



شبکه های خبراتی

سید حمید صفوی

دانشکده فنی و مهندسی

دانشگاه محقق اردبیلی

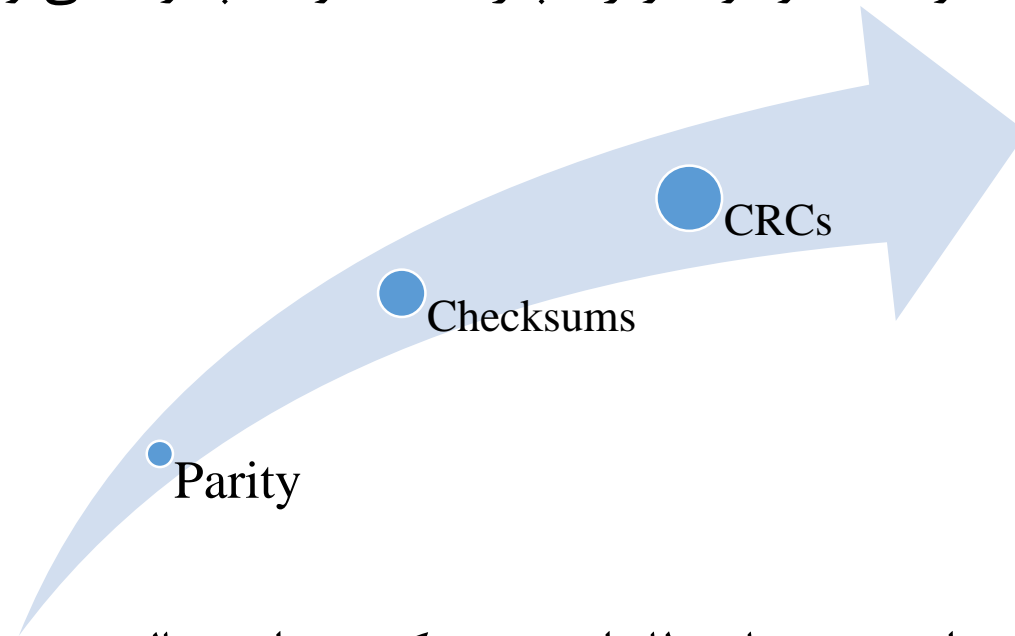
نیمسال دوم ۹۷-۹۸

تشخیص خطا



سرفصل

- ممکن است بعضی از بیت‌ها با توجه به وجود نویز دچار خطا شوند. چگونه می‌توانیم این بیت‌ها را شناسایی کنیم؟

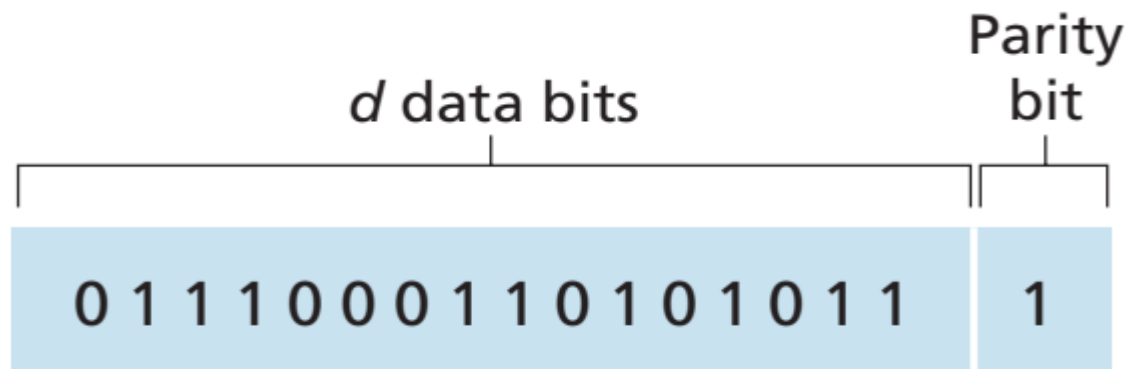


- شناسایی خطا به ما این فرصت را می‌دهد تا خطا را تصحیح کنیم برای مثال، به وسیله ارسال مجدد. (بعداً بررسی خواهد شد)

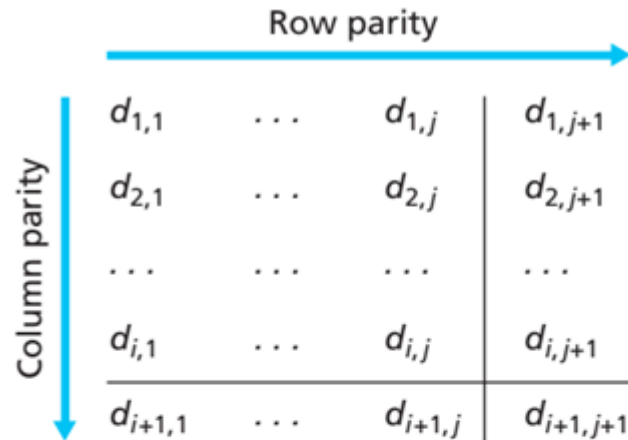


شناسایی خطای ساده – بیت توازن

- به ازای هر D بیت داده یک بیت توازن به محتوا اضافه می‌کند که این بیت توازن حاصل جمع D بیت داده است.
– حاصل جمع در مبنای ۲ یا XOR است.



بیت توازن



No errors

1	0	1	0	1	1
1	1	1	1	0	0
0	1	1	1	0	1
0	0	1	0	1	0

Correctable
single-bit error

1	0	1	0	1	1
1	0	1	1	0	0
0	1	1	1	0	1
0	0	1	0	1	0

Parity error

Parity error

• Parity چقدر خوب می تواند عمل کند؟

– فاصله همینگ (distance) این کد چقدر است؟ ۲

– چند خطا را می تواند شناسایی و اصلاح کند؟

یک بیت خطا را شناسایی می کند ولی نمی تواند خطایی را اصلاح کند.

• در مورد خطاهای بزرگ تر موفق عمل می کند؟

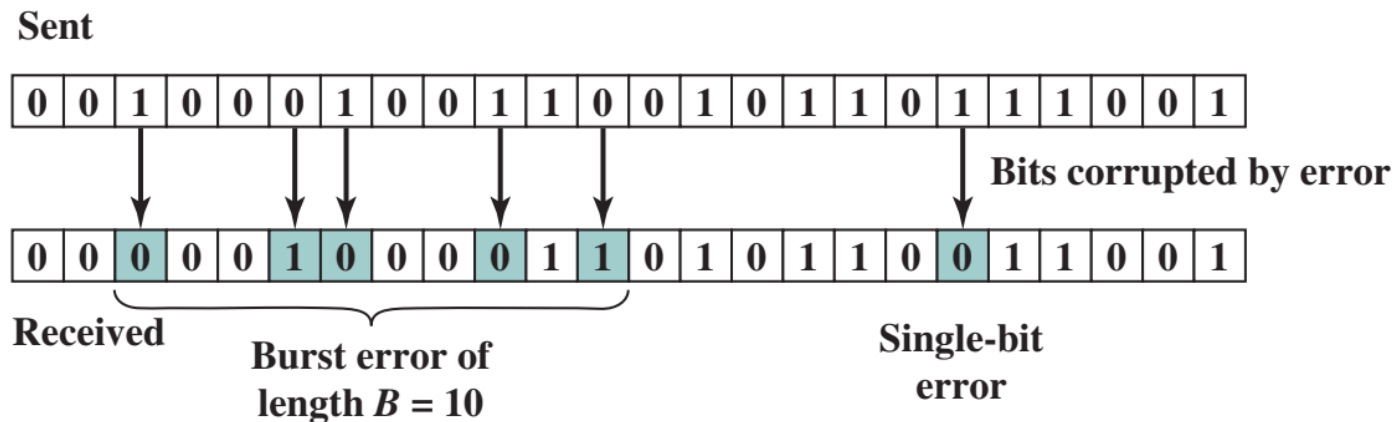
خیر. فقط تعداد فرد خطاها را تشخیص می دهد.

• Parity های دوبعدی



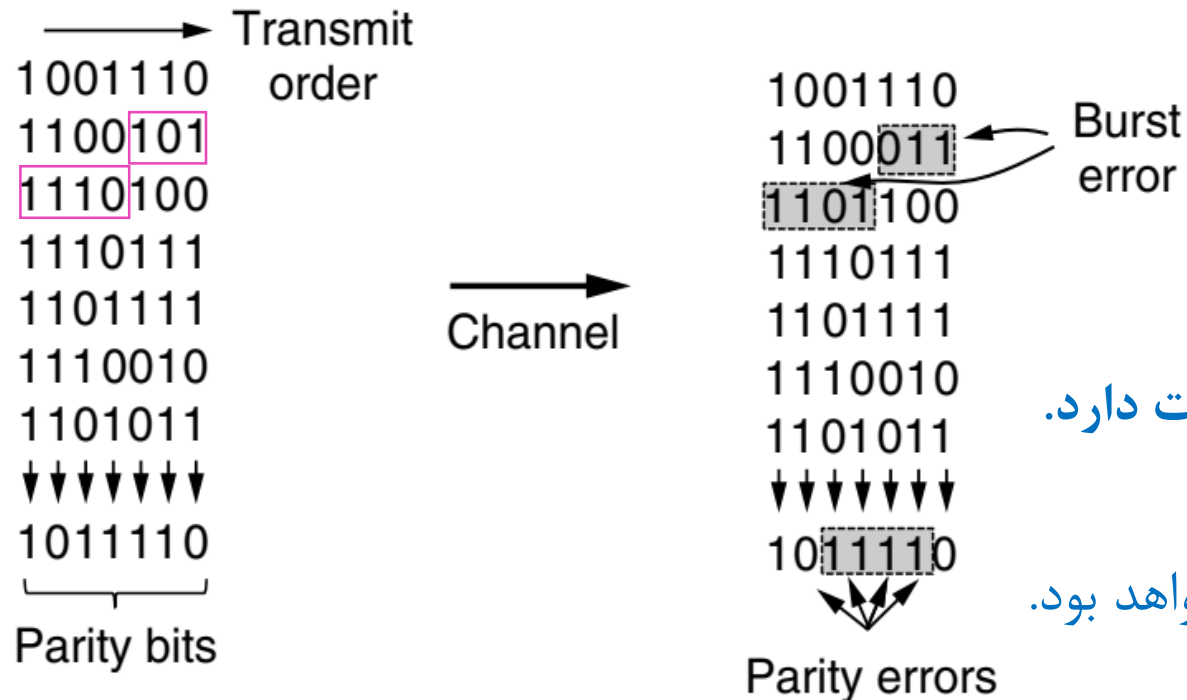
تشخیص خطای انفجاری با بیت توازن

- ایده‌ای برای تشخیص خطای انفجاری با بیت توازن دارید؟
- خطای انفجاری** به این مفهوم نیست که از بسته مثلا M تایی همه بیت‌ها خطا باشند. به این مفهوم است که حداقل بیت ابتدا و انتهای بسته M تایی دارای خطا باشد.



بیت توازن با Interleaving

- بیت توازن با Interleaving برای شناسایی خطاهای انفجاری (Burst error) کاربرد دارد.



- ایده اصلی:

- ترتیب ارسال با ترتیب محاسبه بیت توازن تفاوت دارد.
- داده ها را در n ستون k سطری مرتب می کنیم.
- بدیهی است که تعداد خطای $n+1$ قابل تشخیص نخواهد بود.

بیت توازن محاسبه شده: 0000



Checksums

- ایده اصلی: داده‌ها را در کلمه‌های N بیتی (N -bit words) جمع می‌زنند.
– به صورت گسترده در TCP ، IP ، UDP استفاده می‌شود.



- نسبت به بیت توازن قابلیت حفاظت بیشتری دارد.



Internet Checksums

- حاصل جمع، بر روی **حساب ریاضی مکمل یک** تعریف می شود. در محاسبه باید **حامل ها** نیز جمع شوند.
سپس حاصل را منفی می کنیم.

“The Checksum field is the 16 bit one’s complement of the one’s complement sum of all 16 bit words...” - RFC791



Internet Checksums (2)

00 01 F2 03 F4 F5 F6 F7 00 00

ارسال:

(۱) داده‌ها را به صورت کلمه‌های ۱۶ بیتی (۴ رقم hexadecimal) مرتب کنید.

(۲) صفر را در جایگاه checksum قرار دهید و جمع بزنید.

(۳) برای رسیدن به ۱۶ بیت باید تمام اعداد حامل موجود را از عدد به دست آمده جدا کرده و با خود عدد جمع بزنید.

(۴) حاصل را مکمل منفی کنید.

```
0001
f203
f4f5
f6f7
+ (0000)
-----
2ddf0
  ↓
ddf0
+    2
-----
ddf2
  ↓
220d
```



Internet Checksums (3)

دریافت:

$$\begin{array}{r} 0001 \\ f203 \\ f4f5 \\ f6f7 \\ + 220d \\ \hline 2fffd \\ \downarrow \\ fffd \\ + 2 \\ \hline ffff \\ \downarrow \\ 0000 \end{array}$$

(۱) داده‌ها را به صورت کلمه‌های ۱۶ بیتی مرتب کنید.

(۲) Checksum غیرصفر را به عدد اضافه کنید و با آن جمع بزنید.

(۳) برای رسیدن به ۱۶ بیت باید تمام اعداد حامل موجود را از عدد به دست آمده جدا کرده و با خود عدد جمع بزنید.

(۴) حاصل را منفی کنید. «اگر حاصل صفر باشد به این معناست که خطایی در ارسال پیام رخ نداده است و اگر غیرصفر باشد به معنای وجود خطاست»



Internet Checksums (3)

00 01 F2 03 F4 F5 F6 F7 00 00

Partial sum	$\begin{array}{r} 0001 \\ F203 \\ \hline F204 \end{array}$
Partial sum	$\begin{array}{r} F204 \\ F4F5 \\ \hline 1E6F9 \end{array}$
Carry	$\begin{array}{r} E6F9 \\ 1 \\ \hline E6FA \end{array}$
Partial sum	$\begin{array}{r} E6FA \\ F6F7 \\ \hline 1DDF1 \end{array}$
Carry	$\begin{array}{r} DDF1 \\ 1 \\ \hline DDF2 \end{array}$
Ones complement of the result	220D

(a) Checksum calculation by sender

Partial sum	$\begin{array}{r} 0001 \\ F203 \\ \hline F204 \end{array}$
Partial sum	$\begin{array}{r} F204 \\ F4F5 \\ \hline 1E6F9 \end{array}$
Carry	$\begin{array}{r} E6F9 \\ 1 \\ \hline E6FA \end{array}$
Partial sum	$\begin{array}{r} E6FA \\ F6F7 \\ \hline 1DDF1 \end{array}$
Carry	$\begin{array}{r} DDF1 \\ 1 \\ \hline DDF2 \end{array}$
Partial sum	$\begin{array}{r} DDF2 \\ 220D \\ \hline FFFF \end{array}$

(b) Checksum verification by receiver



Internet Checksums (4)

- checksum تا چه اندازه می تواند خوب کار کند؟
 - فاصله (distance) این کد چقدر است؟ ۲
 - چند خطا را می تواند تشخیص و تصحیح کند؟
 - یک خطا را شناسایی می کند ولی نمی تواند خطایی را تصحیح کند.
 - در مورد خطاهای بزرگ تر موفق عمل می کند؟
 - خطاهای انفجاری تا ۱۶ بیت.
- به دلیل اینکه این روش بر روی کلمه کد عمل می کند و نه بیت ها، می تواند خطاهایی را که دچار تغییر بیت توازن نمی شوند ولی checksum را تغییر می دهند را تشخیص دهد.



Internet Checksums (5)

0 0 0 1 F 2 0 3 F 4 F 5 F 6 F 7 0 0 0 0

0000 0000 0000 0001 1111 0010 0000 0011 1111 0100 1111 0101 1111 0110 1111 0111 0000 0000 0000 0000

- مثال قبلی را در نظر بگیرید. بیت توازن زوج برای رشته بیت برابر صفر است. حال فرض کنید خطای انفجاری به صورت زیر رخ داده است و ۱۲ بیت را تحت تاثیر قرار داده است:

0000 0000 0000 0001 1111 1111 0010 0000 1111 0100 1111 0101 1111 0110 1111 0111 0000 0000 0000 0000

- با رخ دادن خطای انفجاری در بیت‌های نشان داده شده، بیت توازن همچنان صفر است، اما checksum که برابر 220D بود، با رخ دادن خطا عوض می‌شود. گیرنده برای چک کردن خطای پیش آمده از checksum بدون خطا که 220D است، استفاده می‌کند.



Internet Checksums (6)

دریافت:

0	0	0	1	F	F	2	0	F	4	F	5	F	6	F	7	2	2	0	D
0000	0000	0000	0001	1111	1111	0010	0000	1111	0100	1111	0101	1111	0110	1111	0111	0010	0010	0000	1101

0001

FF20 Error

F4F5

F6F7

+ 220D Checksum of correct word

30D1A



0D1A

+ 3

0D1D → F2E2

حاصل غیر صفر، معادل وجود خطا

بنابراین خطای انفجاری تشخیص داده شد.



Cyclic Redundancy Check (CRC)

- قابلیت حفاظت بیشتر و قوی تر

– برای n بیت داده، k تا k check bit را به گونه ای می سازد که $n+k$ بیت به دست آمده به صورت دسته هایی با تعداد زوج بر C که خود یک مولد است بخش پذیر باشد.

- مثال عددی:

$$n = 302, k = \text{one digit}, C = 3$$

$$30\underline{21}$$

$$\text{mod}(21 \div 3) = 0$$

$$30\underline{20}$$

$$\text{mod}(20 \div 3) = 2$$



CRCs (2)

- دست آوردهای این روش:

– بر پایه ریاضیات میدانهای متناهی کار می کند که در آن اعداد بیانگر چند جمله ای ها هستند.

– مثال
e.g. 10011010 is $X^7 + X^4 + X^3 + X^1$

- معنای آنچه در بالا گفته شد چیست؟

– ما با مقادیر دوتایی یا همان باینری کار می کنیم و از محاسبات ریاضی مبنای دو بهره می گیریم.



CRCs (3)

• فرآیند ارسال:

- (۱) n بیت داده را با k تا صفر بسط می دهد.
- (۲) با استفاده از مقدار مولد C تقسیم بندی را انجام می دهد.
- (۳) باقیمانده را نگه می دارد و خارج قسمت را نادیده می گیرد.
- (۴) k تا check bit را به وسیله باقیمانده تنظیم می کند.

• فرآیند دریافت:

- (۱) برای رسیدن به باقیمانده صفر تقسیم بندی و چک می کند.



CRCs (4)

Data Bits:

1101011111

Check Bits:

$$X^4 + X^1 + 1$$

$$C = 10011$$

$$K = 4$$

$$\begin{array}{r}
 10011 \overline{) 11010111110000} \\
 \underline{- 10011} \\
 10011 \\
 \underline{- 10011} \\
 11110 \\
 \underline{- 10011} \\
 11010 \\
 \underline{- 10011} \\
 10010 \\
 \underline{- 10011} \\
 0010
 \end{array}$$

1001111010111110010

باقیمانده

فریم ارسالی



CRCs (5)

- قابلیت محافظت این روش وابسته به مقدار مولد است.

Standard CRC-32 is 100000100110000010001110110110111

- خاصیت‌های این روش:
 - $HD=4$ ، تا سه بیت خطا را می‌تواند تشخیص دهد.
 - همچنین تعداد خطاهای فرد را نیز تشخیص می‌دهد.
 - خطاهای انفجاری تا k بیت را تشخیص می‌دهد.
 - نسبت به خطاهای سیستماتیک آسیب‌ناپذیر است مانند checksums.



تشخیص خطا در عمل

- **CRC ها** به صورت گسترده در انواع **لینک ها** استفاده می شوند:

– Ethernet , 802.11 , ADSL , Cable ...

- **Checksum ها** در **اینترنت** استفاده می شوند:

– IP , TCP , UDP ... (اما ضعیف هستند)

- **بیت توازن**

– بسیار کم استفاده می شوند.

