Combinatorial Systematic Switch codes

Hui Zhang

Joint work with
Y. M. Chee, F. Gao and S. T. H. Teo

Computer Science Department
Technion

April 10, 2016

Reference

Z. Wang, O. Shaked, Y. Cassuto, and J. Bruck, "Codes for network switches," in *ISIT 2013*, 2013, pp. 1057–1061.

Y. M. Chee, F. Gao, S. T. H. Teo, and H. Zhang, "Combinatorial systematic switch codes," in *ISIT 2015*, 2015, pp. 241–245.

Z. Wang, H. M. Kiah, and Y. Cassuto, "Optimal binary switch codes with small query size," in *ISIT 2015*, 2015 pp. 636–640.

Background

Problem Formulation

Consecutive-Generation Switch Codes

Switch Codes Serving Other Requests

Background

- A network switch is a computer networking device that connects devices together in a computer network.

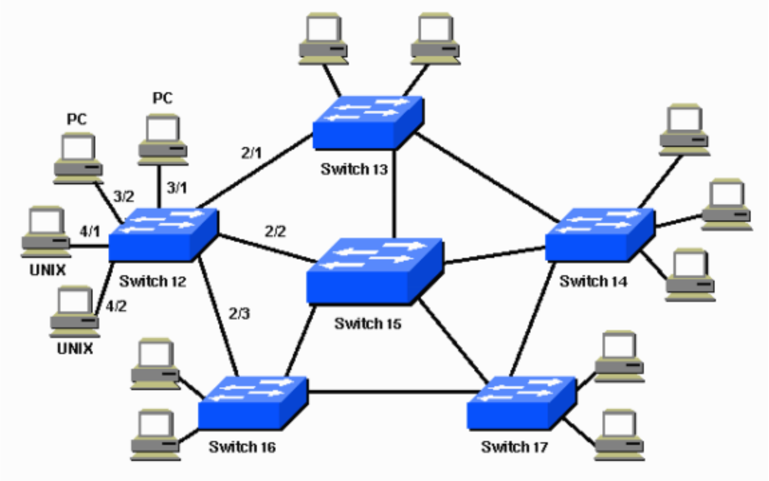
Background

- A network switch is a computer networking device that connects devices together in a computer network.
- A subsystem of a network switch, called switch memory, is used to temporarily cache data packets between their arrival from input ports and departure at output ports.

Background

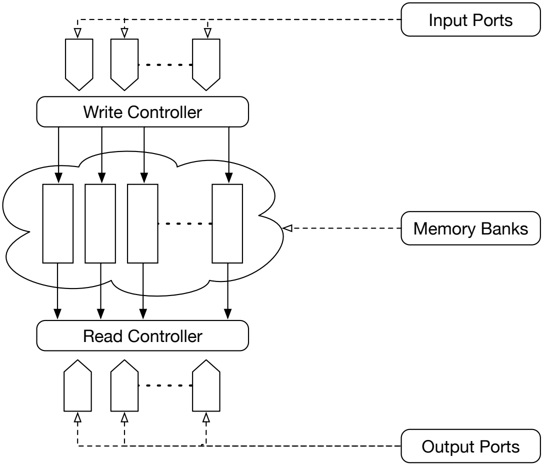
- A network switch is a computer networking device that connects devices together in a computer network.
- A subsystem of a network switch, called switch memory, is used to temporarily cache data packets between their arrival from input ports and departure at output ports.
- Try to improve its data exchange speed by applying coding on the network switches.

Network Switches[1]

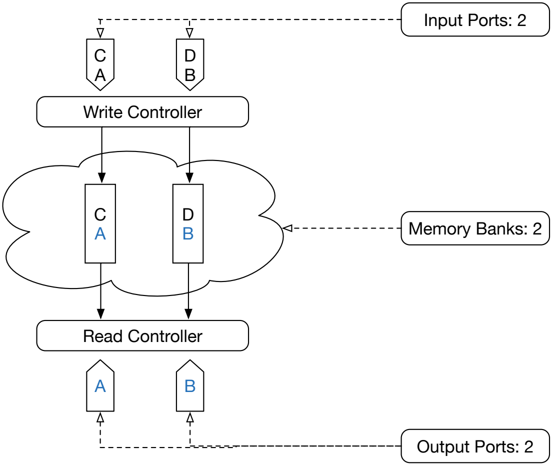


[1] Image from http://www.cisco.com/c/en/us/support/docs/lan-switching/spanning-tree-protocol/5234-5.html

Switch Memory Subsystem

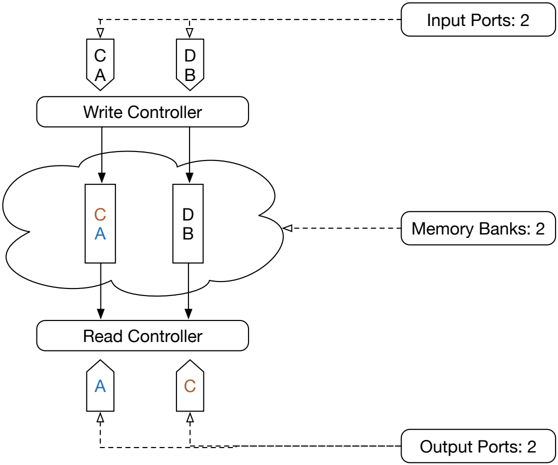


Data Flow

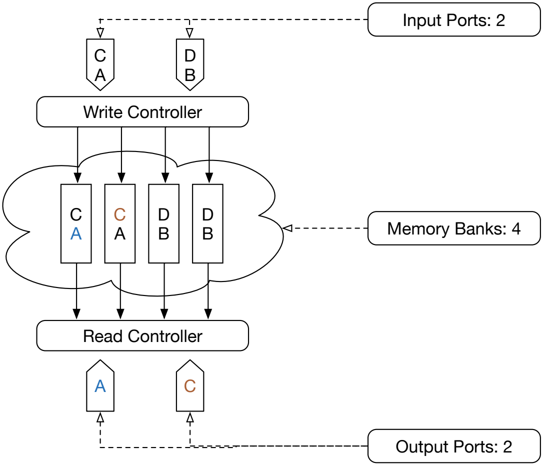


Input Ports: 2
C
A
D
B
Write Controller
C
A
D
B
Memory Banks: 2
Read Controller
A
B
Output Ports: 2

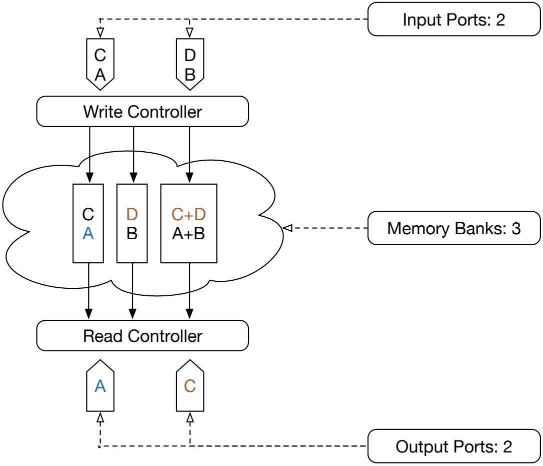
Data Flow with Conflict



Solution by Replication



Solution by Coding[2]



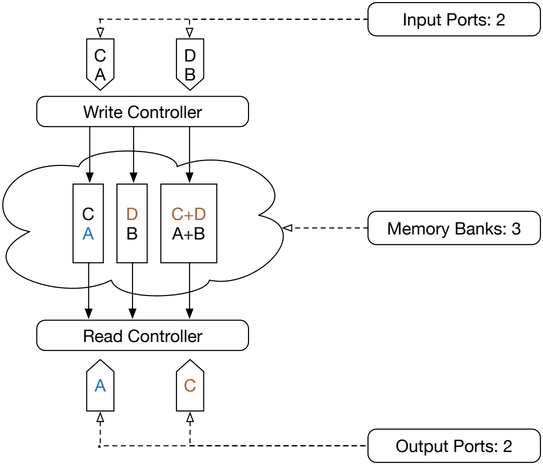
[2] Z. Wang, O. Shaked, Y. Cassuto, and J. Bruck, "Codes for network switches," in *ISIT 2013*, pp. 1057–1061.

Problem Formulation

- R_{in}: number of input ports
- R_{out}: number of output ports
- n: number of memory banks
- Redundancy: $n - R_{\text{in}}$
- Generation: data packets written in the same time slot

Example

$R_{\text{in}} = R_{\text{out}} = 2$, $n = 3$, Redundancy$= 1$, two generations.



Problem Formulation: Encoding

Encoding: A function f from generations of input data packets to written bits in memory banks

$$\begin{aligned} f : \Sigma^{R_{\text{in}}} &\longrightarrow \Sigma^{n} \\ U &\longmapsto X, \end{aligned}$$

where $U = (u_{i,1}, u_{i,2}, \ldots, u_{i,R_{\text{in}}})$ and $X = (x_{i,1}, x_{i,2}, \ldots, x_{i,n})$ are input and written bits of the i-**th** generation.

Problem Formulation: Encoding

Encoding: A function f from generations of input data packets to written bits in memory banks

$$\begin{aligned} f : \Sigma^{R_{\text{in}}} &\longrightarrow \Sigma^n \\ U &\longmapsto X, \end{aligned}$$

where $U = (u_{i,1}, u_{i,2}, \ldots, u_{i,R_{\text{in}}})$ and $X = (x_{i,1}, x_{i,2}, \ldots, x_{i,n})$ are input and written bits of the i-**th** generation.

Let $(u_{i_1,l_1},\ u_{i_2,l_2},\ \ldots,\ u_{i_{R_{\text{out}}},l_{R_{\text{out}}}})$ be arbitrary R_{out} request bits from previous generations.

- Request Bits: $(u_{i_1,l_1},\ u_{i_2,l_2},\ \ldots,\ u_{i_{R_{\text{out}}},l_{R_{\text{out}}}})$
- **Request vector**: $L = (l_1, l_2, \ldots, l_{R_{\text{out}}})$
- Partition of L, L_i, $1 \le i \le t$: denotes request sets from the i-th generation.

Example

The request bits are $(u_{1,1}, u_{1,2}, u_{2,3}, u_{3,3})$, then $L = (1, 2, 3, 3)$, $L_1 = \{1, 2\}$, $L_2 = \{3\}$, $L_3 = \{3\}$.

Solution: A way to read request bits by retrieving **no more than one** bit from each memory bank.

Different type of requests

Request vector: $L = (l_1, l_2, \ldots, l_{R_{\text{out}}})$

- t **consecutive-generation request**
- In the worst case, the request bits are all from different generations.

1. **limited-repetition request**: each integer repeats at most a certain number of times in L.
2. **one burst request**: only one integer appear more that once in one request vector (**Locally recoverable codes**).
3. the request vector consists of all multisets (**Batch Codes**).

Definitions

- Request Bits: $(u_{i_1,l_1},\ u_{i_2,l_2},\ \ldots,\ u_{i_{R_{\text{out}}},l_{R_{\text{out}}}})$
- **Request vector**: $L = (l_1, l_2, \ldots, l_{R_{\text{out}}})$
- Partition of L, L_i, $1 \leq i \leq t$: denotes request sets from the i-th generation.

A t-**consecutive-generation switch code**: $(n, R_{\text{in}}, R_{\text{out}}; t)$ **switch code**

The t-**consecutive-generation switch codes** *are applicable when the data packets expire after a fixed time slot and only the newest* t *generations are of interest.*

For $t = 2$:

Theorem (Wang et al. (2013))

There is an $(n, R_{\text{in}}, R_{\text{out}}; 2)$ *switch code with redundancy* $R_{\text{in}} - 1$ *if* $R_{\text{out}} \leq 2R_{\text{in}} - 1$.

Proof.

Take a path on R_{in} vertices, encoding with this path.

$$\begin{aligned}
(u_{i,1}, u_{i,2}, \ldots, u_{i,R_{\text{in}}}) \Rightarrow \\
& (u_{i,1}, u_{i,2}, \ldots, u_{i,R_{\text{in}}}, \\
& \quad u_{i,1} + u_{i,2}, u_{i,2} + u_{i,3}, \ldots, u_{i,R_{\text{in}}-1} + u_{i,R_{\text{in}}})
\end{aligned}$$

$\square$

Example

$R_{\text{in}} = 4$:

$$
\begin{bmatrix}
u_{2,1} & u_{2,2} & u_{2,3} & \underline{u_{2,4}} & \underline{u_{2,1} + u_{2,2}} & \underline{u_{2,2} + u_{2,3}} & \underline{u_{2,3} + u_{2,4}} \\
\underline{u_{1,1}} & \underline{u_{1,2}} & \underline{u_{1,3}} & u_{1,4} & u_{1,1} + u_{1,2} & u_{1,2} + u_{1,3} & u_{1,3} + u_{1,4}
\end{bmatrix}
$$

For general t:

Lemma

A complete graph on n vertices contains $\lfloor n/2 \rfloor$ edge-disjoint Hamiltonian paths.

Theorem

There is an $(n, R_{\text{in}}, R_{\text{out}}; t)$ switch code with **redundancy** $(R_{\text{in}} - 1)(t - 1)$, *provided*

$$t - 1 \leq \lfloor R_{\text{in}}/2 \rfloor \text{ and } t - 1 \leq R_{\text{in}} - \lfloor R_{\text{out}}/t \rfloor .$$

Proof

- $\sum^{R_{\text{in}}} \longrightarrow \sum^{R_{\text{in}}+(R_{\text{in}}-1)(t-1)}$
- Let G be a subgraph of the complete graph on R_{in} vertices, formed by the edges of $t-1$ edge-disjoint hamiltonian paths, where $t-1 \leq \lfloor R_{\text{in}}/2 \rfloor$.
- Let the $t-1$ edge-disjoint hamiltonian paths in G be $P_1, P_2, \ldots, P_{t-1}$, and let the request sets of the t consecutive generations be $L_0, L_1, \ldots, L_{t-1}$.

Proof.Cont

- W.l.o.g, assume that L_0 has the smallest size among the L_i's, so that $|L_0| \leq \lfloor R_{\text{out}}/t \rfloor \leq R_{\text{in}} - (t-1)$. Therefore we can choose $t-1$ distinct elements $h_1, h_2, \ldots, h_{t-1}$ from $[1, R_{\text{in}}] \setminus L_0$.
- To serve the requests in L_0, we directly read the corresponding systematic elements in written bits. To read the requested bits in L_i, for $1 \leq i \leq t-1$, we use systematic bit h_i and edges in the hamiltonian path P_i.

Example

$R_{\text{in}} = 4$:

$$\begin{bmatrix} \color{red}{u_{3,1}} & \color{red}{u_{3,2}} & \color{red}{u_{3,3}} & \underline{u_{3,4}} & \underline{u_{3,1}+u_{3,2}} & \underline{u_{3,2}+u_{3,3}} & \underline{u_{3,3}+u_{3,4}} & u_{3,2}+u_{3,4} & u_{3,4}+u_{3,1} & u_{3,1}+u_{3,3} \\ u_{2,1} & \color{red}{u_{2,2}} & \underline{\color{red}{u_{2,3}}} & \color{red}{u_{2,4}} & u_{2,1}+u_{2,2} & u_{2,2}+u_{2,3} & u_{2,3}+u_{2,4} & \underline{u_{2,2}+u_{2,4}} & \underline{u_{2,4}+u_{2,1}} & \underline{u_{2,1}+u_{2,3}} \\ \underline{\color{red}{u_{1,1}}} & \underline{\color{red}{u_{1,2}}} & u_{1,3} & u_{1,4} & u_{1,1}+u_{1,2} & u_{1,2}+u_{1,3} & u_{1,3}+u_{1,4} & u_{1,2}+u_{1,4} & u_{1,4}+u_{1,1} & u_{1,1}+u_{1,3} \end{bmatrix}$$

Bound

Lemma

A lower bound for the redundancy of a systematic pair-parity code is $n - R_{\text{in}} \geq \left\lceil \frac{R_{\text{in}}(t-1)}{2} \right\rceil$*, when a request is restricted to* t *consecutive generations.*

Lemma

When $2 \leq t \leq \lfloor R_{\text{out}}/2 \rfloor$*, the lower bound cannot be achieved.*

Background

Problem Formulation

Consecutive-Generation Switch Codes

Switch Codes Serving Other Requests

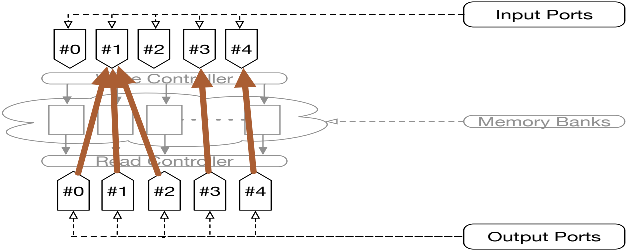
Switch Codes Serving One-Burst Requests

- An $(n, R_{\text{in}}, R_{\text{out}})$ switch code: not require any information on the generations of request bits.
- A **one-burst request**: only one integer repeats in the request vector.
- Burst request switch codes can be used to clean up the longest request queue among the memory banks and improve the worst case delay.

One-Burst Requests

Example

$R_{\text{in}} = R_{\text{out}} = 5,\ L = (1, 1, 1, 3, 4)$



Example

$\{0,1,2\}$ $\{0,2,3\}$ $\{0,1,4\}$ $\{1,2,5\}$ $\{0,3,5\}$
$\{2,3,4\}$ $\{0,4,5\}$ $\{1,3,4\}$ $\{1,3,5\}$ $\{2,4,5\}$

$(16,6,6)$ switch codes for one burst request

$L = (0,0,0,0,0,0)$

$$\begin{array}{c} 0 \\ \{0,1,2\},\{1,2,5\},\{5\} \\ \{0,2,3\},\{2,3,4\},\{4\} \\ \{0,1,4\},\{1,3,4\},\{3\} \\ \{0,3,5\},\{1,3,5\},\{1\} \\ \{0,4,5\},\{2,4,5\},\{2\} \end{array}$$

$(v, \{3\}, 2)^*$-designs

Let X be $\{0, 1, \ldots, R_{\text{in}} - 1\}$ and $\mathcal{B}$ be some triples in X called *blocks*;

(C1) Each pair of X appears in exactly two blocks;

(C2) $|\mathcal{B}_a \cap \mathcal{B}_b| = 1$ for all distinct $a, b \in X$, where $\mathcal{B}_a = \{\{x, y\} : \{a, x, y\} \in \mathcal{B}\}$;

(C3) There do not exist three blocks of the form $\{a, b, c\}$, $\{a, b, d\}$, $\{b, c, d\}$ in $\mathcal{B}$.

Connections

Theorem

A $(v, \{3\}, 2)^*$*-designs gives a solution for any one-burst request for* $R_{in} = R_{out} = v$ *bits.*

- Using sums of blocks in $\mathcal{B}$ as parity check bits, we have an $(R + R(R-1)/3, R, R)$ switch code for any one-burst request.
- Connection to locally repairable codes [3]

[3] A. Vardy and E. Yaakobi, "Constructions of batch codes with optimal redundancy."

Connections

Theorem
There exists a $(v, \{3\}, 2)^*$*-design for all* $v \equiv 0, 1 \pmod 6$*,* $v \geq 1116$*.*

$\rightarrow (R + R(R-1)/3, R, R)$ switch code for any one-burst request

Switch Codes Serving Any Requests

Theorem (Wang et al., 2015[4])

- *Simplex code is an* $(n, R_{\text{in}}, R_{\text{out}})$ *switch code with* $n = 2^k - 1$, $R_{\text{in}} = k$ *and* $R_{\text{out}} = 2^{k-1}$.
- *Concatenation of the Simplex codes is also a switch code.*

$$\begin{bmatrix} \mathbf{G} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{G} & \cdots & \mathbf{0} \\ & \cdots & \cdots & \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{G} \end{bmatrix}$$

where $\mathbf{G}$ is the generater matrix of the Simplex code.

[4] Z. Wang, H. M. Kiah, and Y. Cassuto, "Optimal binary switch codes with small query size," in *ISIT 2015*, pp. 636–640.

Thank you.