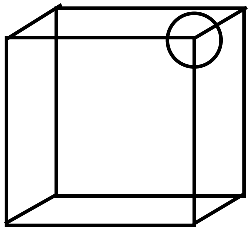
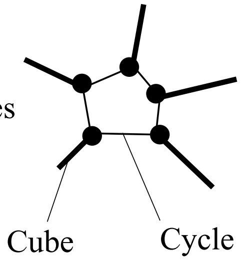
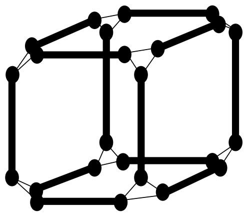
Cube Connected Cycles (CCC)

d-dimensionaler Hypercube mit aufgeschnittenen Ecken, die durch Gruppen von d ringförmig verbundenen Knoten ersetzt sind.



"Ecke" eines CCC (d=5)





Beachte: Jeder Knoten hat immer *drei* Anschlüsse!

Bei Dimension d: $n = d\,2^d$

Maximale / mittlerer Weglänge? ← Denkübung! (wieso < 2d ?)

Anzahl der Verbindungen = 3 n / 2
(statt O(n log n) wie beim Hypercube)

Es gibt viele weitere Verbindungstopologien...
(z.B. de Brujn Graphen)

Zufallstopologien (mit max. Grad = 4)

Verfahren A:

1) Starte mit einem Graphen ohne Kanten.
2) Wähle zwei zufällige Knoten, die noch weniger als 4 Kanten haben, und verbinde diese.
3) Wiederhole Schritt 2 solange wie möglich.
4) Falls ein unzusammenhängender Graph entsteht, beginne von vorne.

Verfahren B ("greedy graphs"):

Motivation: Kurze Zyklen vermeiden

2) Wähle zwei bel. Knoten *mit maximaler Entfernung* (einschliesslich ∞), die noch weniger als 4 Kanten haben, und verbinde diese.

Wie gut sind diese Zufallsgraphen bzgl. mittlerer Knotenentfernung und Routingbelastung?

Verzögerungszeiten, Routingoverhead

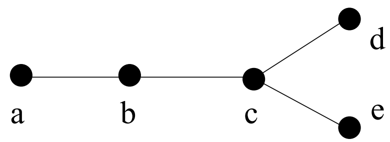
Bottleneck

Bem.: Explizites Routing nötig!

Routing-Belastung eines Knotens:
Zahl von Routen, die durch den Knoten gehen

Routing-Belastung einer Verbindung:
Zahl von Routen, die durch die Verbindung gehen

Beispiel:

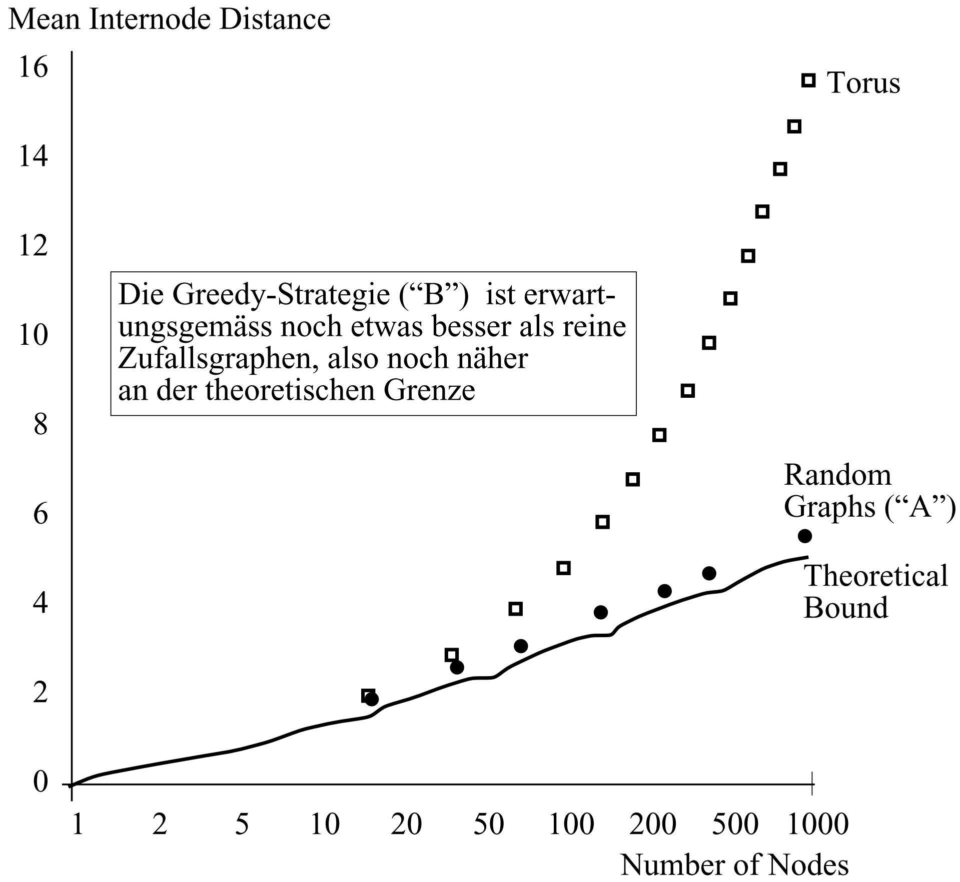


10 Routen von 20 gehen durch c

12 Routen von 20 gehen durch bc

Mittlerer Knotenabstand

- Für Knoten mit Grad 4
- Untersucht von D. Prior, Edinburgh



Man lese zu den Untersuchungen zu Zufallstopologien folgenden Artikel:

D. M. N. Prior, M. G. Norman, N. J. Radcliffe, L. J. Clarke: *What Price Regularity*?, Concurrency, Practice and Experience, Vol 2 No 1, pp. 55-78, 1990

Link Loads for 100 Processors

- Wenn jeder mit jedem anderen kommuniziert: ca. 100^2 Nachrichten (auf kürzesten Pfaden)

- 10×10-Torus: mittlere Entfernung = 5
  → Jede Nachricht benutzt im Mittel 5 links.
  Bei 100^2 Nachrichten, 400 links → 125 Nachrichten / link