WIKIPEDIA

Clos network

In the field of telecommunications, a **Clos network** is a kind of multistage circuit-switching network which represents a theoretical idealization of practical, multistage switching systems. It was invented by Edson Erwin[1] in 1938 and first formalized by Charles Clos (French pronunciation: [ʃaʁl klo])[2] in 1952.

By adding stages, a Clos network reduces the number of crosspoints required to compose a large crossbar switch. A Clos network topology (diagrammed below) is parameterized by three integers n, m, and r: n represents the number of sources which feed into each of r ingress stage crossbar switches; each ingress stage crossbar switch has m outlets; and there are m middle stage crossbar switches.

Circuit switching arranges a dedicated communications path for a connection between endpoints for the duration of the connection. This sacrifices total bandwidth available if the dedicated connections are poorly utilized, but makes the connection and bandwidth more predictable, and only introduces control overhead when the connections are initiated, rather than with every packet handled, as in modern packet-switched networks.

When the Clos network was first devised, the number of crosspoints was a good approximation of the total cost of the switching system. While this was important for electromechanical crossbars, it became less relevant with the advent of VLSI, wherein the interconnects could be implemented either directly in silicon, or within a relatively small cluster of boards. Upon the advent of complex data centers, with huge interconnect structures, each based on optical fiber links, Clos networks regained importance.[3] A subtype of Clos network, the Beneš network, has also found recent application in machine learning.[4]

Contents

Topology

Clos networks have three stages: the ingress stage, the middle stage, and the egress stage. Each stage is made up of a number of crossbar switches (see diagram below), often just called *crossbars*. The network implements an r-way perfect shuffle between stages. Each call entering an ingress crossbar switch can be routed through any of the available middle stage crossbar switches, to the relevant egress crossbar switch. A middle stage crossbar is available for a particular new call if both the link connecting the ingress switch to the middle stage switch, and the link connecting the middle stage switch to the egress switch, are free.

| $r\ n \times m$ crossbars | $m\ r \times r$ crossbars | $r\ m \times n$ crossbars |
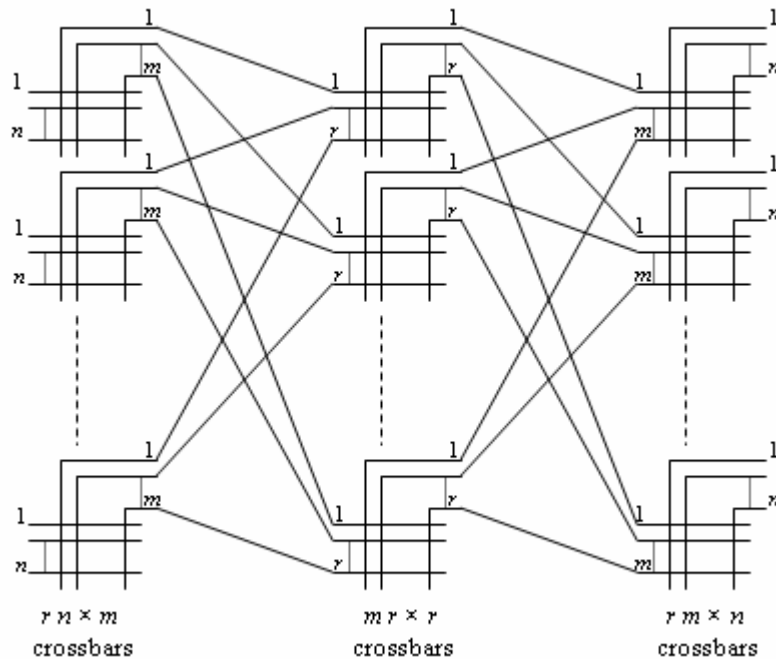
Clos networks are defined by three integers n, m, and r. n represents the number of sources which feed into each of r ingress stage crossbar switches. Each ingress stage crossbar switch has m outlets, and there are m middle stage crossbar switches. There is exactly one connection between each ingress stage switch and each middle stage switch. There are r egress stage switches, each with m inputs and n outputs. Each middle stage switch is connected exactly once to each egress stage switch. Thus, the ingress stage has r switches, each of which has n inputs and m outputs. The middle stage has m switches, each of which has r inputs and r outputs. The egress stage has r switches, each of which has m inputs and n outputs.

Blocking characteristics

The relative values of m and n define the blocking characteristics of the Clos network.

Strict-sense nonblocking Clos networks ($m \geq 2n-1$): the original 1953 Clos result

If $m \geq 2n-1$, the Clos network is *strict-sense nonblocking*, meaning that an unused input on an ingress switch can always be connected to an unused output on an egress switch, *without having to re-arrange existing calls*. This is the result which formed the basis of Clos's classic 1953 paper. Assume that there is a free terminal on the input of an ingress switch, and this has to be connected to a free terminal on a particular egress switch. In the worst case, $n-1$ other calls are active on the ingress switch in question, and $n-1$ other calls are active on the egress switch in question. Assume, also in the worst case, that each of these calls passes through a different middle-stage switch. Hence in the worst case, $2n-2$ of the middle stage switches are unable to carry the new call. Therefore, to ensure strict-sense nonblocking operation, another middle stage switch is required, making a total of $2n-1$.

Rearrangeably nonblocking Clos networks ($m \geq n$)

If $m \geq n$, the Clos network is *rearrangeably nonblocking*, meaning that an unused input on an ingress switch can always be connected to an unused output on an egress switch, but for this to take place, existing calls may have to be rearranged by assigning them to different centre stage switches in the Clos network.[5]

To prove this, it is sufficient to consider $m = n$, with the Clos network fully utilised; that is, $r \times n$ calls in progress. The proof shows how any permutation of these $r \times n$ input terminals onto $r \times n$ output terminals may be broken down into smaller permutations which may each be implemented by the individual crossbar switches in a Clos network with $m = n$.

The proof uses Hall's marriage theorem[6] which is given this name because it is often explained as follows. Suppose there are r boys and r girls. The theorem states that if every subset of k boys (for each k such that $0 \le k \le r$) between them know k or more girls, then each boy can be paired off with a girl that he knows. It is obvious that this is a necessary condition for pairing to take place; what is surprising is that it is sufficient.

In the context of a Clos network, each boy represents an ingress switch, and each girl represents an egress switch. A boy is said to know a girl if the corresponding ingress and egress switches carry the same call. Each set of k boys must know at least k girls because k ingress switches are carrying $k \times n$ calls and these cannot be carried by less than k egress switches. Hence each ingress switch can be paired off with an egress switch that carries the same call, via a one-to-one mapping. These r calls can be carried by one middle-stage switch. If this middle-stage switch is now removed from the Clos network, m is reduced by 1, and we are left with a smaller Clos network. The process then repeats itself until $m = 1$, and every call is assigned to a middle-stage switch.

Blocking probabilities: the Lee and Jacobaeus approximations

Real telephone switching systems are rarely strict-sense nonblocking for reasons of cost, and they have a small probability of blocking, which may be evaluated by the Lee or Jacobaeus approximations,[7] assuming no rearrangements of existing calls. Here, the potential number of other active calls on each ingress or egress switch is $u = n-1$.

In the Lee approximation, it is assumed that each internal link between stages is already occupied by a call with a certain probability p, and that this is completely independent between different links. This overestimates the blocking probability, particularly for small r. The probability that a given internal link is busy is $p = uq/m$, where q is the probability that an ingress or egress link is busy. Conversely, the probability that a link is free is $1-p$. The probability that the path connecting an ingress switch to an egress switch via a particular middle stage switch is free is the probability that both links are free, $(1-p)^2$. Hence the probability of it being unavailable is $1-(1-p)^2 = 2p-p^2$. The probability of blocking, or the probability that no such path is free, is then $[1-(1-p)^2]^m$.

The Jacobaeus approximation is more accurate, and to see how it is derived, assume that some particular mapping of calls entering the Clos network (input calls) already exists onto middle stage switches. This reflects the fact that only the *relative* configurations of ingress switch and egress switches is of relevance. There are i input calls entering via the same ingress switch as the free input terminal to be connected, and there are j calls leaving the Clos network (output calls) via the same egress switch as the free output terminal to be connected. Hence $0 \le i \le u$, and $0 \le j \le u$.

Let A be the number of ways of assigning the j output calls to the m middle stage switches. Let B be the number of these assignments which result in blocking. This is the number of cases in which the remaining $m-j$ middle stage switches coincide with $m-j$ of the i input calls, which is the number of subsets containing $m-j$ of these calls. Then the probability of blocking is:

$$\beta_{ij} = \frac{B}{A} = \frac{\dbinom{i}{m-j}}{\dbinom{m}{j}} = \frac{i!j!}{(i+j-m)!m!}$$

If f_i is the probability that i other calls are already active on the ingress switch, and g_j is the probability that j other calls are already active on the egress switch, the overall blocking probability is:

$$P_B = \sum_{i=0}^{u} \sum_{j=0}^{u} f_i g_j \beta_{ij}$$

This may be evaluated with f_i and g_j each being denoted by a [binomial distribution](). After considerable algebraic manipulation, this may be written as:

$$P_B = \frac{(u!)^2 (2-p)^{2u-m} p^m}{m!(2u-m)!}$$

Clos networks with more than three stages

Clos networks may also be generalised to any odd number of stages. By replacing each centre stage crossbar switch with a 3-stage Clos network, Clos networks of five stages may be constructed. By applying the same process repeatedly, 7, 9, 11,... stages are possible.

Beneš network ($m = n = 2$)

A rearrangeably nonblocking network of this type with $m = n = 2$ is generally called a *Beneš network*, even though it was discussed and analyzed by others before [Václav E. Beneš](). The number of inputs and outputs is $N = r \times n = 2r$. Such networks have $2\log_2 N - 1$ stages, each containing $N/2$ 2×2 crossbar switches, and use a total of $N\log_2 N - N/2$ 2×2 crossbar switches. For example, an 8×8 Beneš network (i.e. with $N = 8$) is shown below; it has $2\log_2 8 - 1 = 5$ stages, each containing $N/2 = 4$ 2×2 crossbar switches, and it uses a total of $N\log_2 N - N/2 = 20$ 2×2 crossbar switches. The central three stages consist of two smaller 4×4 Beneš networks, while in the center stage, each 2×2 crossbar switch may itself be regarded as a 2×2 Beneš network. This example therefore highlights the recursive construction of this type of network.
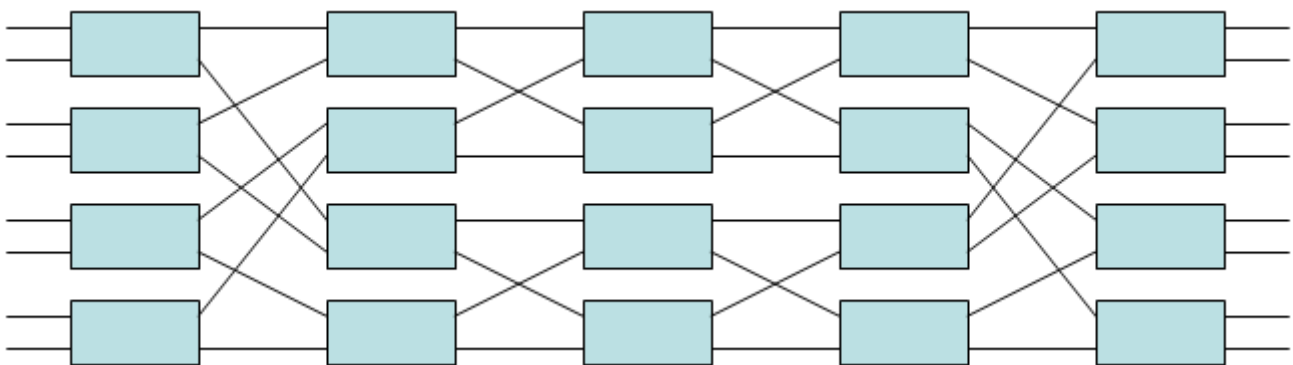
See also

- [Crossbar switch](), describes the switching element of a Clos network
- [Nonblocking minimal spanning switch](), describes the switching algorithm of a Clos network
- [Banyan switch](), an alternative way to connect networks
- [Fat tree](), an alternative way to connect networks
- [Omega network](), an alternative way to connect networks

References

1. US patent 2244004 (https://worldwide.espacenet.com/textdoc?DB=EPODOC&IDX=US2244 004)
2. Clos, Charles (Mar 1953). "A study of non-blocking switching networks" (https://archive.org/d etails/bstj32-2-406). *Bell System Technical Journal*. **32** (2): 406–424. doi:10.1002/j.1538-7305.1953.tb01433.x (https://doi.org/10.1002%2Fj.1538-7305.1953.tb01433.x). ISSN 0005-8580 (https://www.worldcat.org/issn/0005-8580).
3. Hogg, Scott (2014-01-11). "Clos Networks: What's Old Is New Again" (http://www.networkwo rld.com/article/2226122/cisco-subnet/clos-networks--what-s-old-is-new-again.html). Network World.
4. Moore, Samuel (31 October 2018). "Flex Logix Says It Has Solved Deep Learning's DRAM Problem" (https://spectrum.ieee.org/tech-talk/semiconductors/processors/flex-logix-says-its-s olved-deep-learnings-dram-problem). *spectrum.ieee.org*. IEEE Spectrum. Retrieved 1 November 2018.
5. Beneš, Václav E. (11 September 1965). *Mathematical Theory of Connecting Networks and Telephone Traffic*. Academic Press. ISBN 0-12-087550-0.
6. Hall, Philip (January 1935). "On Representatives of Subsets" (https://math.dartmouth.edu/arc hive/m38s12/public_html/sources/Hall1935.pdf) (PDF). *Journal of the London Mathematical Society*. s1. **10** (1): 26–30. doi:10.1112/jlms/s1-10.37.26 (https://doi.org/10.1112%2Fjlms%2 Fs1-10.37.26). Retrieved 2015-06-18.
7. Hui, Joseph Y. (1990). *Switching and Traffic Theory for Integrated Broadband Networks*. Kluwer Academic. ISBN 0-7923-9061-X.