373
375	377	391	393	395	397	399	401	403	405	407	409
411	413	419	421	423	425	427	429	431	433	435	437
439	441	443	453	455	457	459	461	463	465	467	469
471	473	475	477	547	549	551	553	555	557	563	565
567	569	571	581	583	585	587	589	595	597	599	601
603	605	611	613	615	617	619	621	627	629	651	653
659	661	663	665	667	669	675	677	679	681	683	685
691	693	695	697	711	713	715	717	723	725	727	729
731	733	739	741	743	793	795	805	807	809	811	813
819	821	823	825	839	841	843	845	851	853	855	857
859	869	871	873	875	877	1171	1173	1179	1189	1195	1197
1203	1205	1227	1229	1235	1237	1323	1325	1331	1333	1355	1357
1363	1365										

TABLE II

NUMBER OF CHECK-BITS FOR VARIOUS SEC CODES

Codes	Data word length (bits)									
	8	16	32	64	128	256	512	1024	2048	4096
Hamming codes	4	5	6	7	8	9	10	11	12	13
Proposed codes	5	6	7	8	9	10	11	12	13	14
Codes from [15]	6	6	8	10	10	10	12	12	14	14

TABLE III

THE SYNDROME TABLE FOR THE PERFECT (63, 56) INTEGER SEC CODE

	Element of the set $\xi_{7,8}$	i	e		Element of the set $\xi_{7,8}$	i	e		Element of the set $\xi_{7,8}$	i	e		Element of the set $\xi_{7,8}$	i	e		Element of the set $\xi_{7,8}$	i	e
1	1	9	1	**27**	27	7	32	**53**	53	8	64	**79**	79	1	16	**105**	105	5	2
2	2	9	2	**28**	28	3	123	**54**	54	7	64	**80**	80	2	111	**106**	106	8	1
3	3	1	126	**29**	29	5	32	**55**	55	4	8	**81**	81	6	111	**107**	107	2	4
4	4	9	4	**30**	30	3	32	**56**	56	3	119	**82**	82	8	111	**108**	108	7	1
5	5	2	126	**31**	31	1	32	**57**	57	6	64	**83**	83	5	4	**109**	109	4	2
6	6	1	125	**32**	32	9	32	**58**	58	5	64	**84**	84	8	123	**110**	110	4	16
7	7	3	126	**33**	33	2	95	**59**	59	4	64	**85**	85	8	2	**111**	111	9	111
8	8	9	8	**34**	34	4	95	**60**	60	3	64	**86**	86	8	8	**112**	112	3	111
9	9	4	126	**35**	35	6	95	**61**	61	2	64	**87**	87	2	8	**113**	113	3	2
10	10	2	125	**36**	36	4	123	**62**	62	1	64	**88**	88	5	119	**114**	114	6	1
11	11	5	126	**37**	37	8	95	**63**	63	9	63	**89**	89	7	2	**115**	115	1	4
12	12	1	123	**38**	38	7	125	**64**	64	9	64	**90**	90	8	32	**116**	116	5	1
13	13	6	126	**39**	39	5	8	**65**	65	1	63	**91**	91	4	4	**117**	117	2	2
14	14	3	125	**40**	40	2	119	**66**	66	2	63	**92**	92	6	32	**118**	118	4	1
15	15	3	16	**41**	41	8	119	**67**	67	3	63	**93**	93	4	32	**119**	119	9	119
16	16	9	16	**42**	42	8	125	**68**	68	4	63	**94**	94	2	32	**120**	120	3	1
17	17	4	111	**43**	43	8	4	**69**	69	5	63	**95**	95	9	95	**121**	121	1	2
18	18	4	125	**44**	44	5	123	**70**	70	6	63	**96**	96	1	95	**122**	122	2	1
19	19	7	126	**45**	45	8	16	**71**	71	3	8	**97**	97	3	95	**123**	123	9	123
20	20	2	123	**46**	46	6	16	**72**	72	4	119	**98**	98	5	95	**124**	124	1	1
21	21	8	126	**47**	47	2	16	**73**	73	7	63	**99**	99	3	4	**125**	125	9	125
22	22	5	125	**48**	48	1	111	**74**	74	8	63	**100**	100	7	95	**126**	126	9	126
23	23	6	8	**49**	49	5	111	**75**	75	6	4	**101**	101	6	2				
24	24	1	119	**50**	50	7	111	**76**	76	7	123	**102**	102	7	8				
25	25	7	119	**51**	51	7	4	**77**	77	7	16	**103**	103	1	8				
26	26	6	125	**52**	52	6	123	**78**	78	5	16	**104**	104	6	119				

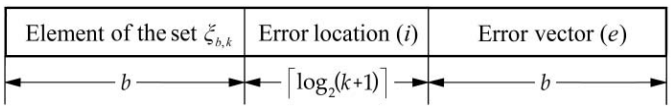

Fig. 1. Bit-width of one syndrome table entry.

Example 1: Let $b = 5$, $k = 2$ and $c = (3, 5)$. Using Definition 2, it is easy to show that the syndrome values $\{\pm 3, \pm 6, \pm 12, \pm 24, \pm 48, \pm 5, \pm 10, \pm 20, \pm 40, \pm 80, \mp 1, \mp 2, \mp 4, \mp 8, \mp 16\}$ are different modulo 31.

B. Error Correction Procedure

The error correction procedure for integer SEC codes is very similar to those described in [10]–[13]. More precisely, it consists of two steps: obtaining the error correction data from the syndrome table (Fig. 1) and executing one of the following operations:

- *for single errors within the i-th data byte*

$$B_i = [\hat{B}_i + e] \,(\text{mod } 2^b - 1), 1 \le i \le k; \quad (5)$$
$$e = [\pm 2^r] \,(\text{mod } 2^b - 1), 0 \le r \le b - 1;$$

- *for single errors within the check-byte*

$$C_B = [\hat{C}_B + e] \,(\text{mod } 2^b - 1); \quad (6)$$
$$e = [\pm\ 2^r] \,(\text{mod } 2^b - 1), 0 \le r \le b - 1;$$

To perform the second step correctly, the decoder must find the entry where the first b bits match that of the syndrome S. If the data are protected with non-perfect codes, this task will be completed after n_1 ($1 \le n_1 \le \lfloor log_2 |\xi_{b,k}| \rfloor + 2$) or n_2 ($1 \le n_2 \le |\xi_{b,k}|$) comparisons (depending on whether the elements of $\xi_{b,k}$ are sorted or not [16]). However, if the data are protected with perfect codes, the comparisons are not necessary. In that case, the syndrome value directly indicates the location of the corresponding entry.

Example 2: Let $b = 7$, $k = 8$ and $c = (3, 5, 7, 9, 11, 13, 19, 21)$. According to Theorems 1 and 2, the syndrome table will have $|\xi_{7,8}| = 126$ entries (Table 3). Given this, let us assume that we want to transmit 56 bits of data, D = 0110111 1111011 0011001 1010101 0001111 1100110 0101010 1110001. In that case, after calculating the value of check-byte C_B

$$C_B = \sum_{i=1}^{8} C_i \cdot B_i \,(\text{mod } 127) = 32 = 0100000_2$$

the codeword will have the form: $x = (B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8, C_B) = (55, 123, 25, 85, 15, 102, 42, 113, 32)$.

Scenario 1: Suppose that during data transmission an error on the 25th bit has occurred. In that case, the received vector will have the form: $y = (\underline{B}_1, \underline{B}_2, \underline{B}_3, \underline{B}_4, \underline{B}_5, \underline{B}_6, \underline{B}_7, \underline{B}_8, \underline{C}_B) = (55, 123, 25, 93, 15, 102, 42, 113, 32)$. As explained above, after calculating the syndrome S

$$S = \sum_{i=1}^{8} C_i \cdot \underline{B}_i - \underline{C}_B \,(\text{mod } 127) = 72$$

the decoder will instantly know the location of the appropriate table entry (Table 3). As a result, the following procedure will take place:

$$B_4 = [\underline{B}_4 + e] \,(\text{mod } 127) = [93 + 119] \,(\text{mod } 127) = 85.$$

Scenario 2: Assume that during data transmission an error on the 62th bit has occurred. In that case, the received vector will have the form: $y = (\underline{B}_1, \underline{B}_2, \underline{B}_3, \underline{B}_4, \underline{B}_5, \underline{B}_6, \underline{B}_7, \underline{B}_8, \underline{C}_B) = (55, 123, 25, 85, 15, 102, 42, 113, 34)$. Again, after calculating

$$S = \sum_{i=1}^{8} C_i \cdot \underline{B}_i - \underline{C}_B \,(\text{mod } 127) = 125$$

the decoder will instantly know the location of the appropriate table entry (Table 3). Hence, in the next step, it will perform error correction by using

$$C_B = [\underline{C}_B + e] \,(\text{mod } 127) = [34 + 125] \,(\text{mod } 127) = 32.$$

C. Potential Application

Although the proposed codes have weak error correcting capabilities, they could be useful in protocols for delivering multimedia content. For instance, it is known that multimedia applications mostly use UDP at the transport layer. One of the features of this protocol is that it drops errored packets even if one bit is wrong. By using the proposed codes, instead of the UDP checksum (UDPC) [17], the number of dropped packets can be significantly reduced. Of course, to achieve this it is necessary to perform some changes in router software (note that the UDPC is a special case of IECCs where $b = 16$ and $c = (1, 1, \ldots, 1) \in Z^k_{2^{16}-1}$).

III. Conclusion

This letter proposed a class of integer codes capable of correcting single errors. Unlike Hamming codes, the proposed codes are constructed with the help of a computer. The obtained results have shown that for practical data lengths up to 4096 bits, the proposed codes require one check bit more compared to Hamming codes. In addition, it has been shown that, for some values of b and k, the proposed codes are perfect. The parameters of these codes are $(2^{b-1} - 1, 2^{b-1} - b - 1)$, which makes them unique among all perfect codes.

References

[1] R. W. Hamming, "Error detecting and error correcting codes," *Bell Syst. Tech. J.*, vol. 29, no. 2, pp. 147–160, Apr. 1950.

[2] O. Heden, "A survey of perfect codes," *Adv. Math. Commun.*, vol. 2, no. 2, pp. 223–247, May 2008.

[3] F. I. Solov'eva, "On perfect binary codes," *Discrete Appl. Math.*, vol. 156, no. 9, pp. 1488–1498, May 2008.

[4] R. R. Varshamov and G. M. Tenengol'ts, "Correction code for single asymmetric errors," *Automat. Telemekh.*, vol. 26, no. 2, pp. 288–292, Feb. 1965.

[5] S. D. Constantin and T. R. N. Rao, "On the theory of binary asymmetric error correcting codes," *Inf. Control*, vol. 40, no. 1, pp. 20–26, Jan. 1979.

[6] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Sov. Phys.-Dokl.*, vol. 10, no. 8, pp. 707–710, Feb. 1966.

[7] W. C. Ferreira *et al.*, "Insertion/deletion correction with spectral nulls," *IEEE Trans. Inf. Theory*, vol. 43, no. 2, pp. 722–732, Mar. 1997.

[8] A. J. H. Vinck and H. Morita, "Codes over the ring of integers modulo m," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E81-A, no. 10, pp. 2013–2018, Oct. 1998.

[9] U. Tamm, "Reflections about a single checksum," in *Proc. 3rd Int. Conf. Arithmetic Finite Fields (WAIFI)*, Jun. 2010, pp. 238–249.

[10] A. Radonjic and V. Vujicic, "Integer codes correcting burst errors within a byte," *IEEE Trans. Comput.*, vol. 62, no. 2, pp. 411–415, Feb. 2013.

[11] A. Radonjic *et al.*, "Integer codes correcting double asymmetric errors," *IET Commun.*, vol. 10, no. 14, pp. 1691–1696, Sep. 2016.

[12] A. Radonjic and V. Vujicic, "Integer codes correcting spotty byte asymmetric errors," *IEEE Commun. Lett.*, vol. 20, no. 12, pp. 2338–2341, Dec. 2016.

[13] A. Radonjic and V. Vujicic, "Integer codes correcting high-density byte asymmetric errors," *IEEE Commun. Lett.*, vol. 21, no. 4, pp. 694–697, Apr. 2017.

[14] J. Fletcher, "An arithmetic checksum for serial transmissions," *IEEE Trans. Commun.*, vol. COM-30, no. 1, pp. 247–252, Jan. 1982.

[15] D. Bajic and C. Stefanovic, "Low power consuming FEC scheme," in *Proc. 4th Int. Workshop Optim. Codes Rel. Topics*, Jun. 2005, pp. 7–13.

[16] K. Mehlhorn and P. Sanders, *Algorithms and Data Structures: The Basic Toolbox*. Berlin, Germany: Springer, 2008.

[17] D. E. Comer, *Internetworking With TCP/IP*, 6th ed. Upper Saddle River, NJ, USA: Pearson, 2013.