Universität Karlsruhe (TH)
Forschungsuniversität · gegründet 1825

Netzwerk-Kenngrößen und -Topologien

Prof. Dr. Walter F. Tichy
Dr. Victor Pankratius
A. Jannesari

Fakultät für **Informatik**
Lehrstuhl für Programmiersysteme

Vorlesung „Cluster Computing“

- Architektur von Multikern-Rechnern und Rechnerbündeln
  → Hochgeschwindigkeitsnetzwerke
  (Für Multikern-Rechner wird es Netzwerke auf dem Chip geben.)
  - Theoretische Bewertungskriterien
  - Netztopologien
  - Praktische Bewertungskriterien
  - Vermittlungstechnik
- Hochgeschwindigkeitskommunikation

Rüstzeug

Netzwerktopologie =

- Geometrische Anordnung der Leitungen und Schalter, die Prozessoren und Speichermodule verbinden.
- Statisches Netz vs. dynamisches (mit Schaltern)

Vermittlungstechnik (Routing/Switching) =

- Wie und entlang welchen Pfades wird eine Nachricht vom Sender zum Empfänger übertragen?
- Gibt es spezielle Vermittlungs-Prozessoren, oder machen die Knoten selbst die Vermittlung?

Vorlesung „Cluster Computing“

- Architektur von Multikern-Rechnern
  Rechnerbündeln
  - → Hochgeschwindigkeitsnetzwerke
    (Für Multikern-Rechner wird es Netzwerke auf dem Chip geben.)
    - → Theoretische Bewertungskriterien
    - Netztopologien
    - Praktische Bewertungskriterien
    - Vermittlungstechnik
  - Hochgeschwindigkeitskommunikation

Bewertungskriterien für Netze

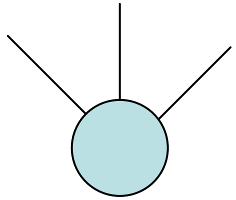
- Graph-Notation
  - Ungerichteter Kommunikationsgraph G=(V,E)
  - V = Knotenmenge
  - E = Menge der direkten bidirektionalen Verbindungen zwischen Knoten
  - $(v_0, ..., v_k)$ Pfad der Länge k zwischen Knoten 0 und k, wenn $(v_i,v_{i+1}) \in E$
- Theoretische Kriterien
  - Grad
  - Durchmesser
  - Kantenkonnektivität
  - Bisektionsbreite
- Praktische Kriterien
  - Latenzzeit
  - Bandbreite
  - Verzögerung/Übertragungszeit
  - Durchsatz

Grad (1)

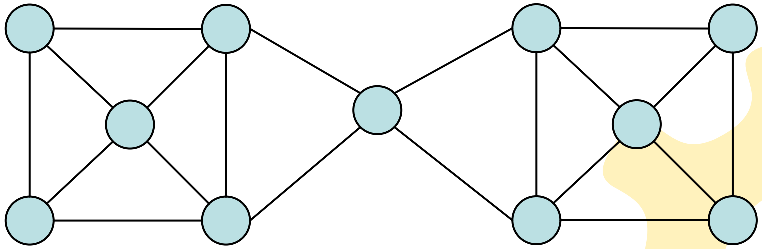
Grad eines Knoten =

Anzahl der adjazenten, d.h. ein- bzw. auslaufenden, Kanten des Knotens

Beispiel Grad 3:



Beispiel Grad 4:



Grad eines Netzes =

Maximaler Grad aller Knoten im Netz

Grad (2)

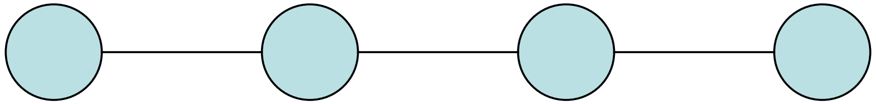
- Je höher der Grad,
  - desto mehr Parallelität und Bandbreite in der Kommunikation,
  - desto teurer ist die Kommunikationshardware; steigt wegen der Anzahl der Anschlüsse am Chip/Platine
  - Der Grad kann mit der Größe des Rechners nicht beliebig wachsen, da die Anzahl der Anschlüsse an ein Chip begrenzt ist (Rand).

- Ziel: kleiner Grad, um Kosten niedrig zu halten und Platz zu sparen.

Durchmesser (1)

Distanz eines Knotenpaares =

- Länge des kürzesten Pfades zwischen den zwei Knoten
- Wie viele Kanten muss eine Nachricht auf ihrem Weg zum Empfänger passieren?
- Optimistischer Ansatz: Die Vermittlung ist so gut, dass in der Regel der kürzeste Pfad verwendet wird.



Durchmesser 3

Durchmesser des Netzes =

Maximum der minimalen Distanzen aller Knotenpaare im Netz

Durchmesser (2)

- Je höher die Distanz zwischen zwei Knoten ist,
  - desto länger dauert die Kommunikation
  - desto kleiner ist die Wahrscheinlichkeit eines Ausfalls, weil mehr Knoten korrekt funktionieren müssen.

- Ziel: kleiner Durchmesser

Kantenkonnektivität (1)

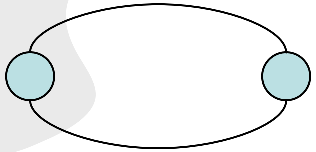
Kantenkonnektivität =

Minimale Anzahl von Kanten, die aus dem Netz entfernt werden müssen, um das Netz zu unterbrechen, d.h. in zwei unverbundene Teilnetze (evtl. unterschiedlicher Größe) zu zerlegen.

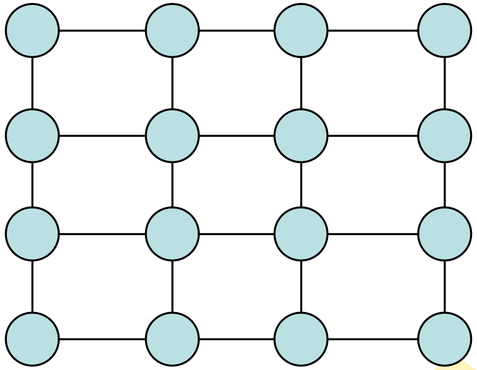
Kantenkonnektivität (2)

Beispiele mit Kantenkonnektivität 2:

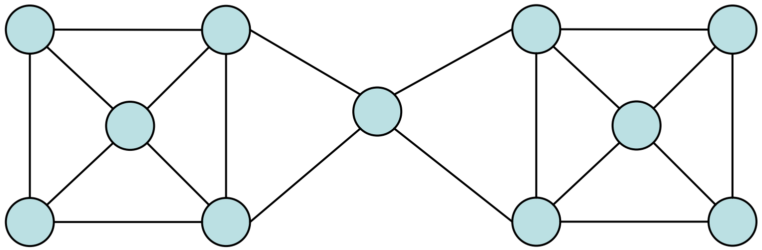
a)



c)



b)



Kantenkonnektivität (3)

- Je höher die Kantenkonnektivität,
  - desto mehr unabhängige Pfade zwischen je zwei Knoten gibt es
  - desto besser ist die Ausfallsicherheit, weil es über andere Pfade "Umwege" gibt.
  - desto schneller ist die Kommunikation, weil hoch belastete Stellen des Netzes (Staus) "umfahren" werden können.

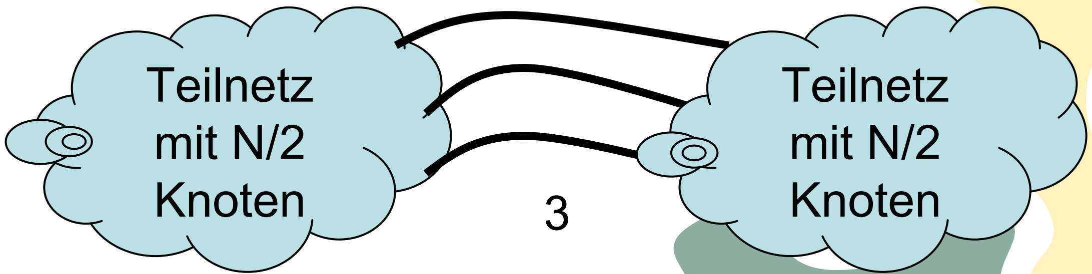
- Ziel: Hohe Kantenkonnektivität

Bisektionsbreite (1)

Bisektionsbreite =

- Minimale Anzahl von Kanten, die aus dem Netz entfernt werden müssen, um das Netz in zwei *gleichgroße* (±1) Teilnetze zu zerlegen
- Wenn eine Hälfte der Knoten an die Knoten der anderen Hälfte Nachrichten senden will, wie viele Botschaften können gleichzeitig im Netz unterwegs sein, ohne Leitungen mehrfach zu belegen?

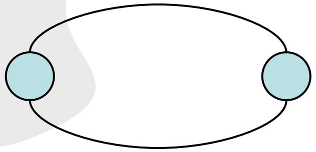
konzeptuelles Beispiel:



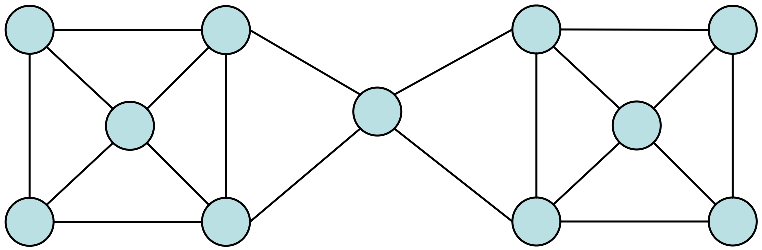
Bisektionsbreite (2)

Beispiele mit Bisektionsbreite 2:

a)



b)



Bisektionsbreite (3)

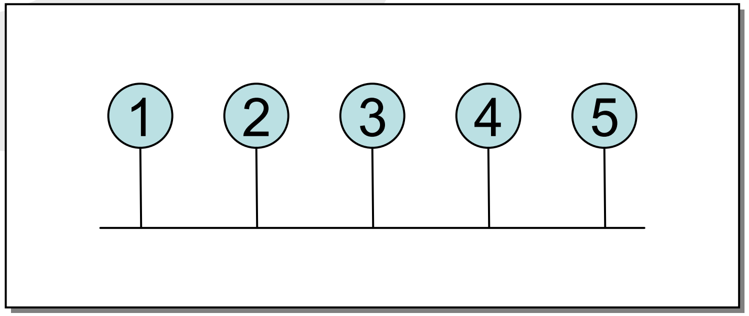
- Je kleiner die Bisektionsbreite ist,
  - desto weniger Nachrichten können das Netz sättigen, wenn diese einen solchen Flaschenhals überwinden müssen.
    Extremfall Ethernet: Bisektionsbreite 1, d.h. eine einzige Nachricht sättigt das Netz.

- Ziel: Hohe Bisektionsbreite, ideal: Knotenzahl/2

Vorlesung „Cluster Computing“

- Architektur von Multikern-Rechnern
  Rechnerbündeln
  → Hochgeschwindigkeitsnetzwerke
    (Für Multikern-Rechner wird es Netzwerke auf dem Chip geben.)
    - Theoretische Bewertungskriterien
    → Netztopologien
    - Praktische Bewertungskriterien
    - Vermittlungstechnik
  - Hochgeschwindigkeitskommunikation

Bus (z.B. Ethernet)



Einfachstes und preis-wertestes statisches Netz

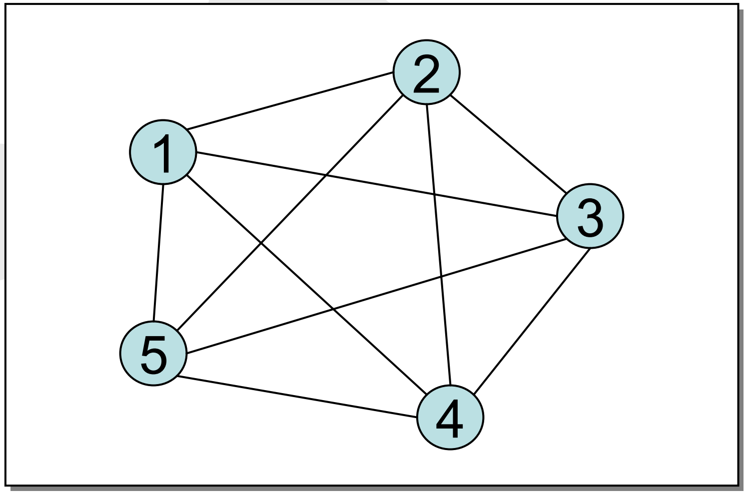
- Grad = 1
- Durchmesser = 1
  - Keine Wegewahl (Routing) erforderlich.
- Kantenkonnektivität = 1
- Bisektionsbreite = 1

CSMA/CD-Protokoll
(**C**arrier **S**ense -**M**ultiple **A**ccess / **C**ollision **D**etect)

Mindestpaketlänge,

Maximale Buslänge

Vollständiger Graph (Clique)



- Grad = n-1

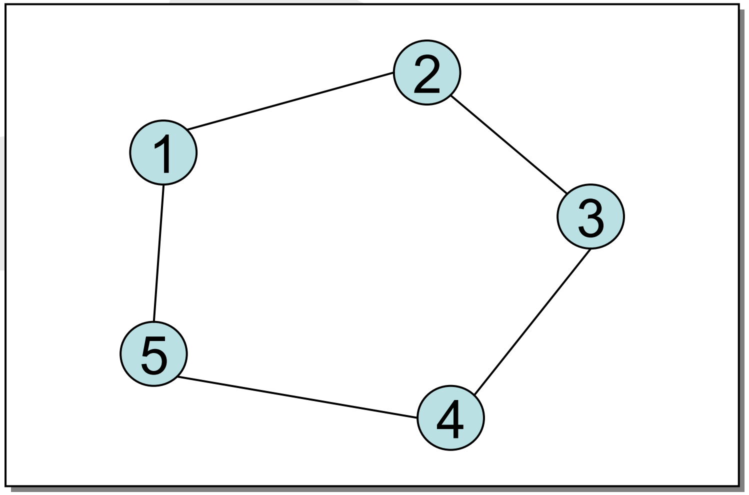
  zu teuer für große Netze
- Durchmesser = 1
- Kantenkonnektivität = n-1
- Bisektionsbreite = $\lfloor n/2 \rfloor \lceil n/2 \rceil$

Statisches Netz

Jeder Knoten ist mit jedem anderen Knoten direkt verbunden.

Bei Aufteilung der Knoten in zwei Hälften hat jeder Knoten Leitungen zu n/2 Knoten der anderen Hälfte. In einer Hälfte gibt es n/2 solche Knoten.

Ring



Statisches Netz

Jeder Knoten i ist verbunden mit den Knoten i+1 und i-1 modulo n.

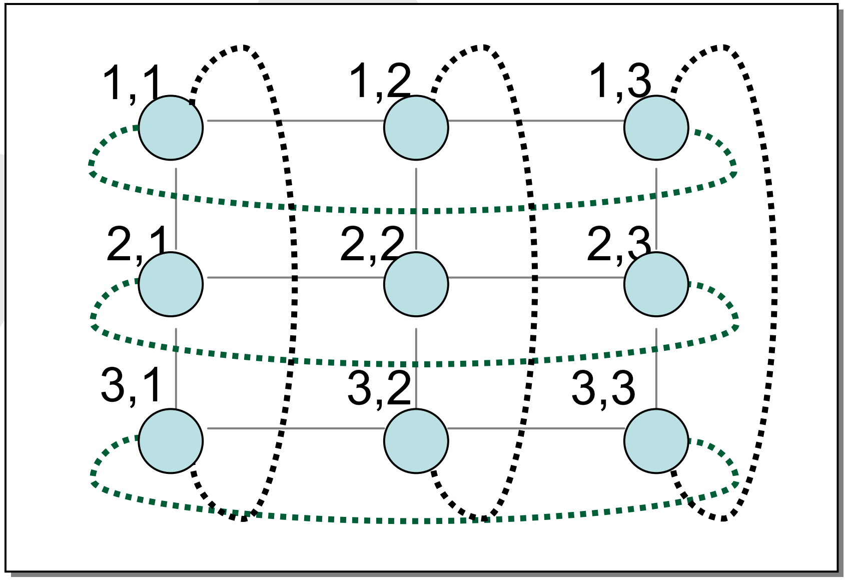
Verwendet z.B. bei der KSR-1

- Grad = 2
- Durchmesser = $\lfloor n/2 \rfloor$

  Bei großen Netzen zu langsam.
- Kantenkonnektivität = 2
- Bisektionsbreite = 2

  Zuverlässigkeitsprobleme?

d-dimensionales Gitter und Torus



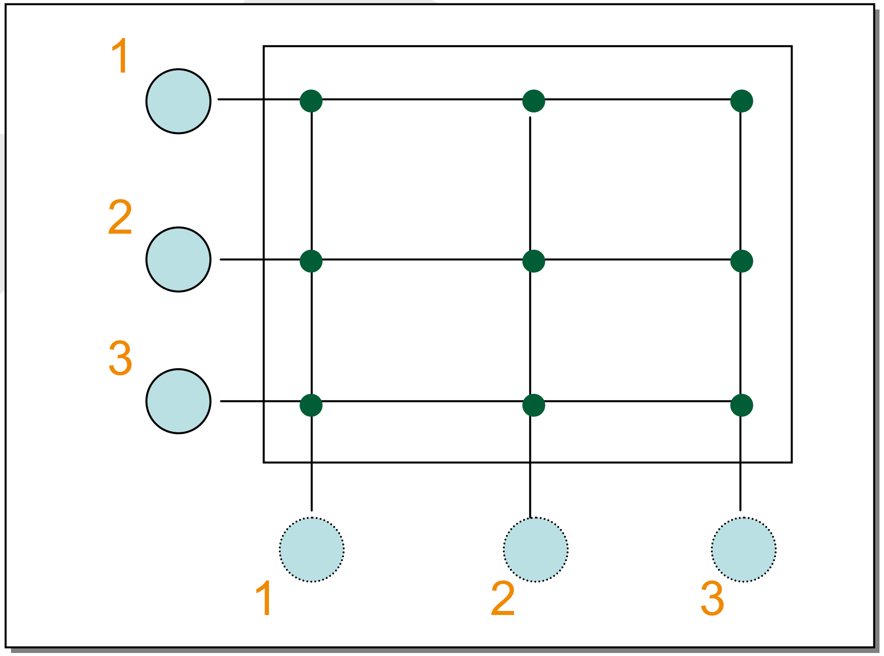
Statisches Netz.

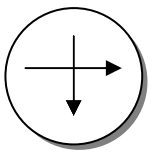
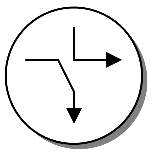
Bei Torus entlang jeder Dimension ringförmig verbunden.
Halber Durchmesser, doppelte Kantenkonnektivität, doppelte Bisektionsbreite. Cray T3D und T3E.

Gitter mit d gleichgroßen Dimensionen

- Grad = 2d
  - Zu teuer bei d>3
- Durchmesser = $d\,(\sqrt[d]{n} - 1)$
  - Entlang jeder Dimension höchstens $\sqrt[d]{n} - 1$ Schritte
- Kantenkonnektivität = d
  - An den Ecken reicht es, d Kanten aufzutrennen.
- Bisektionsbreite = $(\sqrt[d]{n})^{d-1}$
  - Entlang jeder Dimension (bis auf eine) $\sqrt[d]{n}$ Kanten

Kreuzschienenverteiler (Crossbar)



Dabei sind die ● umschaltbar zwischen  und 

Dynamisches Netz, einstufig

n Prozessoren, schnell und teuer wegen n^2 Schaltern

Meist: Prozessor x Speicher

- Grad = 1 (bidirektional)
- Durchmesser = 2

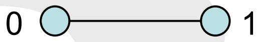
  Ganze Kreuzschiene als eine Einheit betrachten.
- Kantenkonnektivität = 1
- Bisektionsbreite = n/2

Auf 1 Chip: 4x4, 8x8, 16x16

Hyperwürfel (1)

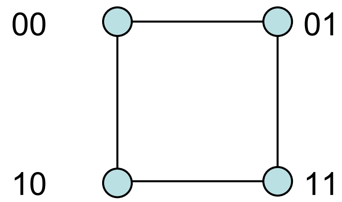
Konstruktion eines Hyperwürfels:

Dim: 1

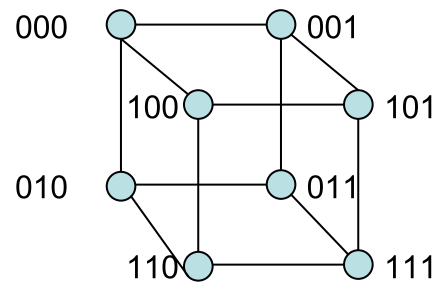


- Verdoppeln
- Bit ergänzen
- korrespondierende Kanten verbinden

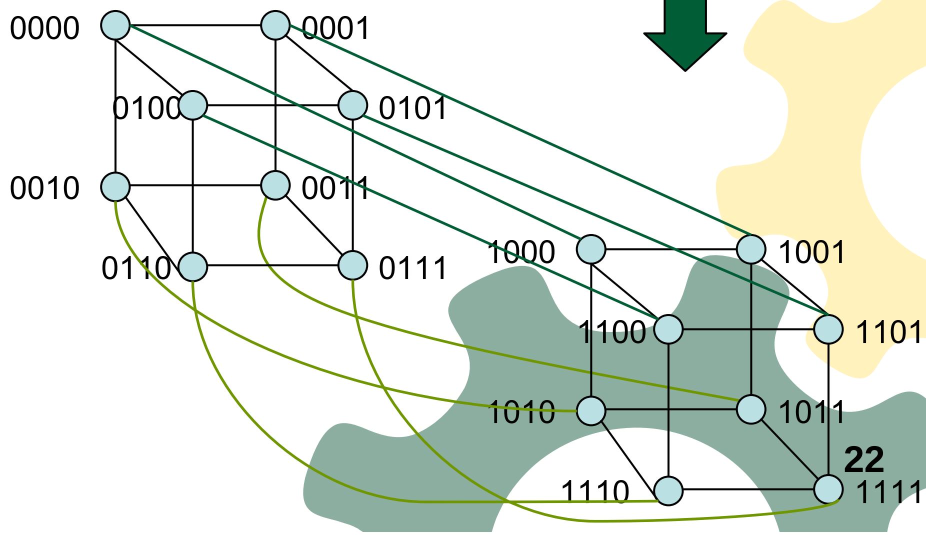
Dim: 2

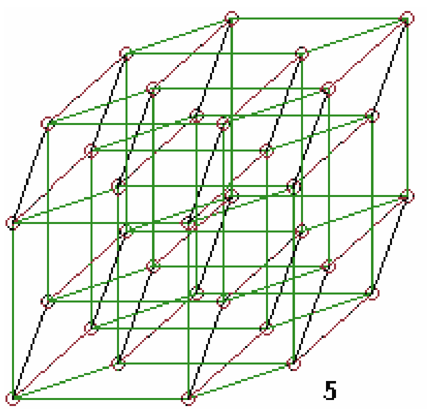


Dim: 3

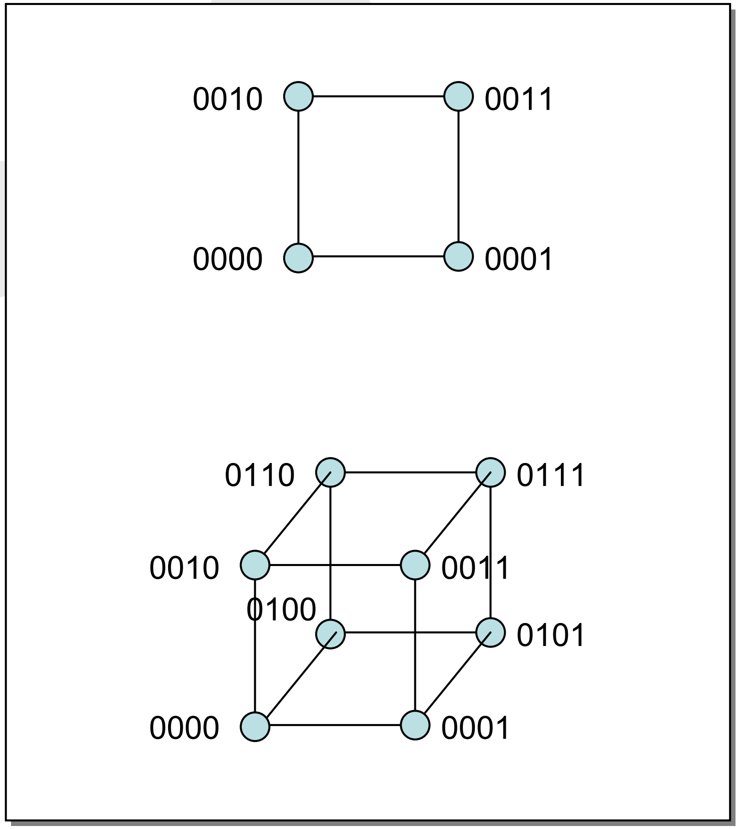


Dim: 4





Hyperwürfel (2)



Statisches Netz

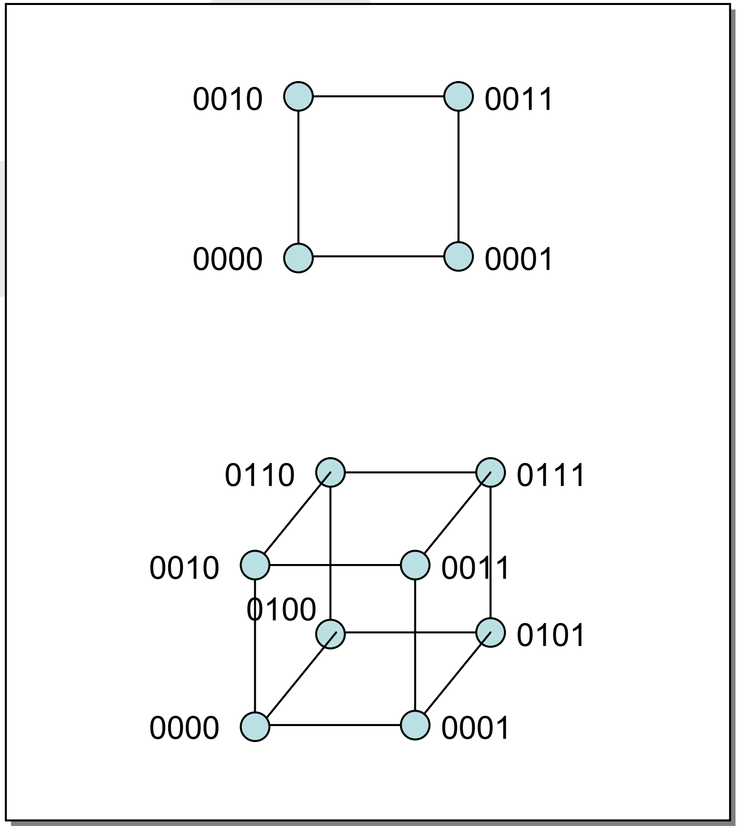
Hamming-Distanz =

Anzahl der Bits, die sich in zwei gleichlangen binären Worten unterscheiden

Im Hyperwürfel sind zwei Knoten direkt miteinander verbunden, wenn die Hamming-Distanz der Knotennummern 1 ist.

Wegewahl gemäß Anpassung von Start- an Zieladresse.

Hyperwürfel (3)



Intel iPSC/860,
SGI Origin 2000

k Dimensionen, n= 2^k Knoten:

- Grad = k
- Durchmesser = k
- Kantenkonnektivität = k
- Bisektionsbreite = n/2

  Zwei Hyperwürfel halber Größe werden durch n/2 Kanten miteinander verbunden.

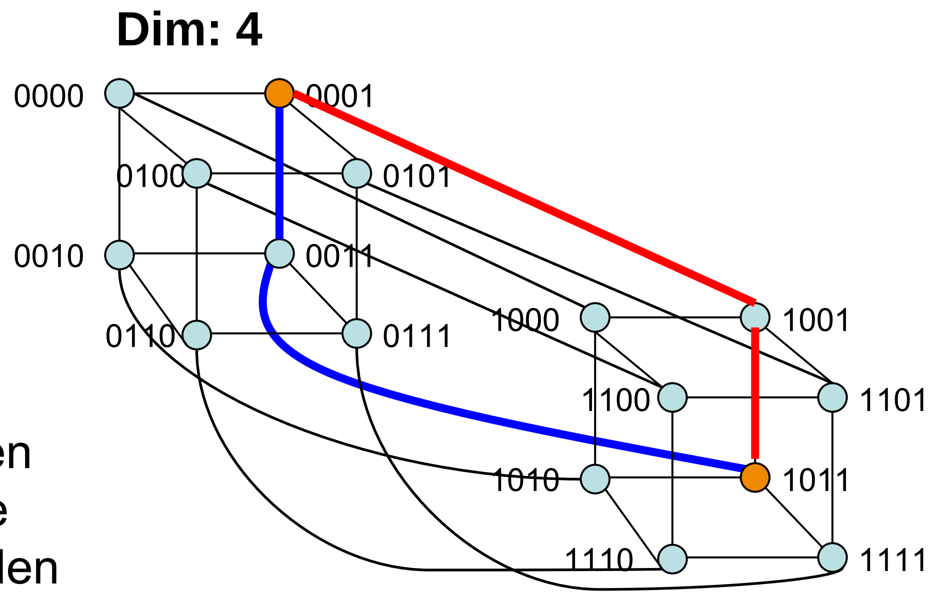
Hyperwürfel (4)

Routing im Hyperwürfel:

Beispiel:

Von : 1011
Nach : 0001
============
XOR : 1010

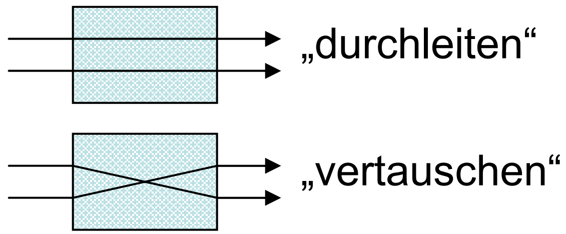
- Gibt an, welche Dimensionen „nicht stimmen“ (d.h. welche Dimension gewechselt werden muss).
- Hier: 2 mögliche Wege.
- Anzahl Einsen entspricht erforderlicher Weglänge



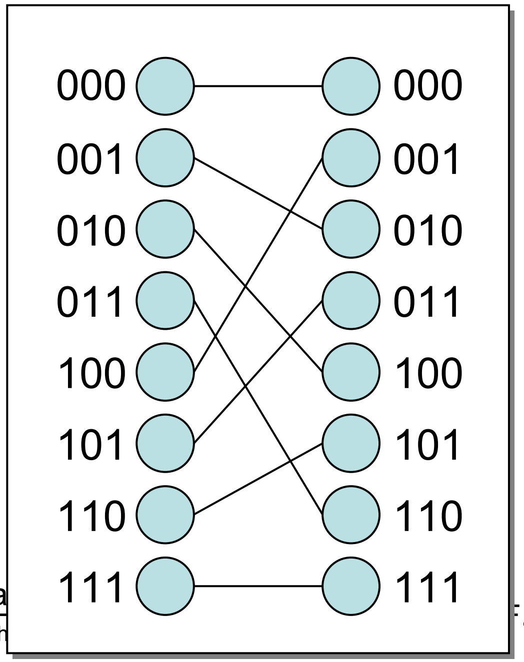
Omega-Netz (1) - Vorüberlegungen

- 2x2 Shuffle-Bausteine

  Schalterstellungen, die ein Schalter in einem Omega-Netzwerk realisieren kann:



- Verbindung der Schalter nach folgendem Schema:

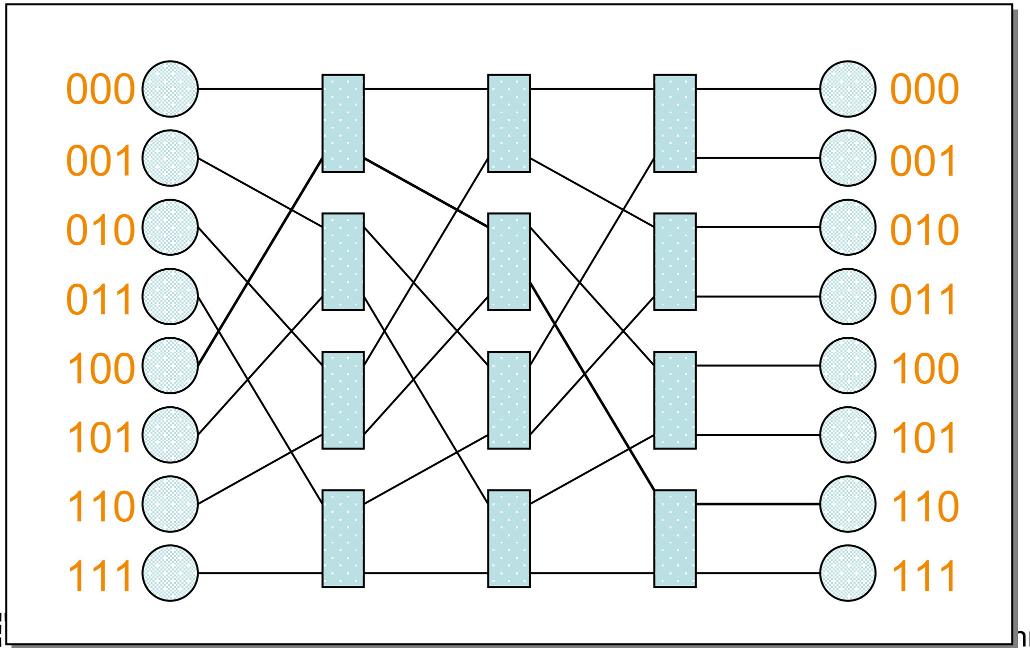


- Perfect Shuffle (Mischungspermutation):

  Zieladresse (rechte Seite) ergibt sich aus zyklischen Links-Shift von Adresse auf der linken Seite

Omega-Netz (2) - Definition

- Ein nxn Omega-Netz besteht aus
  - 2x2 –Crossbar-Schaltern,
  - die in $Log_2 n$ Stufen angeordnet sind, wobei
  - jede Stufe n/2 Schalter enthält und
  - jeder Schalter 2 Eingänge und 2 Ausgänge hat.

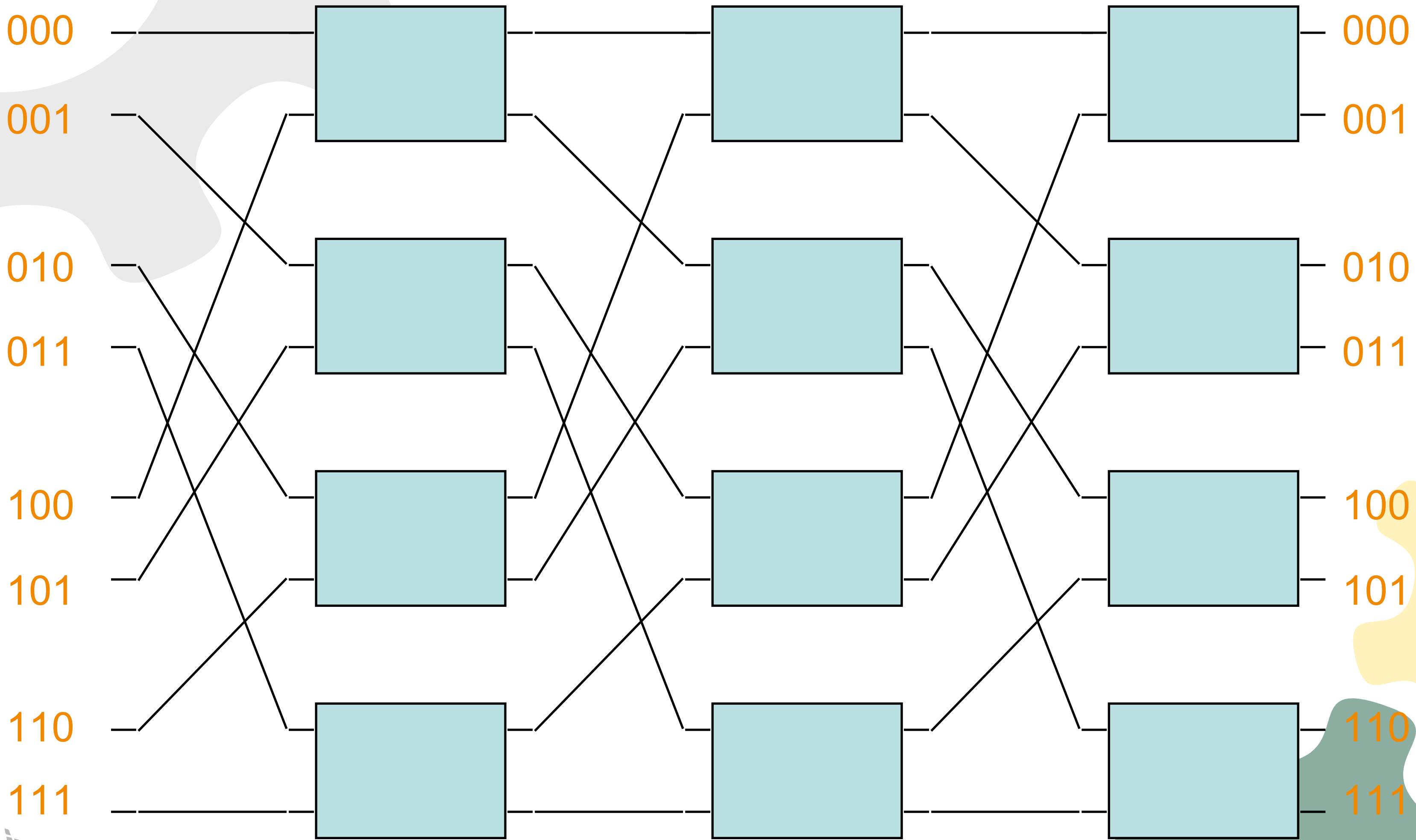


- *Beispiel: 8x8 Omega-Netz:*
  - *$Log_2 8$ = 3 Stufen*
  - *8/2=4 Schalter pro Stufe*
  - *Insgesamt 3*4=12 Schalter*

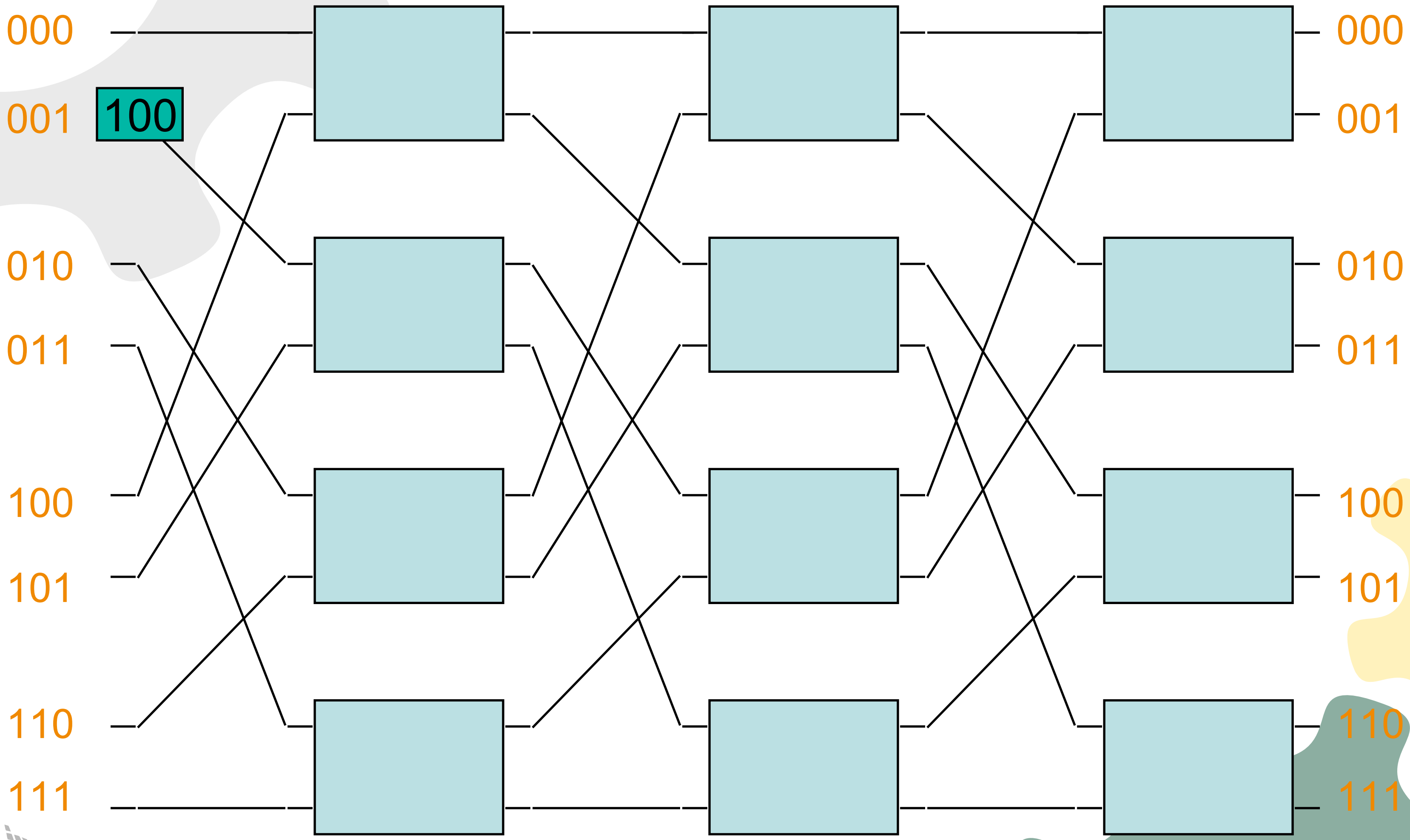
Omega-Netz (3) - Routing

- Dynamisch: Jeder Schalter leitet Nachricht ohne Koordination mit anderen Schaltern weiter
- Ein Schalter auf Stufe k, der Nachricht für Zieladresse z erhält, wertet k-tes Bit z_k (von links) aus und wählt Ausgang nach folgenden Regeln:
  - Ist das k-te Bit $z_k=0$, wird Nachricht über den oberen Ausgang weitergeleitet
  - Ist das k-te Bit $z_k=1$, wird Nachricht über den unteren Ausgang weitergeleitet

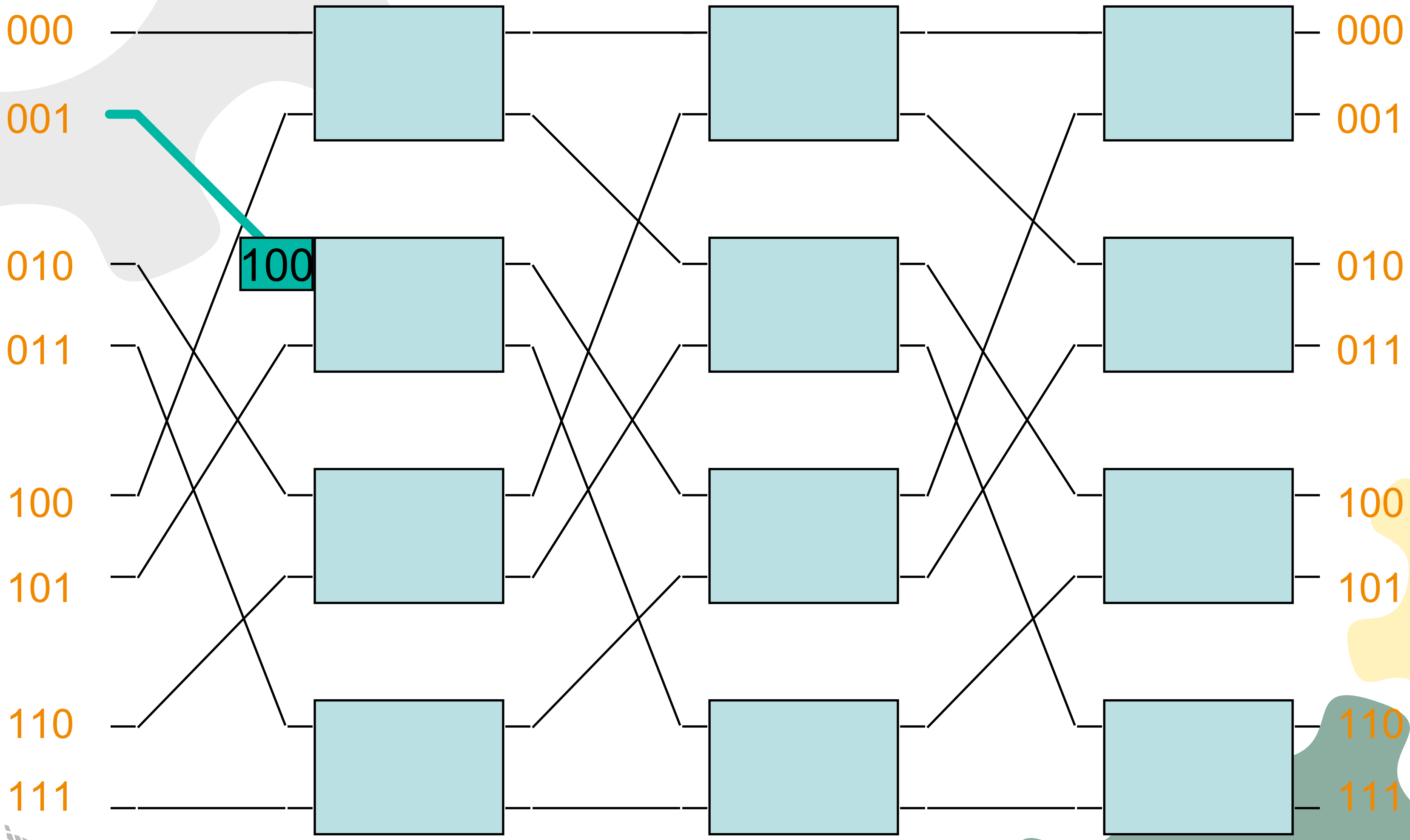
Omega-Netz (3) – Beispiel Output auf Port 4 ($=100_2$)



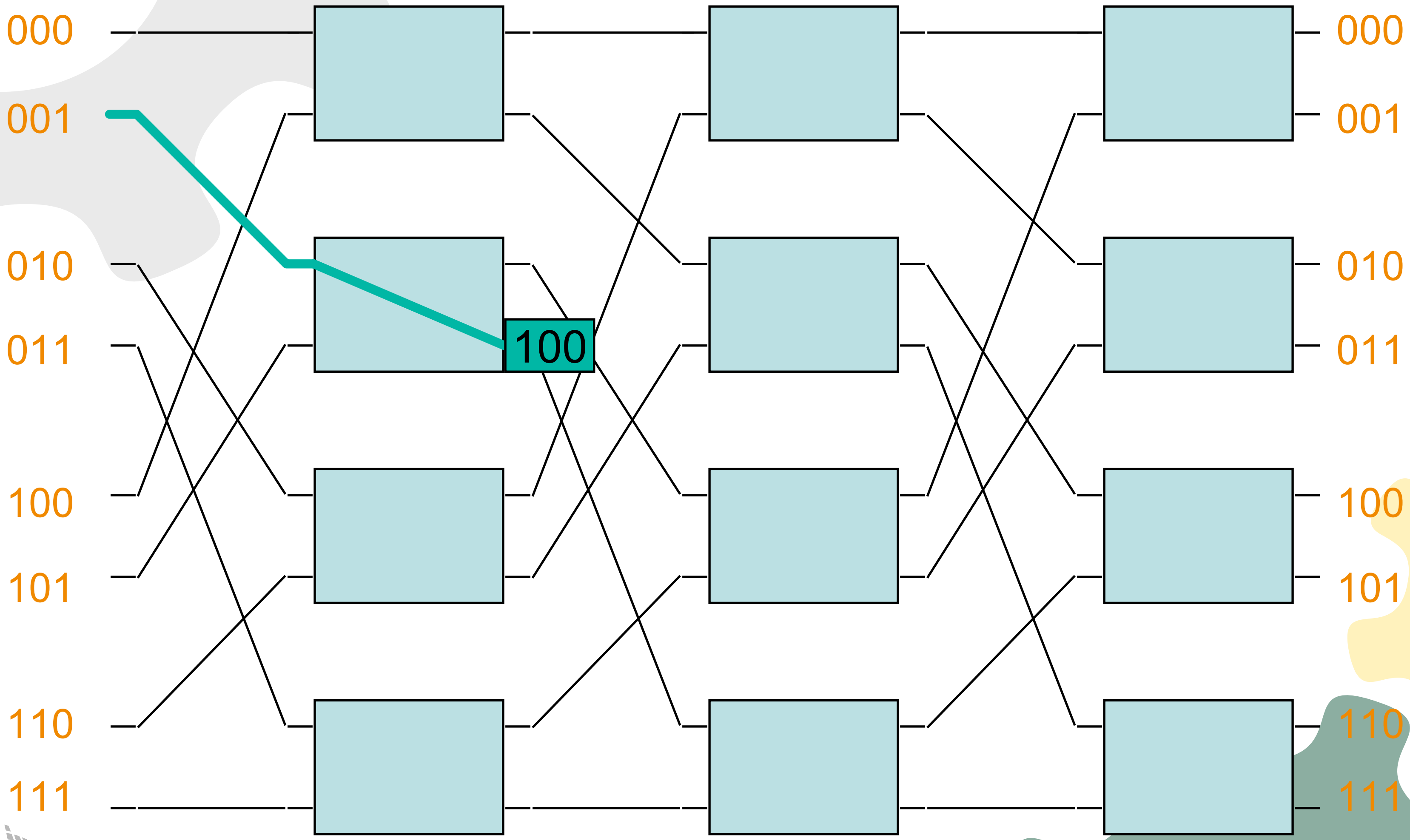
Omega-Netz (3) – Beispiel
Output auf Port 4 (=100_2)



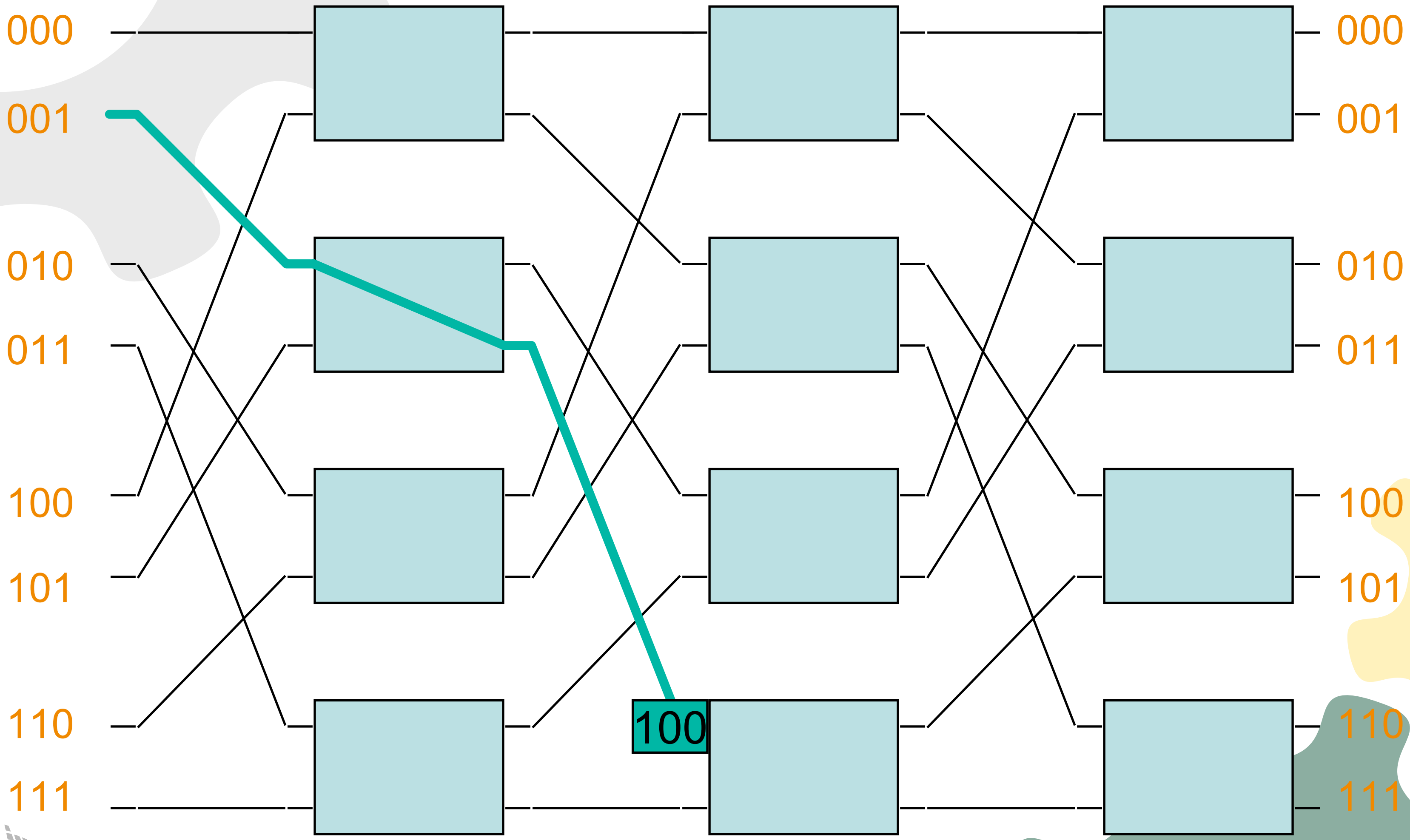
Omega-Netz (3) – Beispiel
Output auf Port 4 (=100_2)



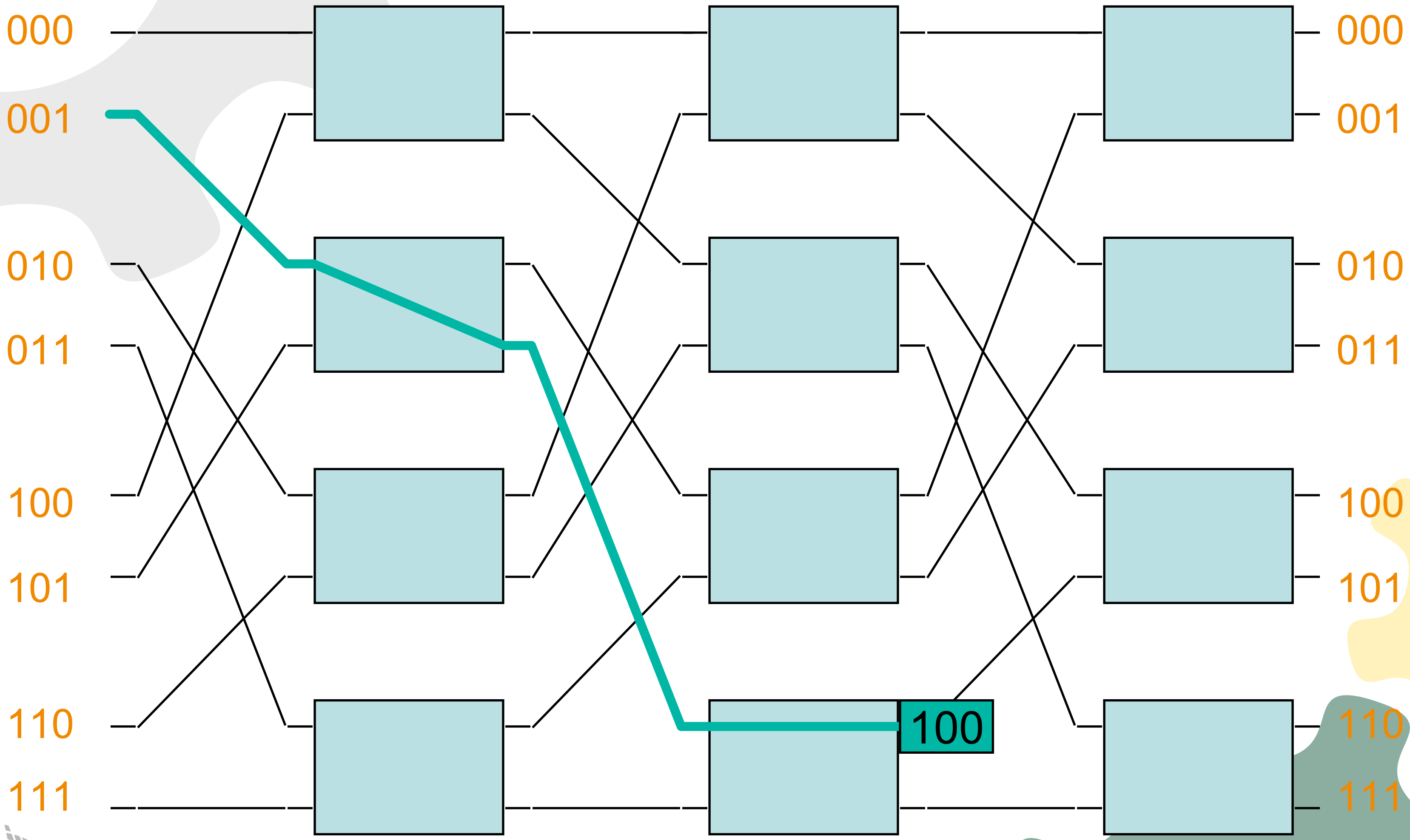
Omega-Netz (3) – Beispiel
Output auf Port 4 ($=100_2$)



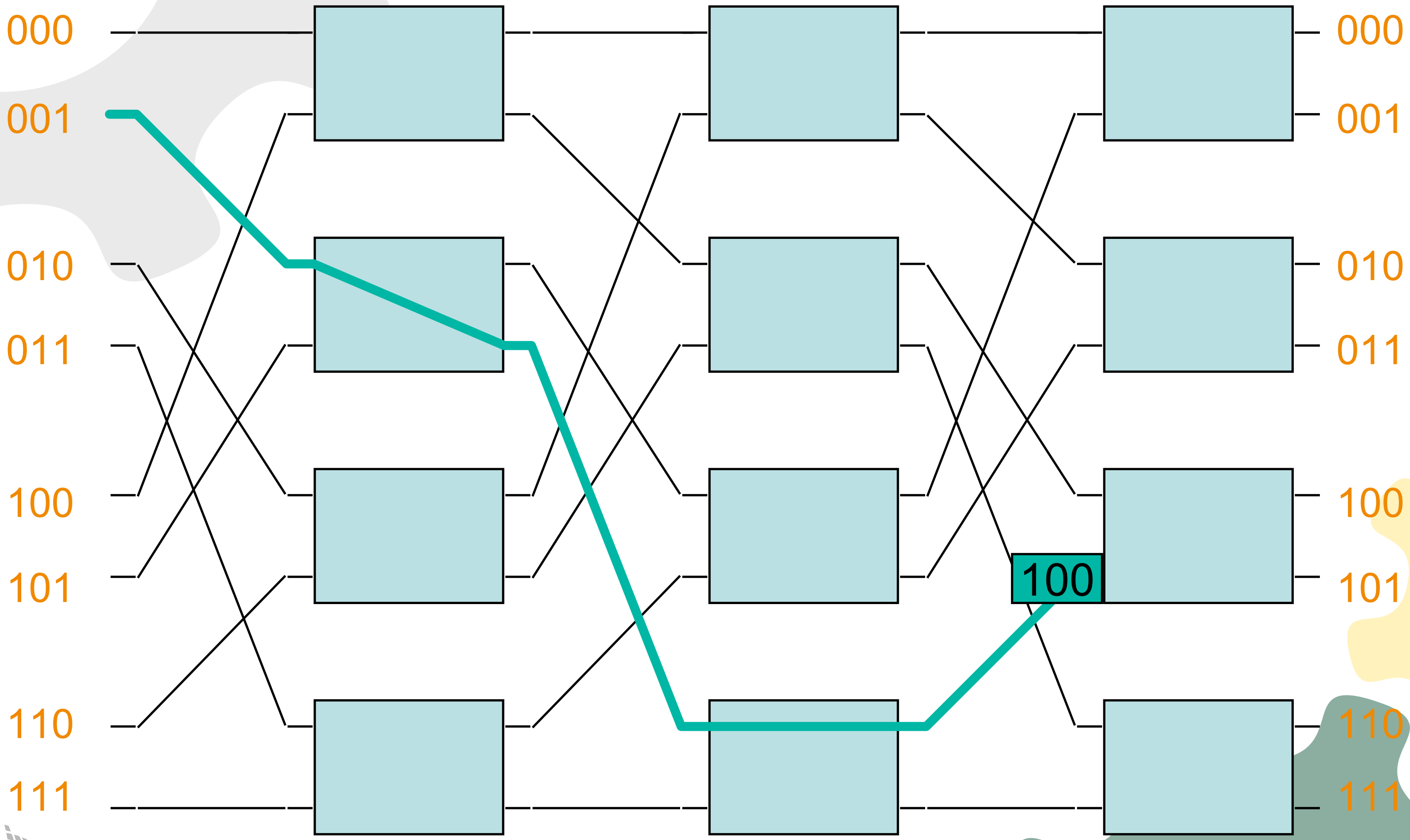
Omega-Netz (3) – Beispiel Output auf Port 4 ($=100_2$)



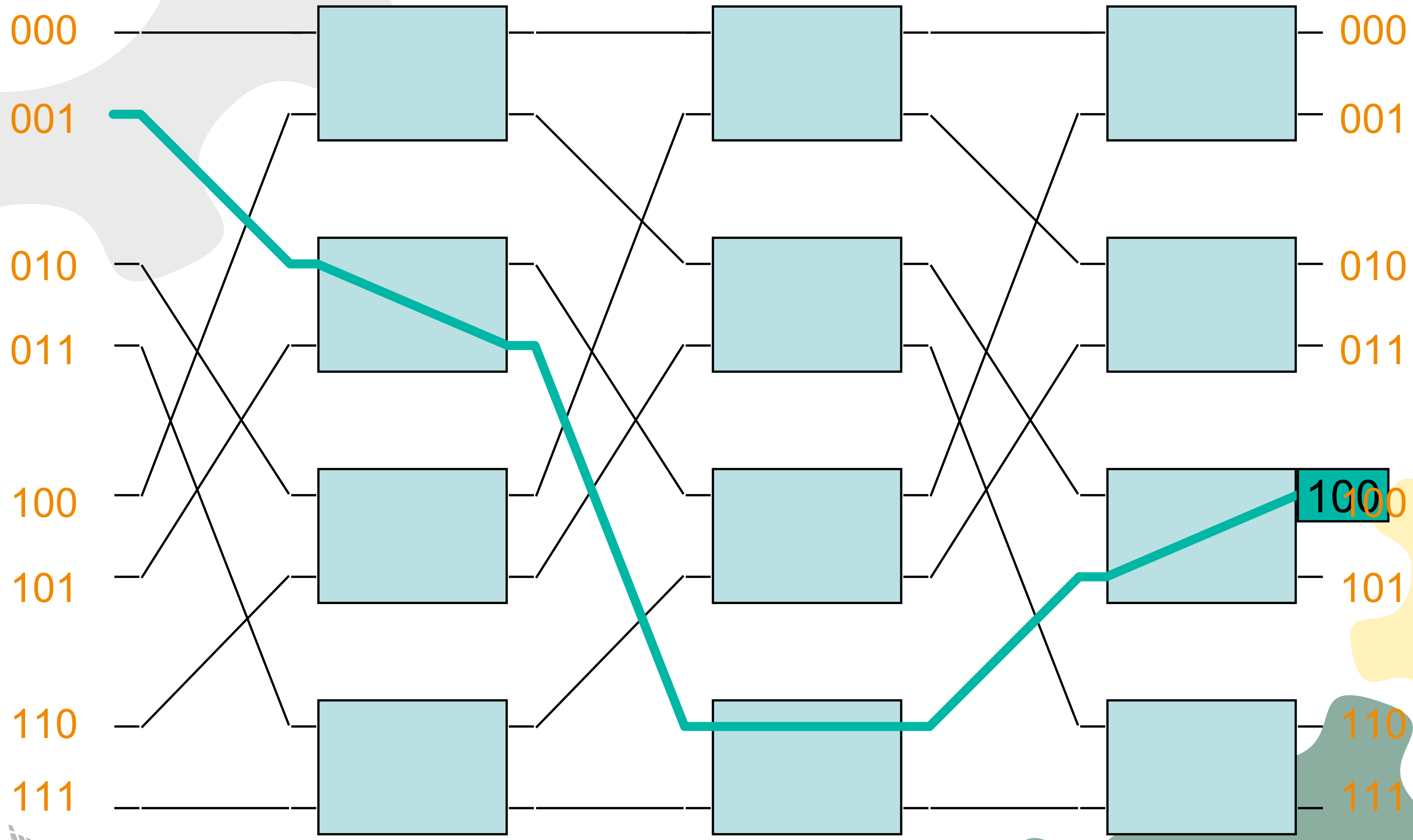
Omega-Netz (3) – Beispiel
Output auf Port 4 (=100_2)



Omega-Netz (3) – Beispiel
Output auf Port 4 ($=100_2$)



Omega-Netz (3) – Beispiel
Output auf Port 4 ($=100_2$)



Omega-Netz (3)

n Knoten, $(n/2) \log_2 n$ Kreuzschienenverteiler

- Grad = 2 für Knoten, 4 für Kreuzschienenverteiler
- Durchmesser = $\log_2 n$
  - Alle $\log_2 n$ Schichten müssen durchlaufen werden
- Kantenkonnektivität = 2
  - Abtrennung eines Knotens
- Bisektionsbreite = n/2
  - n/2 Nachrichten gleichzeitig möglich (wenn auch nicht bei allen Permutationen!).
- Auch einstufige Variante (mit log n Durchläufen) möglich wegen gleichem Verdrahtungsplan zwischen den Stufen

Clos-Netz (1)

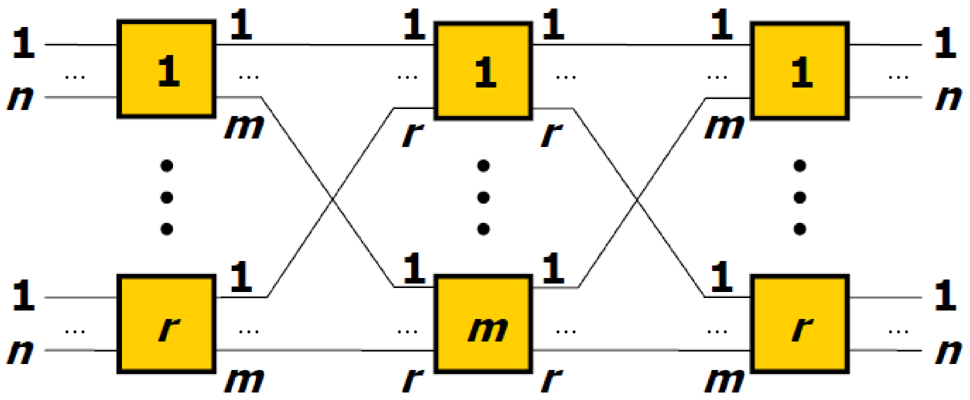
- Netzwerktopologie, entwickelt 1953 von Charles Clos für Telefonverbindungsnetzwerke.

  "A study of non-blocking switching networks ",
  Bell System Technical Journal, Vol. 32
- Geschaltetes, nicht-blockierendes Netzwerk.
- Bis dahin: Einsatz von Kreuzschienen zur Verbindung jedes Teilnehmers mit jedem anderen Teilnehmer.
- Große Kreuzschienenverteiler sind jedoch unpraktisch und nicht effizient:
  - Tatsächlich bogen sich die langen Metallschienen durch, so dass die Kontakte nicht mehr ausgerichtet waren.
  - Alle Kontakte einer Schiene bis auf einen bleiben immer frei
    → nicht effiziente Ressourcennutzung.

Clos-Netz (2)

- Idee des Clos-Netzes:
  - Ersetze den Kreuzschienenverteiler durch drei Schichten mit kleineren Kreuzschienenverteilern.
  - Eingangs- und Ausgangsschicht: r Verteiler vom Typ n×m bzw. m×n
  - mittlere Schicht: m Verteiler vom Typ r×r
- Das Theorem von Clos besagt, dass ein solches Netzwerk genau dann nicht-blockierend im strengen Sinne ist, wenn gilt: $m \geq 2n-1$.

r·n Eingänge

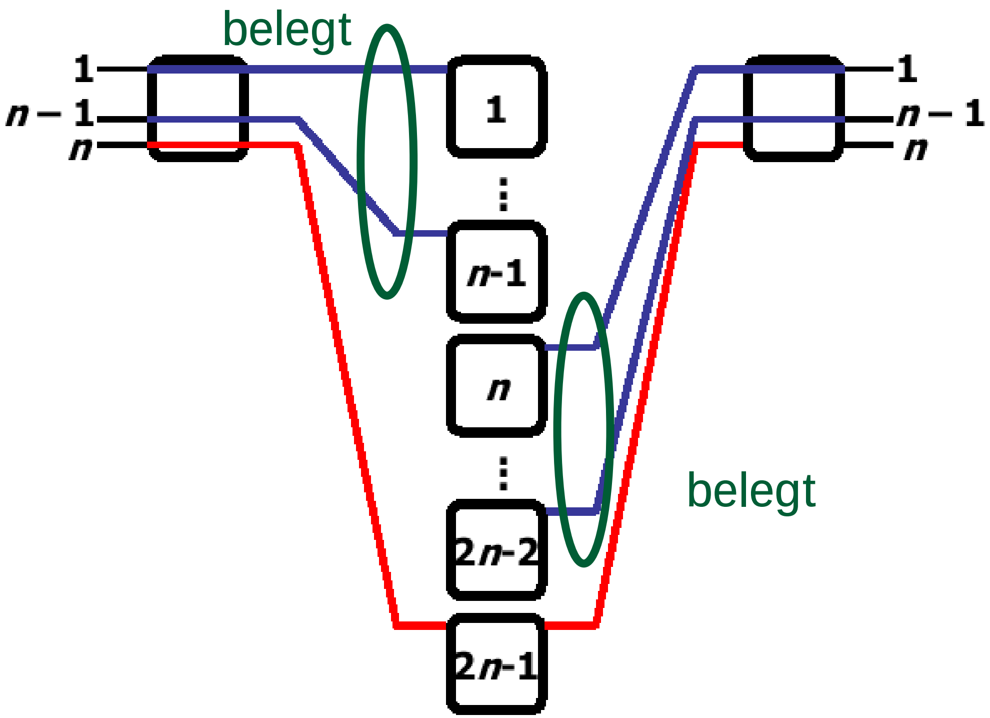


r·n Ausgänge

- Bei großen Netzwerken deutliche Reduzierung der Schaltkontakte gegenüber einzelnem Kreuzschienenverteiler.

Clos-Netz (3)

- Zum Beweis des Theorems von Clos betrachten wir den schlimmsten Fall, der bei der Leitungswahl auftreten kann.

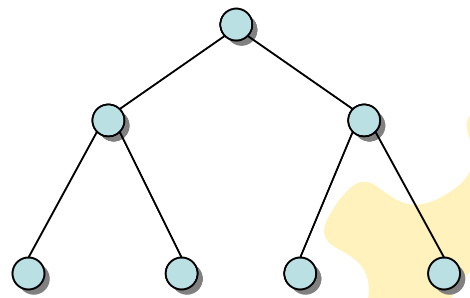


- Der Übersicht wegen sind jeweils nur ein Verteiler der Ein- und der Ausgangsschicht gezeichnet.

Fat Tree (1)

- Modell eines geschalteten Netzwerkes für Parallelcomputer
- beschrieben 1985 von Charles Leiserson

  "Fat-Trees: Universal Networks for Hardware Efficient Supercomputing" in: IEEE Transactions on Computers C-34(10).
- Ausgangspunkt: Binärbaum
  - Blätter: Rechenknoten
  - Kanten: Netzwerkverbindungen (bidirektional!)
  - Knoten: Schalter



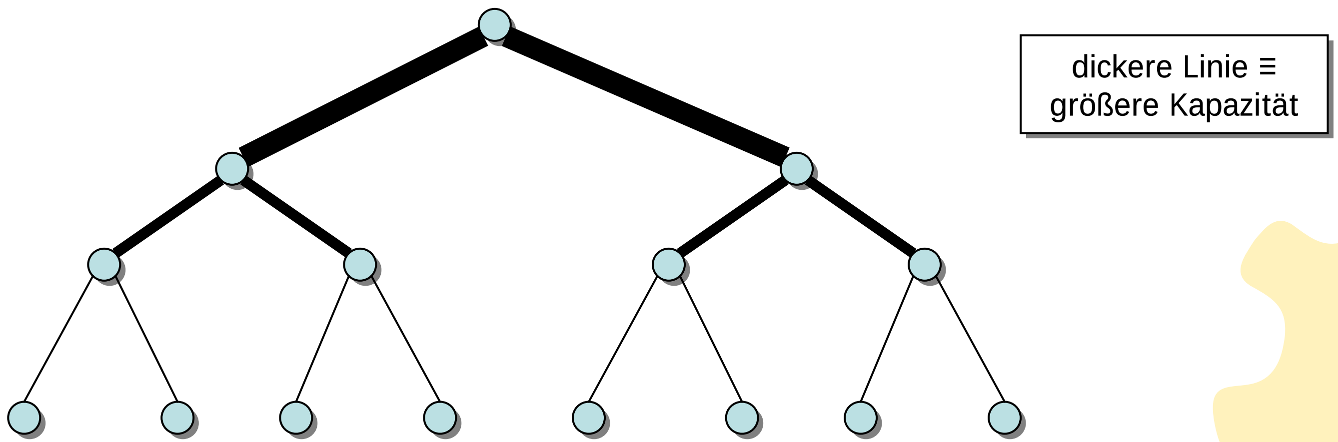
- Baum hat Durchmesser $2\log_2 n$

  „von ganz links über die Wurzel nach ganz rechts“
- Ein einfacher Baum hätte eine Bisektionsbreite von Eins

  (der Wurzelknoten stellt einen Flaschenhals dar)

Fat Tree (2)

- Idee des Fat Tree:
  - Gebe den Kanten auf Ebene i (0=Blätter) eine höhere Kapazität als den Kanten auf Ebene (i-1).
  - Wenn man die Kapazität in jeder Ebene verdoppelt, erhält man ein Netz mit der Bisektionsbreite n/2.



- Wegewahl:

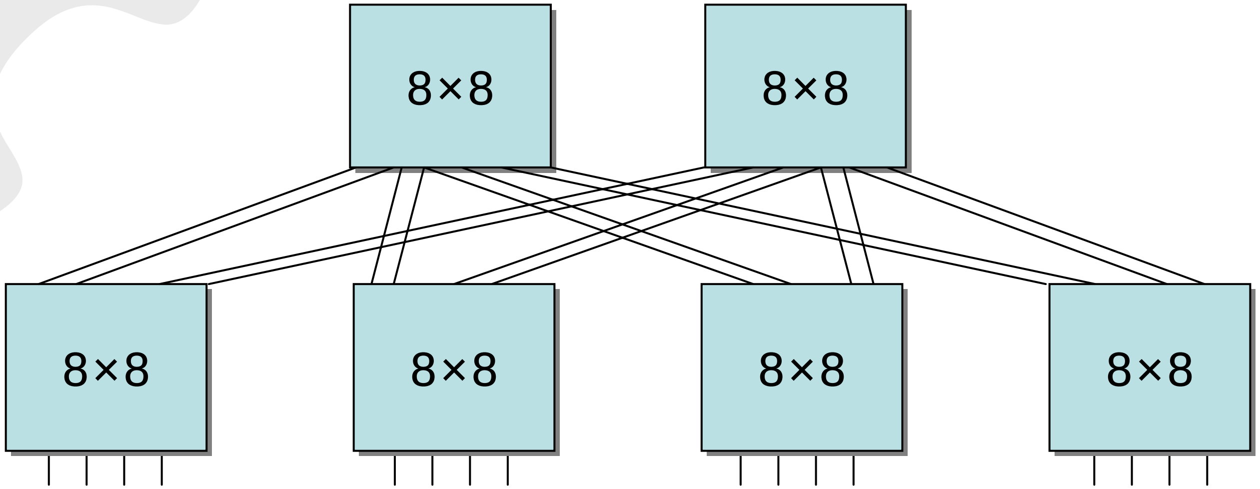
  auf direktem Weg über den niedrigsten gemeinsamen Vorfahren

CBB-Netzwerk (1)

- CBB steht für Constant Bisectional Bandwith.
- Verallgemeinerung der Konzepte von Clos-Netz und Fat-Tree:
  - Konstruiere ein nicht-blockierendes Netzwerk für M Endknoten mittels N×N Kreuzschienenverteilern (mit M>N).
- Konstruktionsschema
  - In jeder Ebene des Netzwerkes muss die Bandbreite “nach unten” (in Richtung der Endknoten) gleich der Bandbreite “nach oben” (in Richtung der Wurzel) sein.
- Beispielproblem: (siehe nächste Folie)
  - Verbinde 16 Knoten mit einem CBB-Netz aus 8×8 Switchen.

CBB-Netzwerk (2)

- Beispiel – Auflösung: Es werden 6 Schalter mit insgesamt 48 Anschlüssen benötigt.



- generell gilt für 2-stufige CBB-Netze: Man braucht 3n Anschlüsse für n Knoten.
  - Für jeden Knoten werden zwei Anschlüsse in der ersten und einer in der zweiten Ebene benötigt.
- Anmerkung: kein Clos-Netz
  - Ein Clos-Netz hätte 7 4×4 Verteiler in der zweiten Ebene.
- Merke: Nur ein Beispiel! CBB Netze können natürlich auch gewünscht werden für eine andere Anzahl Knoten und andere Kreuzschienenverteiler-Bausteine!

Vorlesung „Cluster Computing“

- Architektur von Multikern-Rechnern
  Rechnerbündeln

  → Hochgeschwindigkeitsnetzwerke

  (Für Multikern-Rechner wird es Netzwerke auf dem Chip geben.)

  - Theoretische Bewertungskriterien
  - Netztopologien

  → Praktische Bewertungskriterien

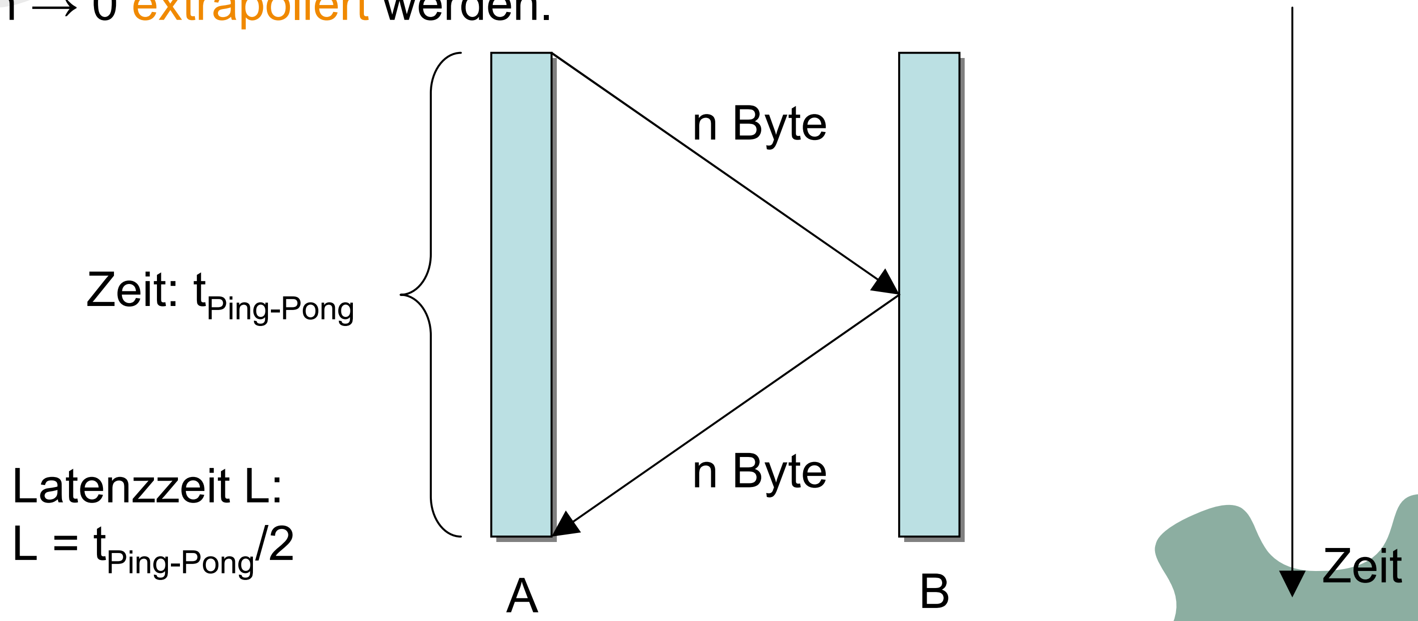
  - Vermittlungstechnik
- Hochgeschwindigkeitskommunikation

Latenzzeit L (Messung)

- Verzögerung einer Kommunikation (Zeit zw. Absenden und Ankunft des Paketanfangs)
- Einheit: Sekunde
- Übliches Vorgehen für Messung:
  - Eine Botschaft ohne Nutzinhalt wird per "ping-pong" ausgetauscht
  - Vorlaufbotschaften, um Aufwärmeffekte zu ignorieren
  - Viele Iterationen durchführen, wegen Taktung d. Uhren
  - Latenzzeit = (ping-pong-Zeit / Iterationszahl) / 2
- Wenn mehr als ein Prozessorpaar gleichzeitig ping-pong durchführt, dann steigt in Netzen mit kleiner Bisektionsbreite die gemessene Latenz.

Messungen: Ping-Pong

- Prozess A schickt ein Datenpaket der Größe n Byte zu Prozess B, der es zurückschickt. A wartet, bis das Paket zurückkommt.
- Verwende kleinstmögliches Paket.
  - Wenn die Latenz als Verzögerung der Nullnachricht definiert ist (anders als hier), aber keine Nullnachrichten gesendet werden können, muss für $n \rightarrow 0$ extrapoliert werden.



Bandbreite B

- Wie viele Daten sind pro Zeit bestenfalls übertragbar?
- Einheit: Byte/Sekunde
- Probleme der Messung:
  - Nur bei asymptotisch großen Datenmengen kann die immer auftretende Latenzzeit ignoriert werden.
  - Manche Kommunikationsprotokolle erlauben aber nur kleine Paketgrößen fester Länge
  - Daher „Burst"-Ansatz: schnelle Folge maximal großer Datenpakete versenden.
  - Zwei Alternativen für „Pong":
    - Paket zurücksenden: dann gemessene Zeit halbieren.
    - Leerbotschaft: dann Dauer der Antwort wegen asymptotischer Länge der Nutzbotschaft ignorieren.
  - Unterschiedliche Ergebnisse!
- Bandbreite ist abhängig von Bisektionsbreite.

Verzögerung V(G): Übertragungszeit

- Die Verzögerung V ist abhängig von Größe G der Botschaft.
- Wenn die Laufzeit proportional zur Botschaftsgröße ist, gilt folgende Berechnungsvorschrift:

  $$V(G) = L + (G-G_{min})/B$$

  Dabei ist G_{min} die Länge der minimalen Botschaft:

  $$V(G_{min}) = L$$

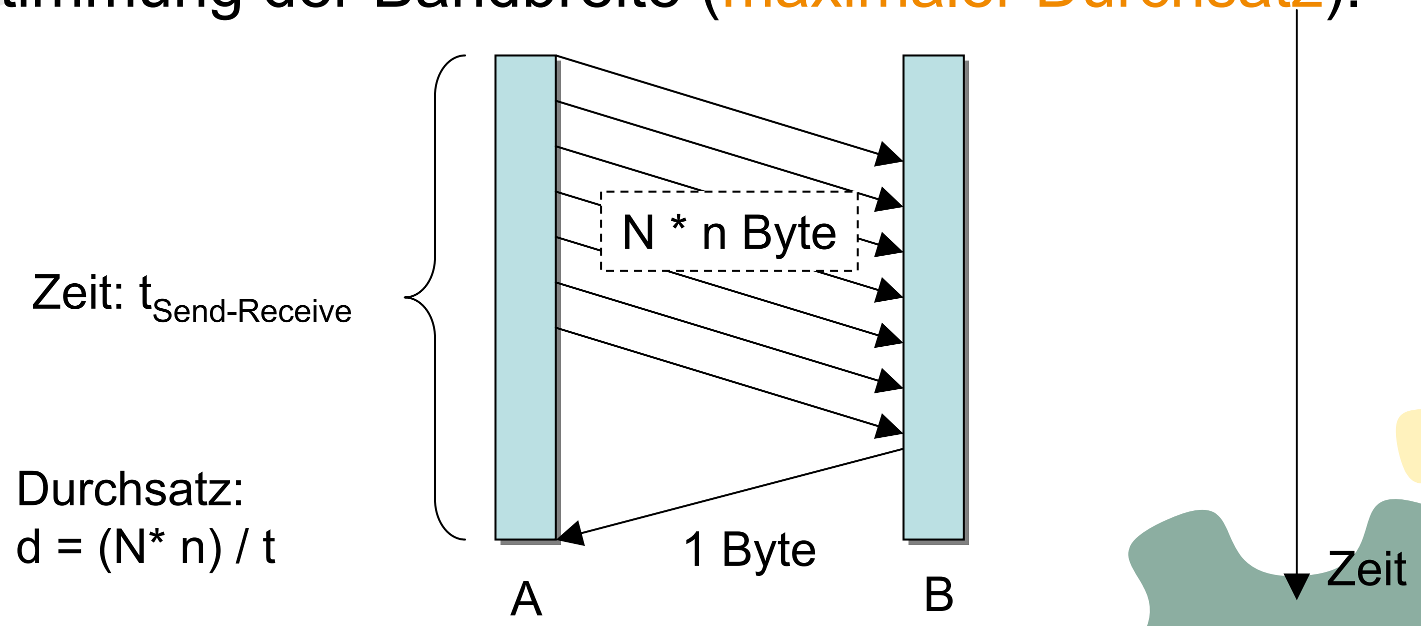
  Einheit: Sekunde
- Falls Pufferungen oder Paketfragmentierungen nötig sind, dann ist Verzögerung durch Treppenfunktion bestimmt.

Durchsatz D(G)

- Bandbreite = Durchsatz(G_{max})
- Einheit: Byte/Sekunde
- Meist wird die Bandbreite aber bei üblichen Botschaftsgrößen (bei weitem) nicht erreicht.
- Durchsatz ist interessanteres Maß: Verhältnis von Botschaftsgröße zur zugehörigen Verzögerung
  $D(G) = G/V(G)$
- Weil graphisch Darstellung von D(G) aufwändig und schwer vergleichbar ist, knappe Kurzcharakterisierung „half-power-point“: Mit welcher Paketgröße G_h wird die Hälfte der Bandbreite erreicht? $D(G_h) = B/2$.

Messungen: Send-Receive

- A sendet viele Pakete, ohne auf die Bestätigung von B zu warten. Erst am Ende quittiert B den Empfang aller Pakete.
- Messung von unidirektionalem Durchsatz.
- Bestimmung der Bandbreite (maximaler Durchsatz).



Vorlesung „Cluster Computing“

- Architektur von Multikern-Rechnern
  Rechnerbündeln
  → Hochgeschwindigkeitsnetzwerke
    (Für Multikern-Rechner wird es Netzwerke auf dem Chip geben.)
    - Theoretische Bewertungskriterien
    - Netztopologien
    - Praktische Bewertungskriterien

    → Vermittlungstechnik
  - Hochgeschwindigkeitskommunikation

Vermittlungstechnik

- Besteht zwischen zwei Knoten keine Punkt-zu-Punkt-Verbindung, muss für eine Nachricht ein Pfad im Netz gefunden werden.

- Leitungsvermittlung (circuit switching) gegenüber Paketvermittlung (packet switching)

- Routing = Wegewahl
- Switching = Art und Weise des Datentransfers innerhalb eines Vermittlungsknotens.

Leitungsvermittlung

- Zunächst werden Adressierungsdaten gesendet.
- Im Zuge der Adressdekodierung bauen die Vermittlungsknoten einen Weg vom Sender zum Empfänger auf.
- Wenn der Weg steht, dann folgen die Nutzdaten.
- Während die Nutzdaten übertragen werden, ist kein weiterer Vermittlungs- und Wegefindungs-aufwand nötig.
- Verbindungsabbau ist erforderlich.

Paketvermittlung

- Nutzdatenstrom wird in Pakete eingeteilt. Jedes Paket wird mit Adressierungsinformationen versehen (Paketkopf) und separat verschickt.
- Pakete können verloren gehen oder in veränderter Reihenfolge ankommen.

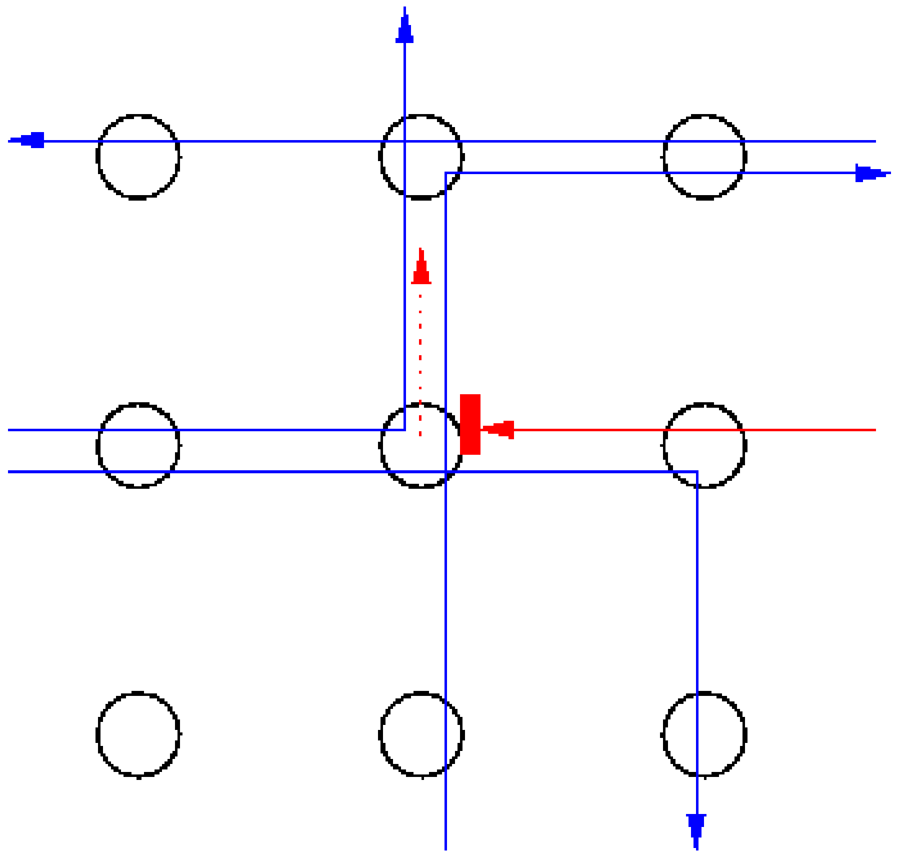
Paketvermittlung: Store-and-Forward

- Paket wird vollständig im Vermittlungsknoten aufgenommen (store), dann analysiert, dann über den ausgewählten Ausgang weitergeleitet (forward).
- Paket ist immer auf höchstens zwei Knoten und eine Verbindungsleitung verteilt.
- Paketlaufzeit ist proportional zur Paketlänge und zum Durchmesser des Netzes.
- Vermittlungsknoten benötigt ausreichend Pufferkapazität.
- Immer nur Einzelschritte im Netz „belegt", Blockierungsgefahr klein.

Paketvermittlung: Wormhole

- Sobald Paketkopf (bzw. genügend viel davon) angekommen ist, wird entschieden, über welchen Ausgang das (ganze) Paket weitergegeben wird.
- Ist Ausgang belegt, wird Paketrest nicht angenommen.
- Adresse (evtl. verkürzt/aktualisiert) verlässt den Knoten, noch ehe der Paketrest empfangen worden ist.
- Paket ist ggf. über viele Knoten und Leitungen verteilt.
- In jedem Vermittlungsknoten fallen nur die (kleinen) Zeiten für die Adressdekodierung an. Vermittlungszeit ist damit nur unbedeutend vom Netzdurchmesser abhängig.

Blockierung bei Wormhole bzw. Cut-Through



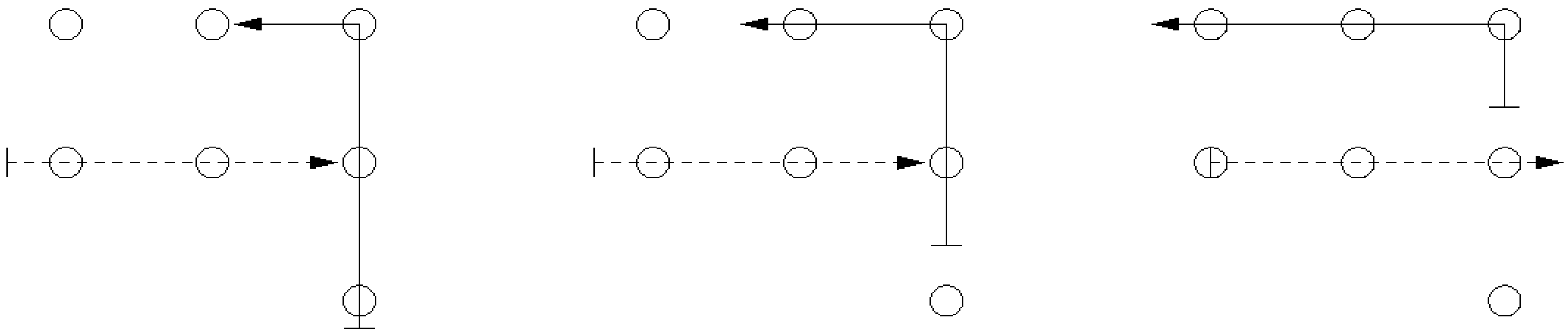
Blaue Pfeile: Pakete, die im Netz unterwegs sind.

Roter Pfeil: Paket kann nicht weitervermittelt werden, weil Ausgang belegt ist.

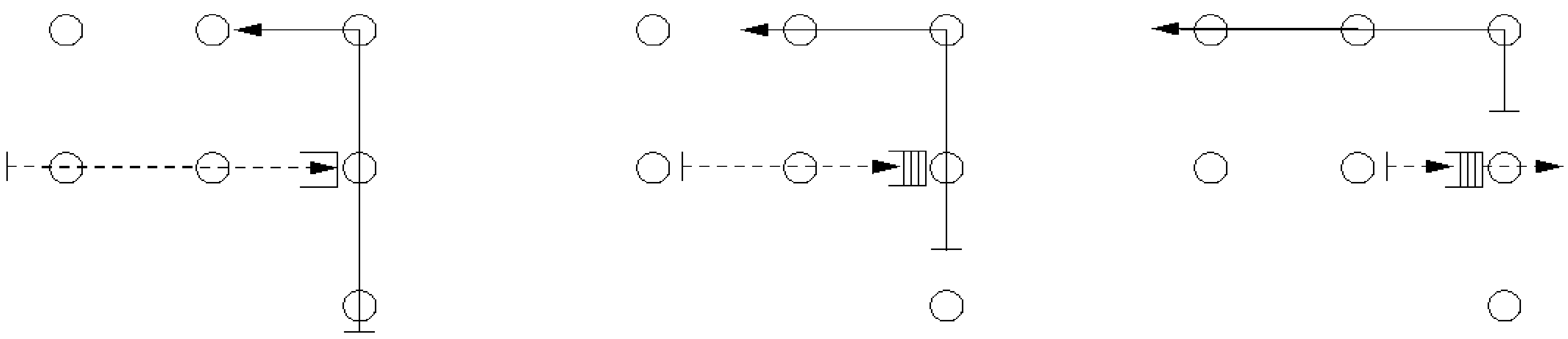
Paketvermittlung: Virtual Cut-Through

- Im Unterschied zum Wormhole-Routing, wird bei Virtual Cut-Through im Blockierungsfall der Paketrest empfangen und zwischengespeichert.
- Das führt in der Tendenz dazu, dass Blockierungen lokalen Charakter haben und sich wieder auflösen (statt Verklemmungen zu werden.)

Paketvermittlung: Wormhole und Virtual Cut-Through



a) Wormhole



b) Virtual-Cut-Through

Tabellenbasiertes Routing

- Statisches Routingverfahren: für alle Zieladressen ist die Route vorberechnet.
- Mit der dekodierten Adresse greift der Vermittlungsknoten in eine Tabelle und liest dort den für die Adresse zu verwendenden Ausgang.
- Die Größe der Routingtabelle ist proportional zur Knotenanzahl

Source Routing

- Der komplette Pfad wird vom Sender (Source) bestimmt.
- Jedes Paket führt die Nummern der zu verwendenden Ausgänge der Reihe nach mit.
- Die verwendete Ausgangsnummer wird an den nächsten Knoten nicht mehr mitgeschickt.