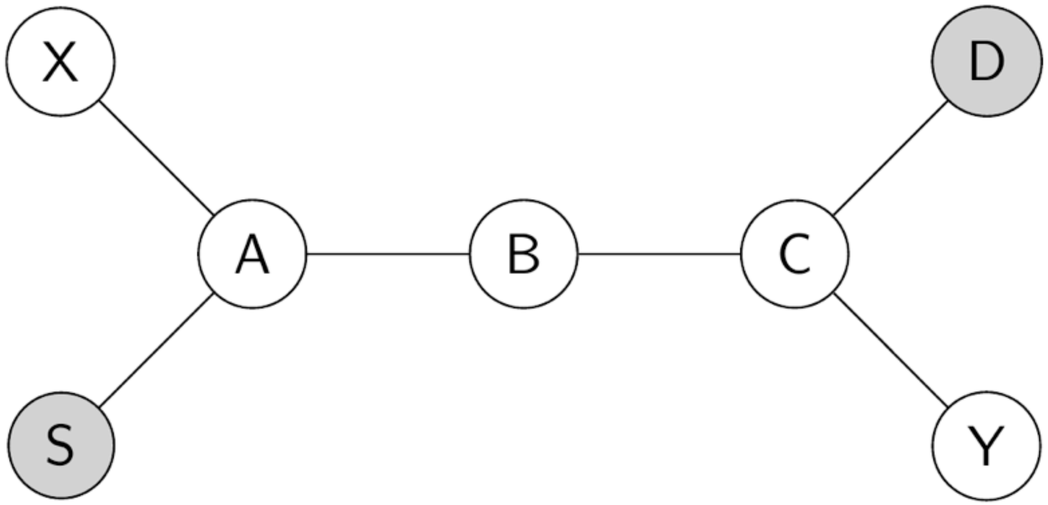
Connection Setup in a Quantum Network

Rodney Van Meter, Takaaki Matsuo
draft-van-meter-qirg-quantum-connection-setup-00
QIRG @IRTF/IETF104
Prague
2019/3/26

Quantum Connection

Distribution of end-to-end Bell pairs:

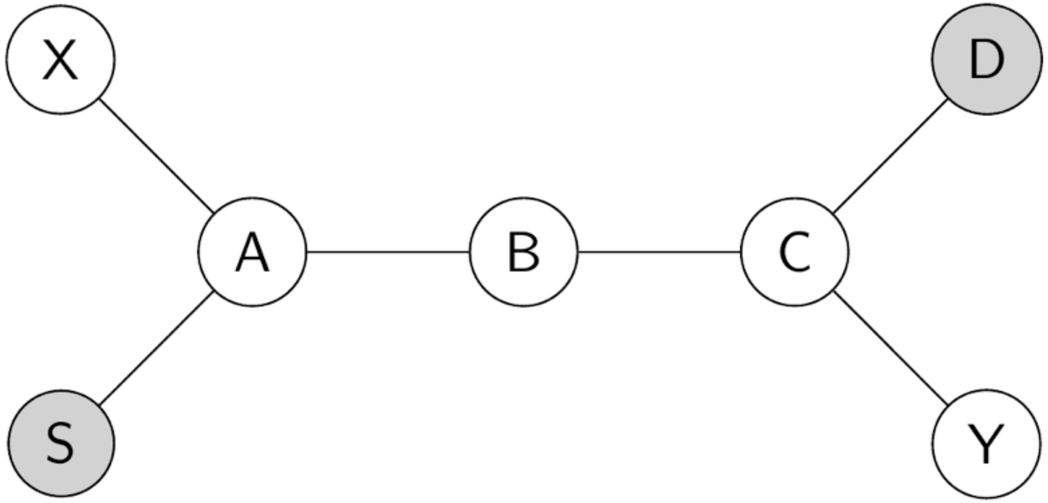
- On request from source node S
- Middle nodes perform entanglement swapping and error management



Quantum Connection

Distribution of end-to-end Bell pairs:

- On request from source node S
- Middle nodes perform entanglement swapping and error management

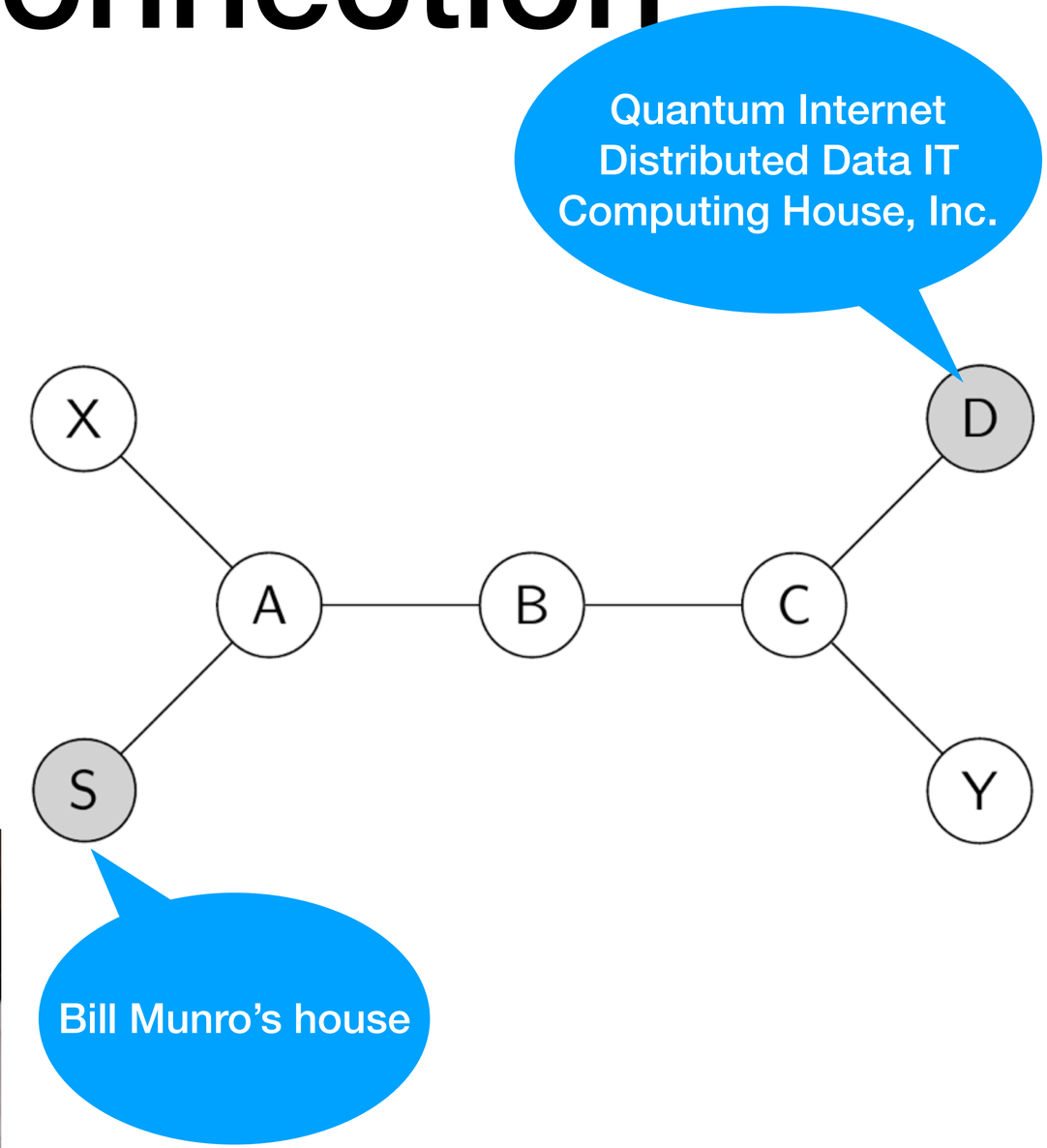


Bill Munro’s house

Quantum Connection

Distribution of end-to-end Bell pairs:

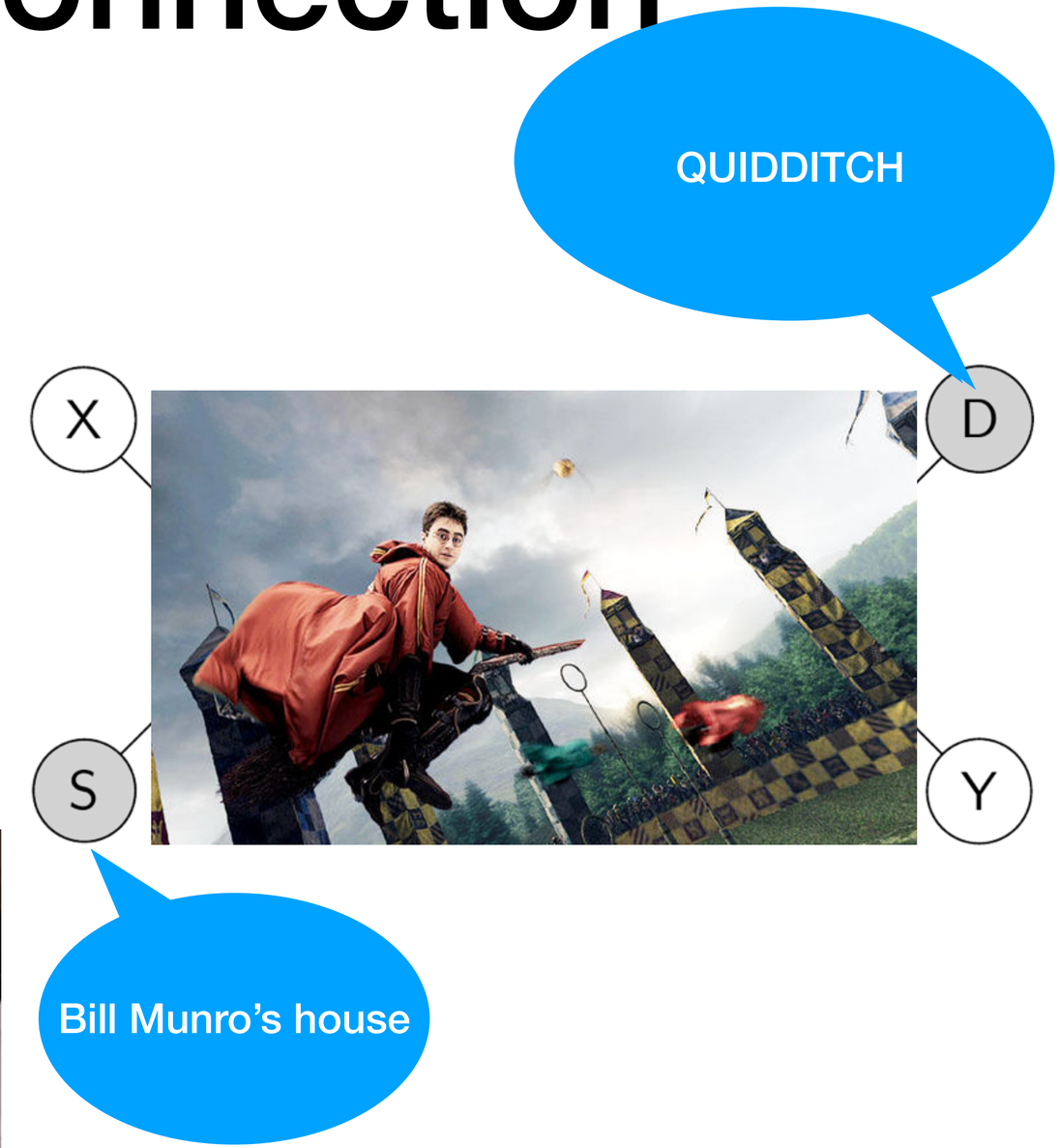
- On request from source node S
- Middle nodes perform entanglement swapping and error management



Quantum Connection

Distribution of end-to-end Bell pairs:

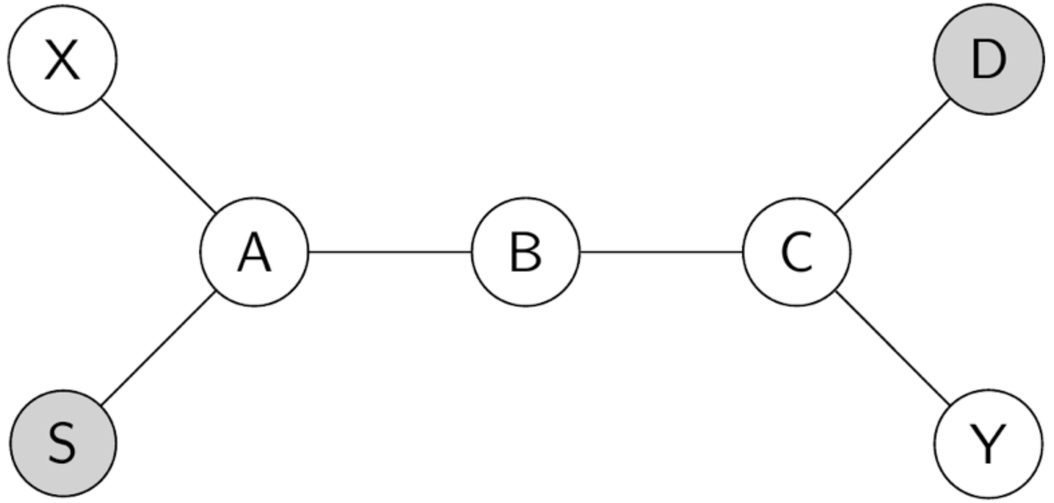
- On request from source node S
- Middle nodes perform entanglement swapping and error management



Quantum Connection

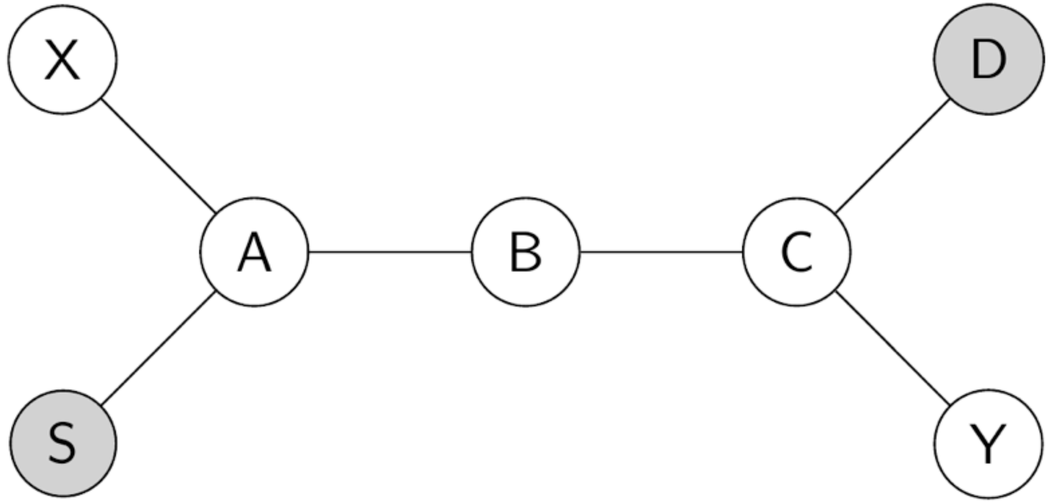
Distribution of end-to-end Bell pairs:

- On request from source node S
- Middle nodes perform entanglement swapping and error management



Stages of the Problem

- Need to select a path (routing) (rdv et al., *Networking Science* 2013)
- **Plan sequences of operations**
- **Convey sequences to nodes**



Constraints/assumptions

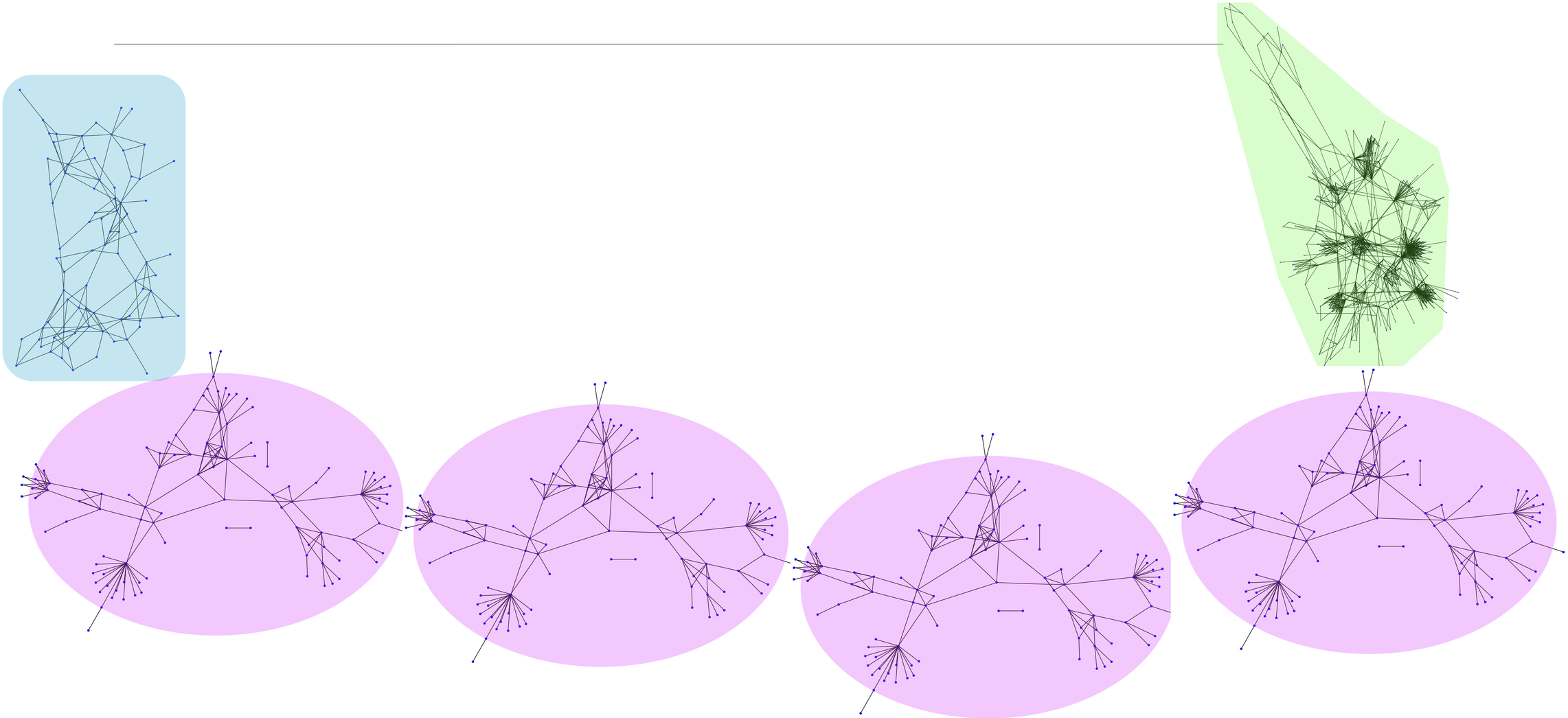
- Links are heterogeneous and not *a priori* known
- Resource management (multiplexing scheme) beyond today's scope, but critically important

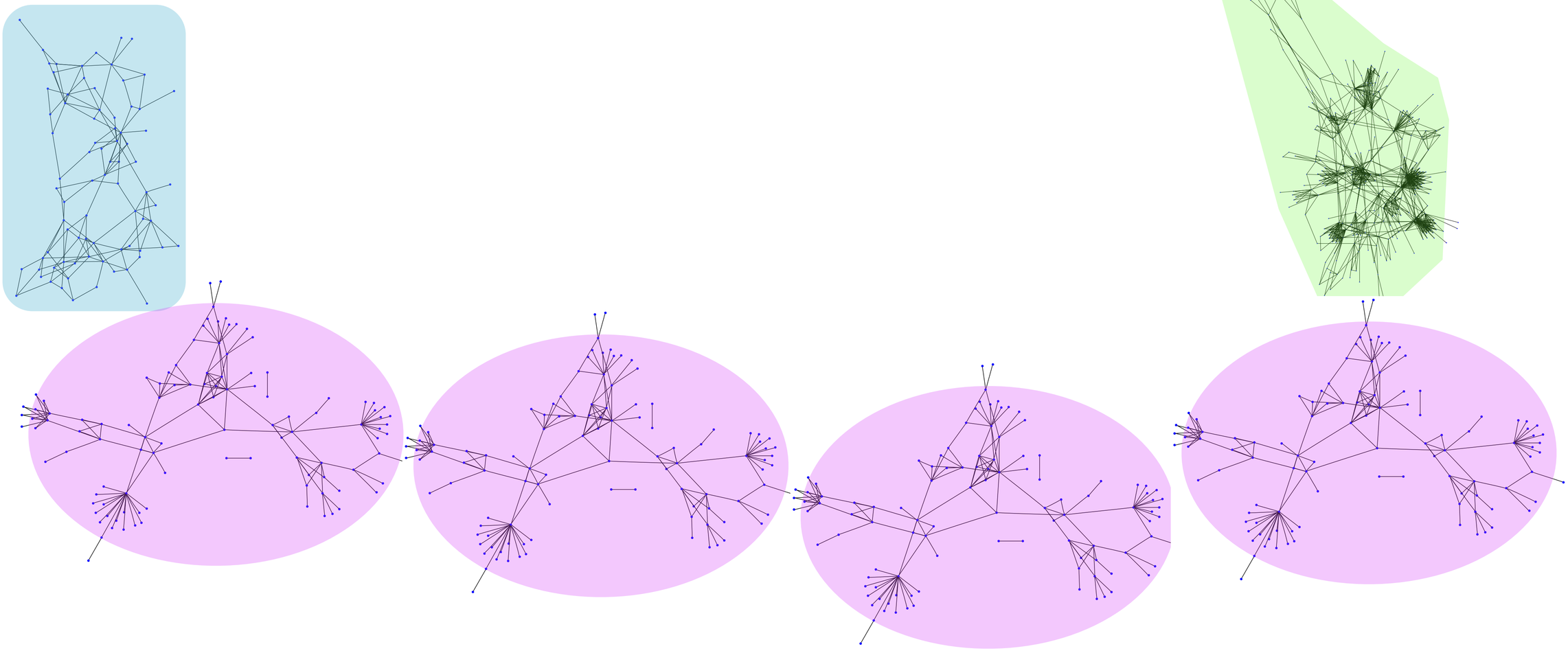
Information Each Node Holds

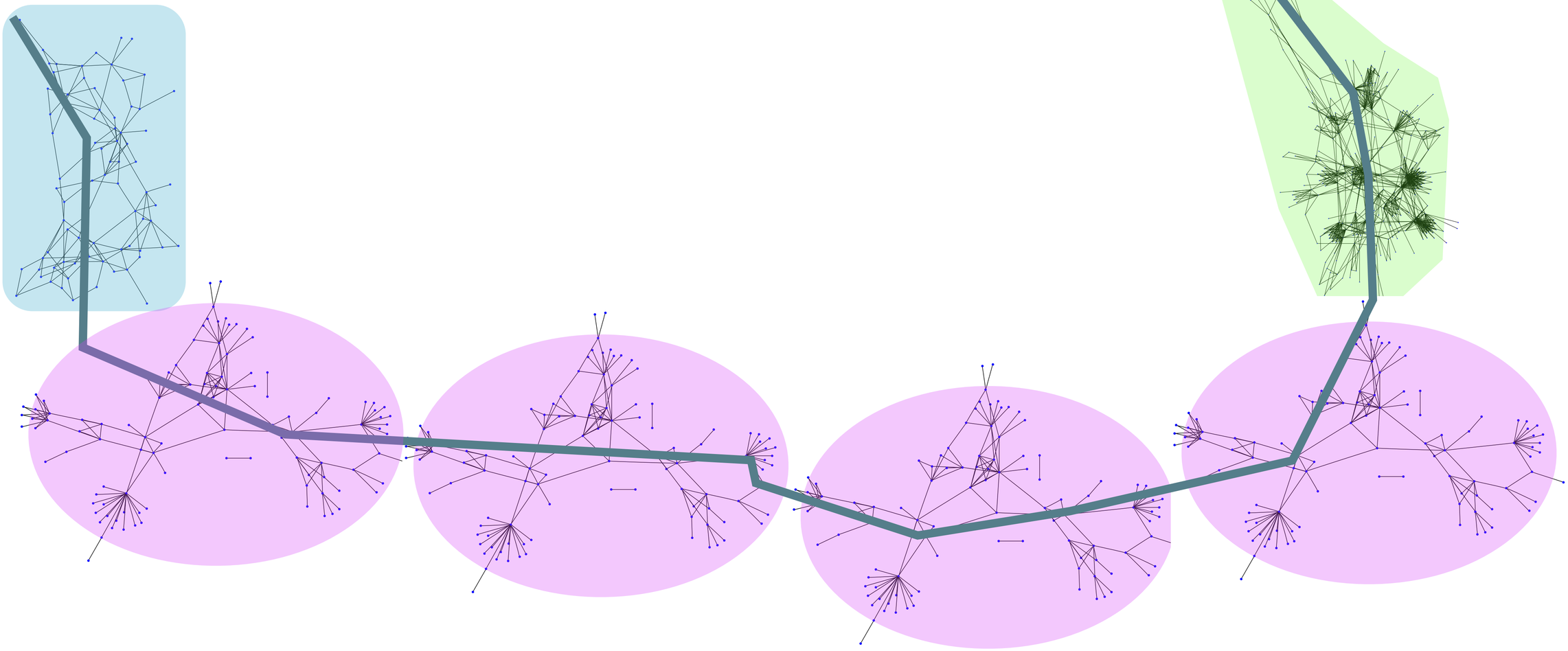
- Its own capabilities
  - amount of memory, memory lifetime
  - gate fidelities
- Link information
  - who neighbors are
  - link entanglement trial rate, success probability, fidelity (or full density matrix)
- Topology of the *local* network, with a routing metric
- Where the gateway to the outside world is

Information Each Node Does *Not* Have

- Full density matrix of the base Bell pairs generated by every node
- Number of qubits in every “QNIC” in the whole network
- Local gate fidelities for other nodes
- Anything at all about the internals of neighboring networks

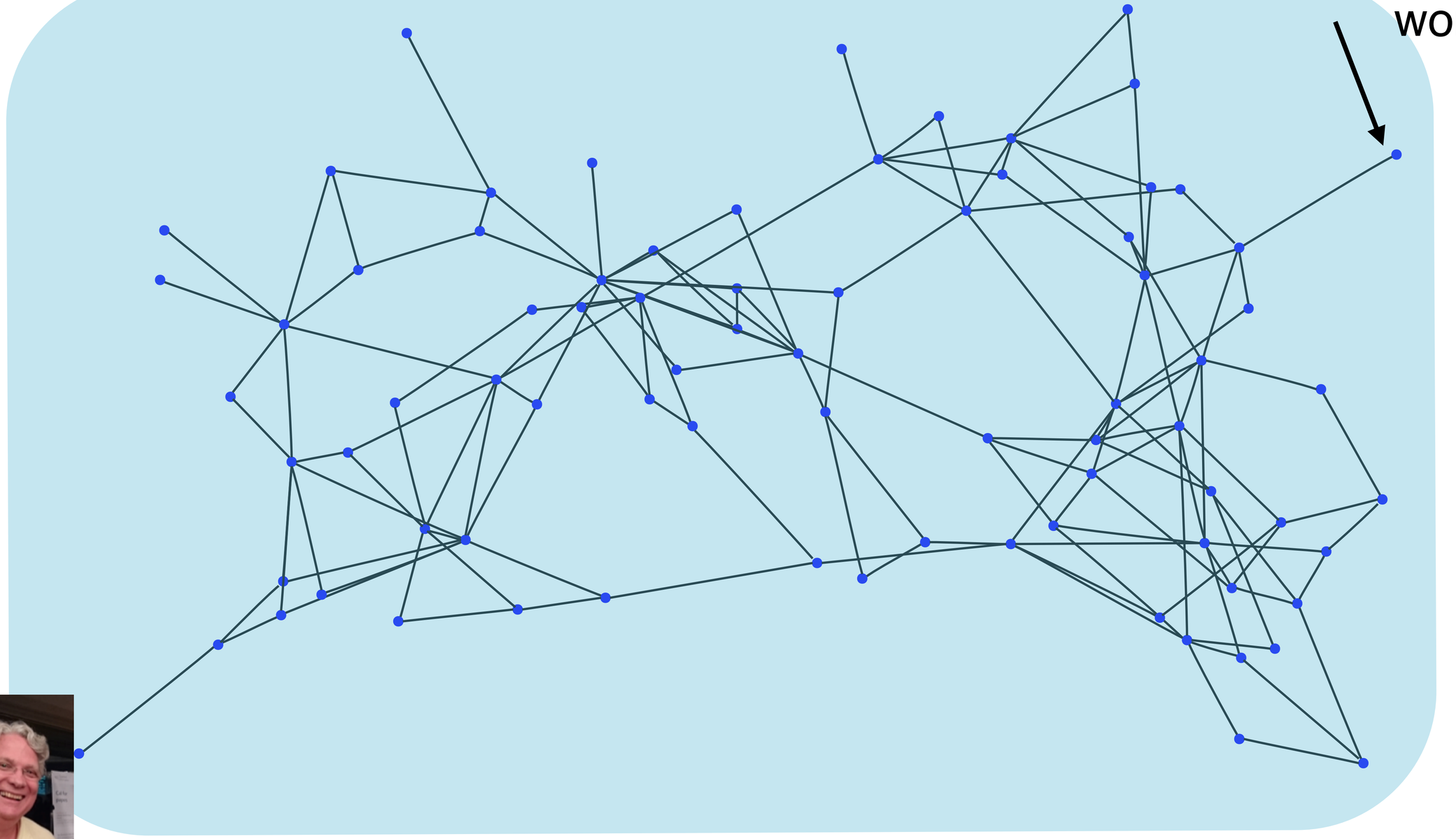






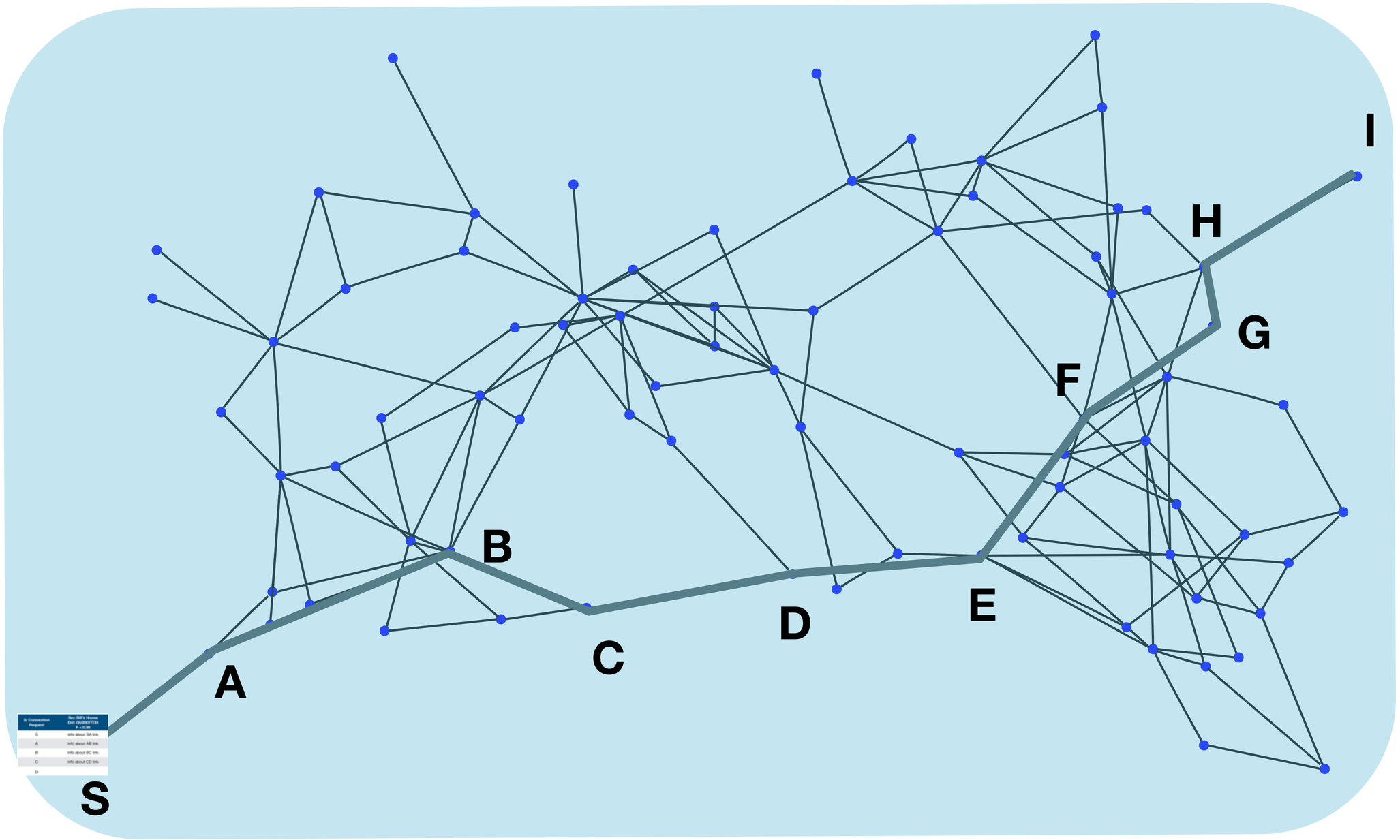
All opaque to Bill
(or at the *very* least,
not known until he tries to
connect to QUIDDITCH)

Gateway to the outer world

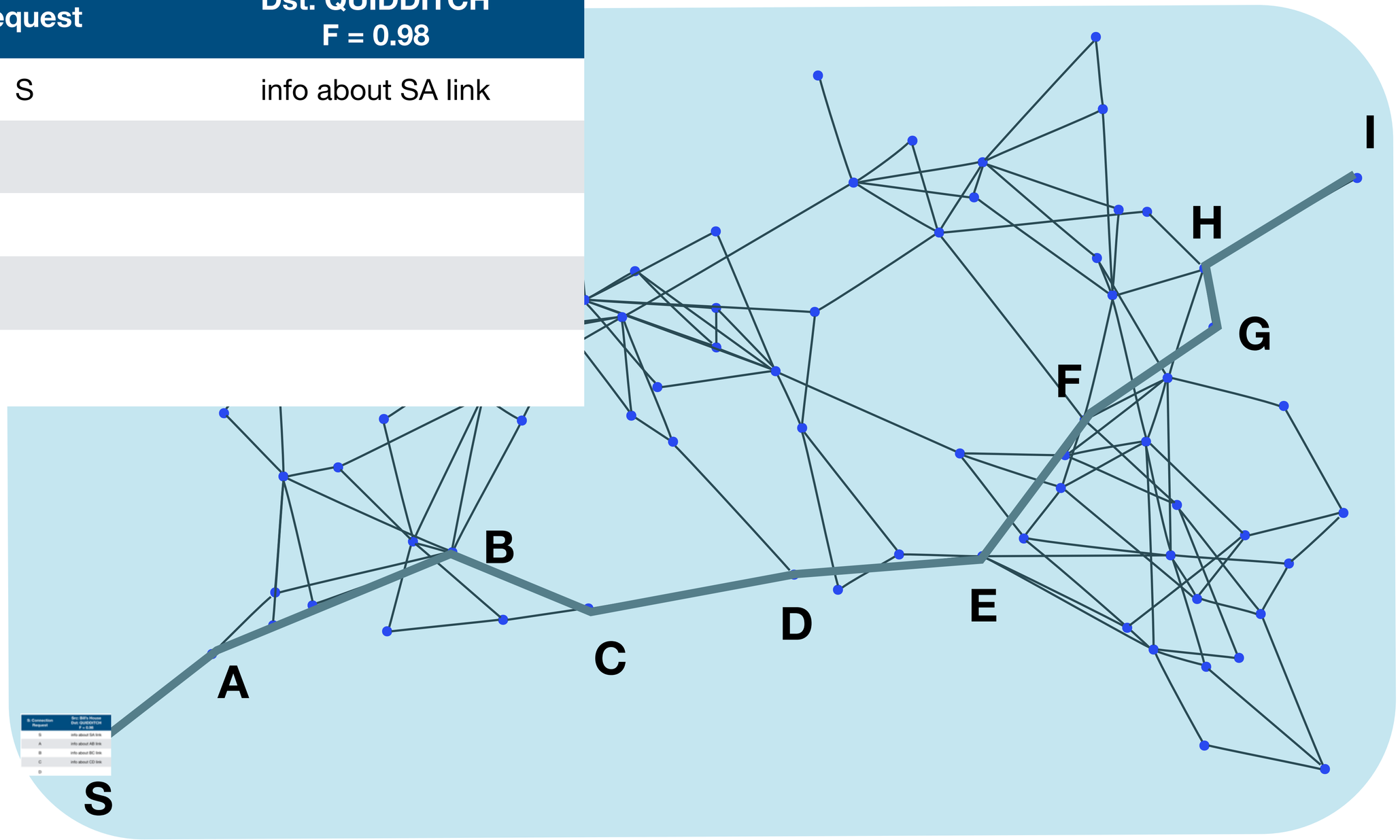


Gateway to the outer world

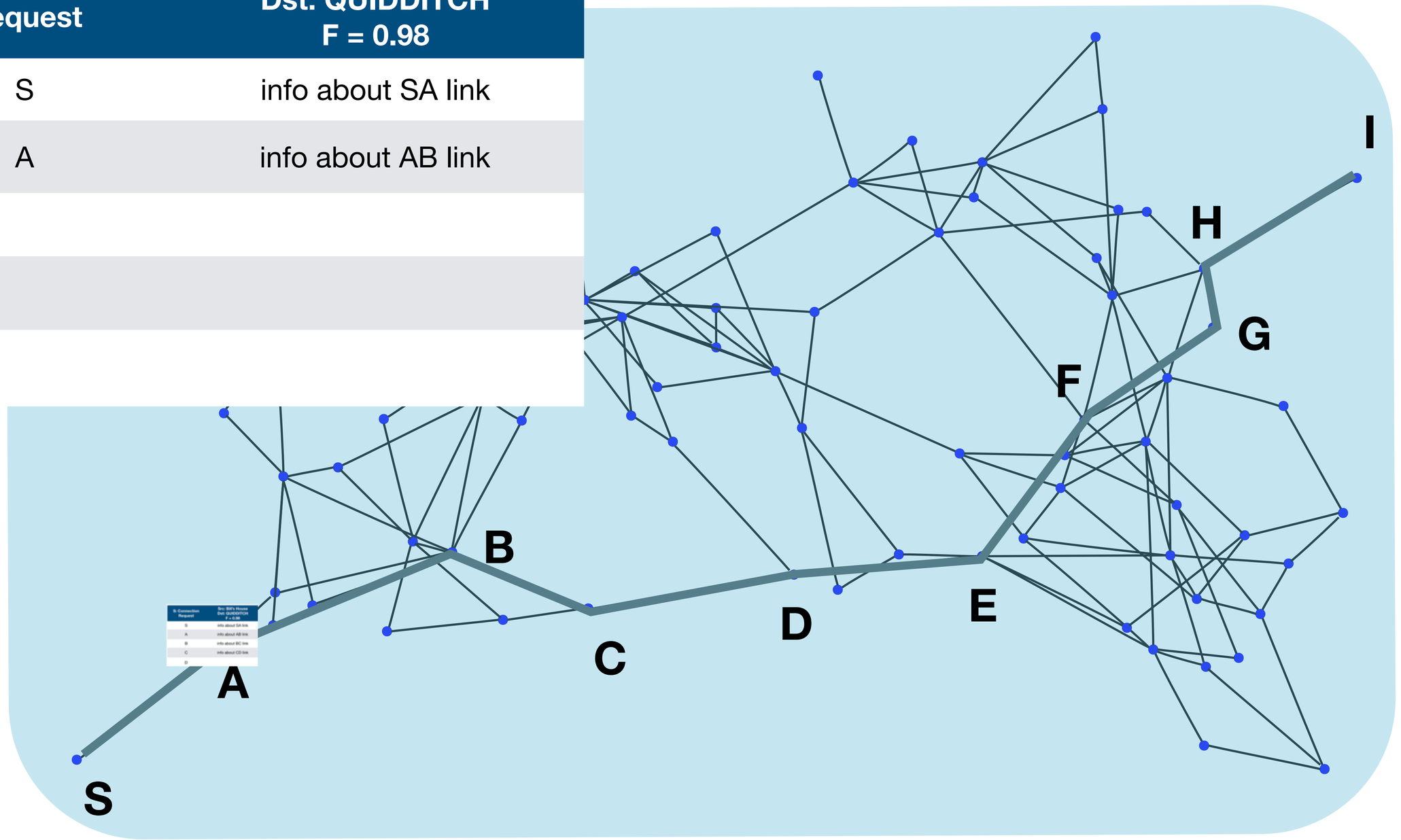
Path selected using minimal information


I
H
G
F
B
C
D
E
A
S
S: Connection Request
Src: Bill's House
Dst: QUIDDITCH
Y = 0.98
S
info about SA link
A
info about AB link
B
info about BC link
C
info about CD link
D

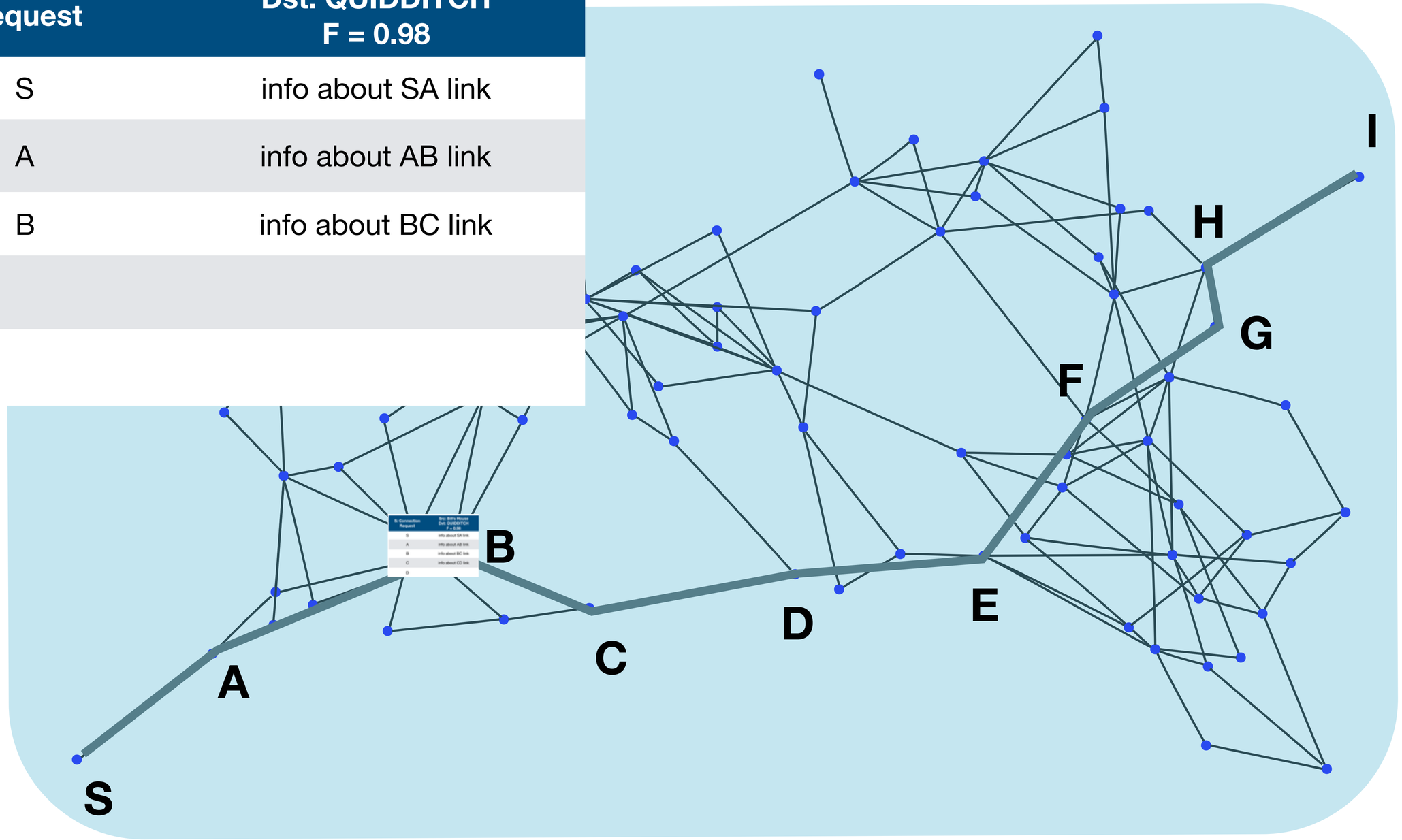
S: Connection Request	Src: Bill's House Dst: QUIDDITCH F = 0.98
S	info about SA link



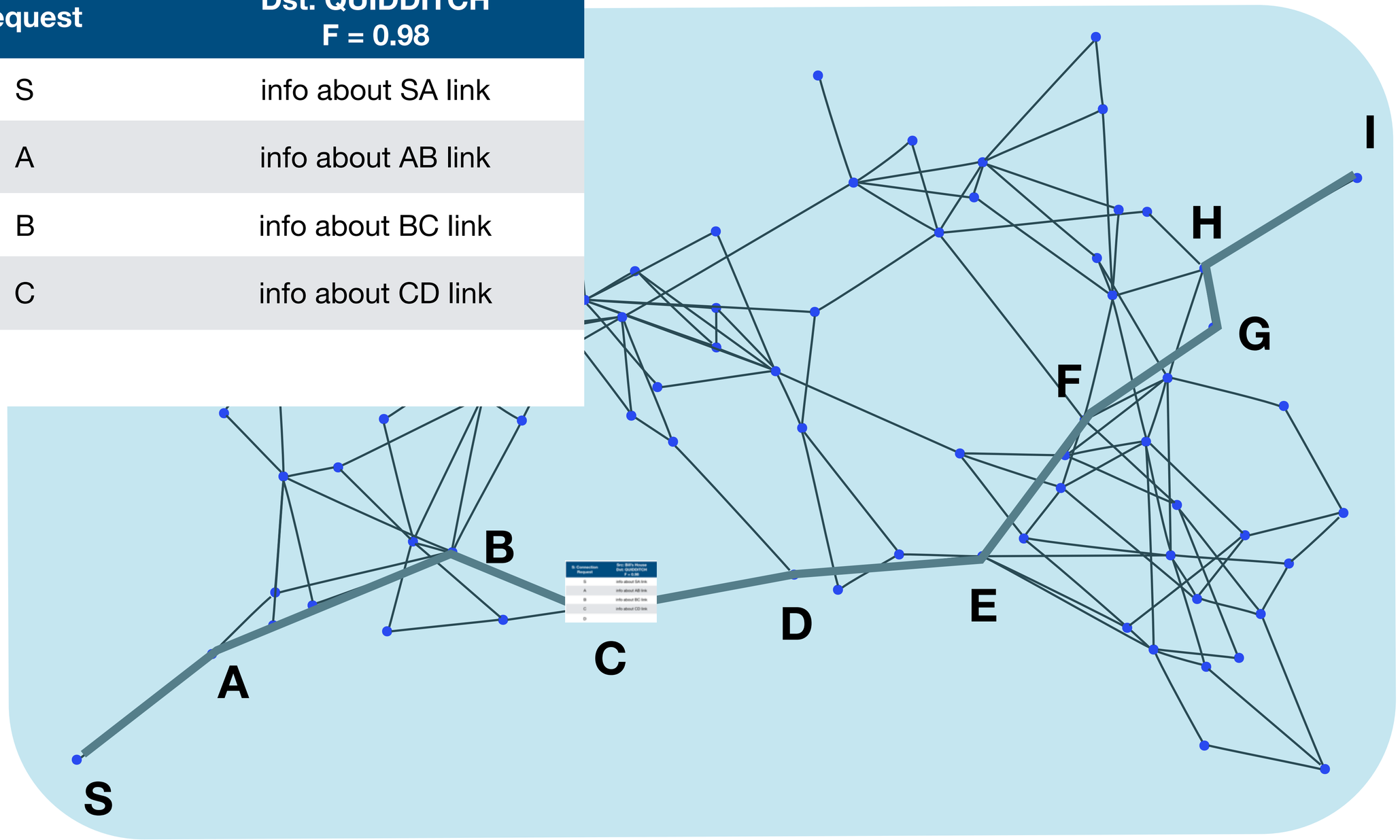
S: Connection Request	Src: Bill's House Dst: QUIDDITCH F = 0.98
S	info about SA link
A	info about AB link



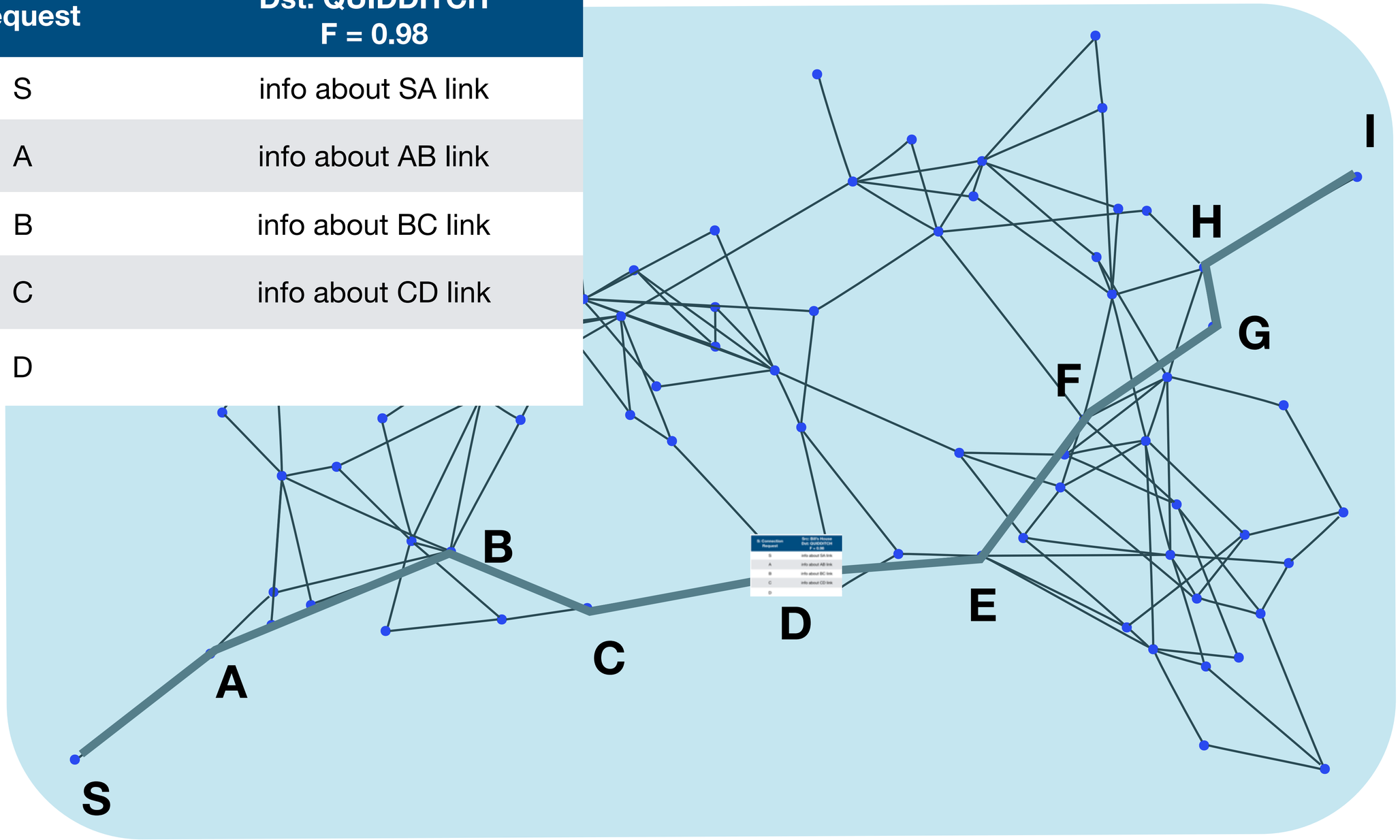
S: Connection Request	Src: Bill's House Dst: QUIDDITCH F = 0.98
S	info about SA link
A	info about AB link
B	info about BC link



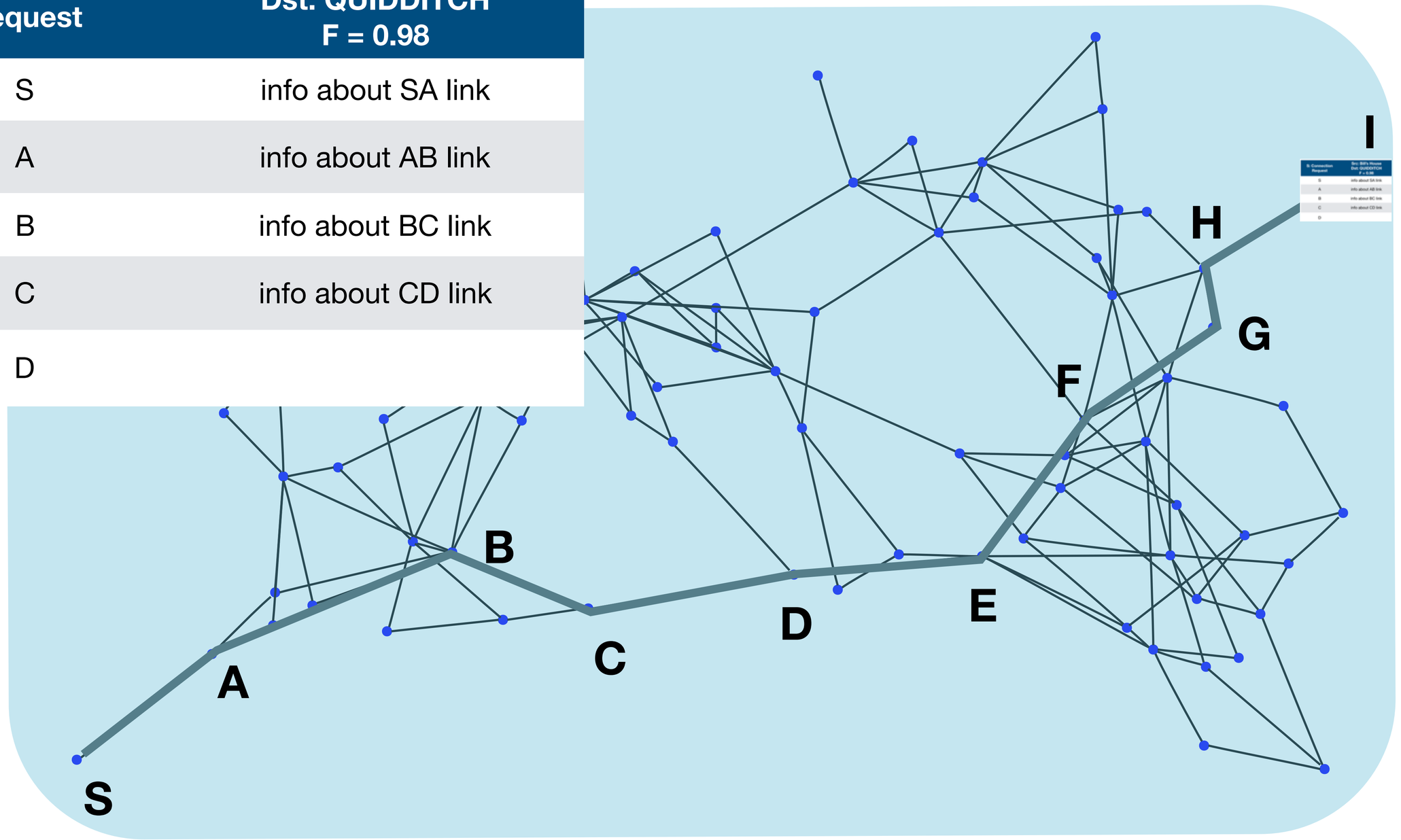
S: Connection Request	Src: Bill's House Dst: QUIDDITCH F = 0.98
S	info about SA link
A	info about AB link
B	info about BC link
C	info about CD link



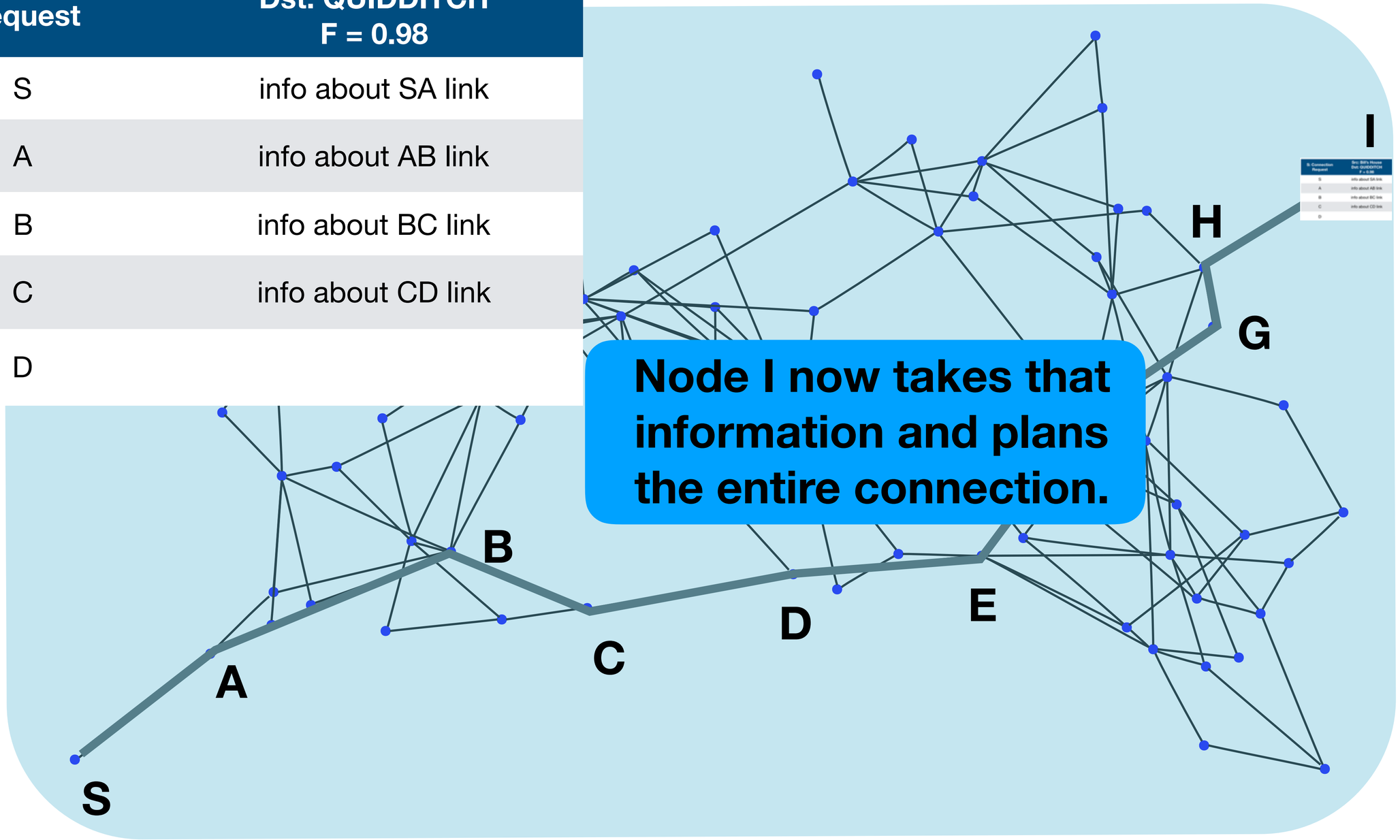
S: Connection Request	Src: Bill's House Dst: QUIDDITCH F = 0.98
S	info about SA link
A	info about AB link
B	info about BC link
C	info about CD link
D	



S: Connection Request	Src: Bill's House Dst: QUIDDITCH F = 0.98
S	info about SA link
A	info about AB link
B	info about BC link
C	info about CD link
D	



S: Connection Request	Src: Bill's House Dst: QUIDDITCH F = 0.98
S	info about SA link
A	info about AB link
B	info about BC link
C	info about CD link
D	



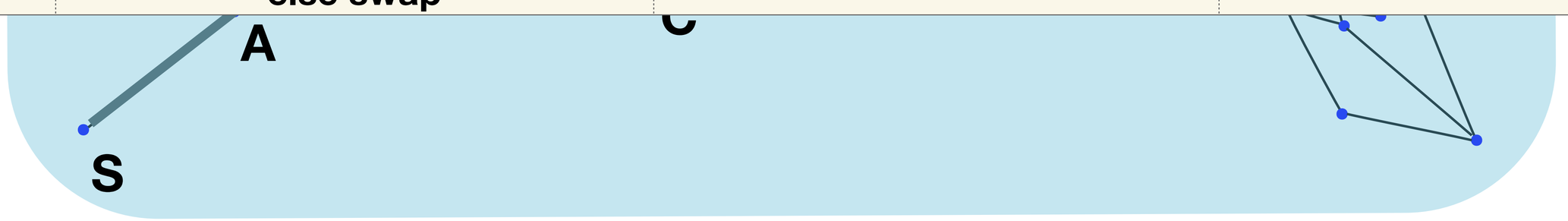
Node I now takes that information and plans the entire connection.

D: Path Setup	Src: Bill's House Dst: QUIDDITCH F = 0.98		
S	conditions to purify conditions to swap		
A	conditions to purify conditions to swap		
B	purify if FG < 0.98, EF < 0.98 else swap		
C	purify if BC < 0.98, CD < 0.98	purify if AC < 0.98, CE < 0.98 else swap	
D	purify if CD < 0.98, DE < 0.98 else swap		
E	purify if DE < 0.98, EF < 0.98	purify if CE < 0.98, EG < 0.98	purify if AE < 0.98, EI < 0.98 else swap
F	purify if EF < 0.98, FG < 0.98 else swap		
G	purify if FG < 0.98, GH < 0.98	purify if EG < 0.98, GI < 0.98 else swap	
H	purify if GH < 0.98, HI < 0.98 else swap		

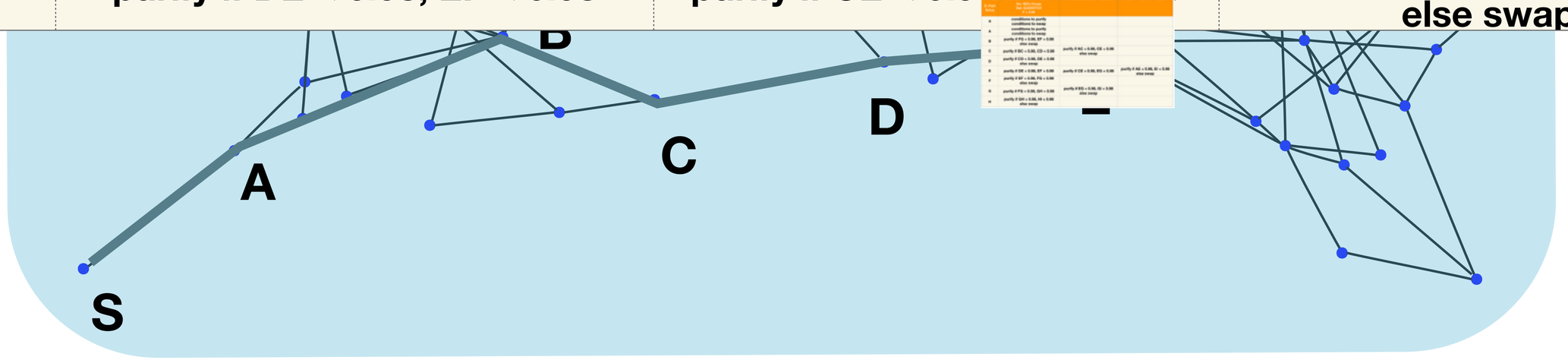
D: Path Setup	Src: Bill's House Dst: QUIDDITCH F = 0.98		
S	conditions to purify conditions to swap		
A	conditions to purify conditions to swap		
B	purify if FG < 0.98, EF < 0.98 else swap		
C	purify if BC < 0.98, CD < 0.98	purify if AC < 0.98, CE < 0.98 else swap	
D	purify if CD < 0.98, DE < 0.98 else swap		
E	purify if DE < 0.98, EF < 0.98	purify if CE < 0.98, EG < 0.98	purify if AE < 0.98, EI < 0.98 else swap
F	purify if EF < 0.98, FG < 0.98 else swap		
G	purify if FG < 0.98, GH < 0.98	purify if EG < 0.98, GI < 0.98 else swap	

S

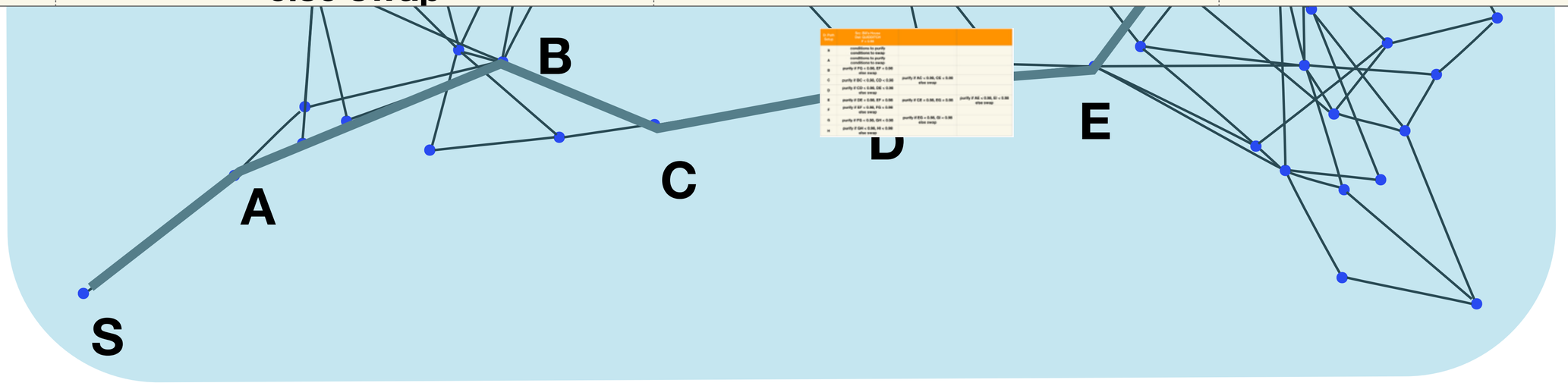
D: Path Setup	Src: Bill's House Dst: QUIDDITCH F = 0.98		
S	conditions to purify conditions to swap		
A	conditions to purify conditions to swap		
B	purify if FG < 0.98, EF < 0.98 else swap		
C	purify if BC < 0.98, CD < 0.98	purify if AC < 0.98, CE < 0.98 else swap	
D	purify if CD < 0.98, DE < 0.98 else swap		
E	purify if DE < 0.98, EF < 0.98	purify if CE < 0.98, EG < 0.98	purify if AE < 0.98, EI < 0.98 else swap
F	purify if EF < 0.98, FG < 0.98 else swap		



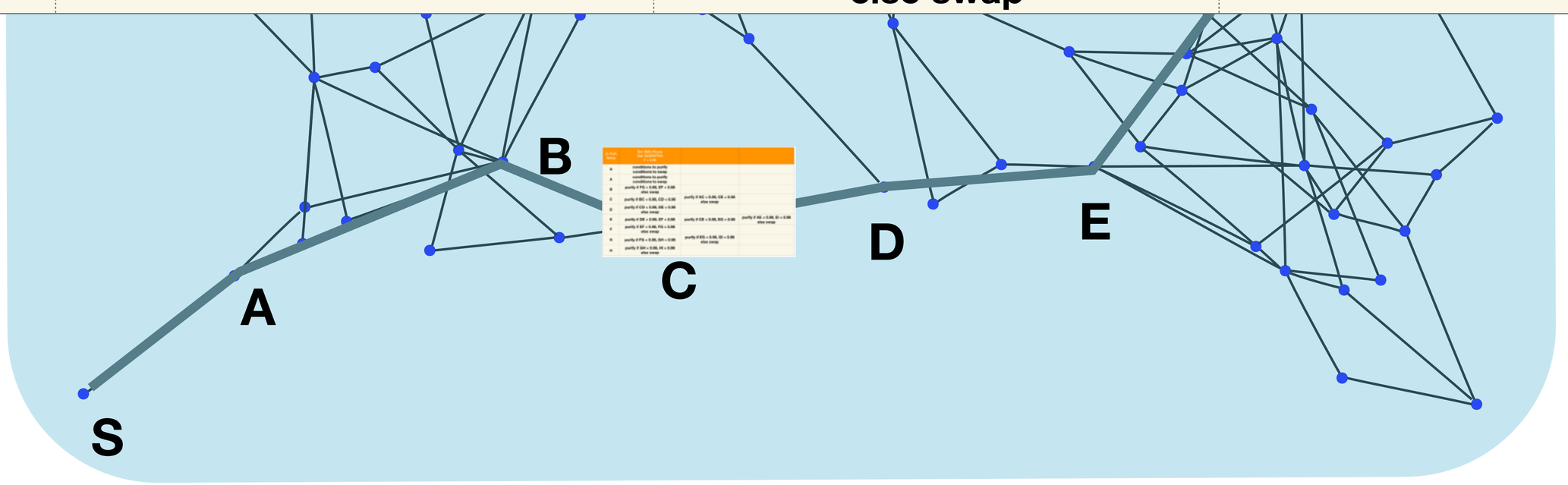
D: Path Setup	Src: Bill's House Dst: QUIDDITCH F = 0.98		
S	conditions to purify conditions to swap		
A	conditions to purify conditions to swap		
B	purify if FG < 0.98, EF < 0.98 else swap		
C	purify if BC < 0.98, CD < 0.98	purify if AC < 0.98, CE < 0.98 else swap	
D	purify if CD < 0.98, DE < 0.98 else swap		
E	purify if DE < 0.98, EF < 0.98	purify if CE < 0.98, EG < 0.98	purify if AE < 0.98, EI < 0.98 else swap



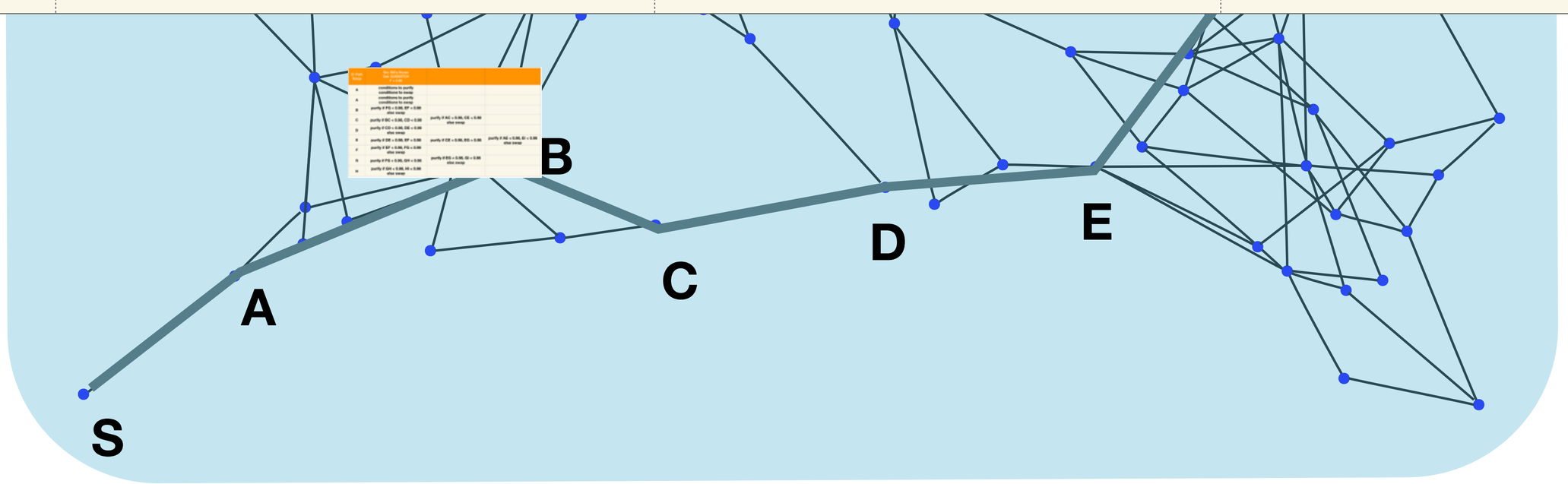
D: Path Setup	Src: Bill's House Dst: QUIDDITCH F = 0.98		
S	conditions to purify conditions to swap		
A	conditions to purify conditions to swap		
B	purify if FG < 0.98, EF < 0.98 else swap		
C	purify if BC < 0.98, CD < 0.98	purify if AC < 0.98, CE < 0.98 else swap	
D	purify if CD < 0.98, DE < 0.98 else swap		



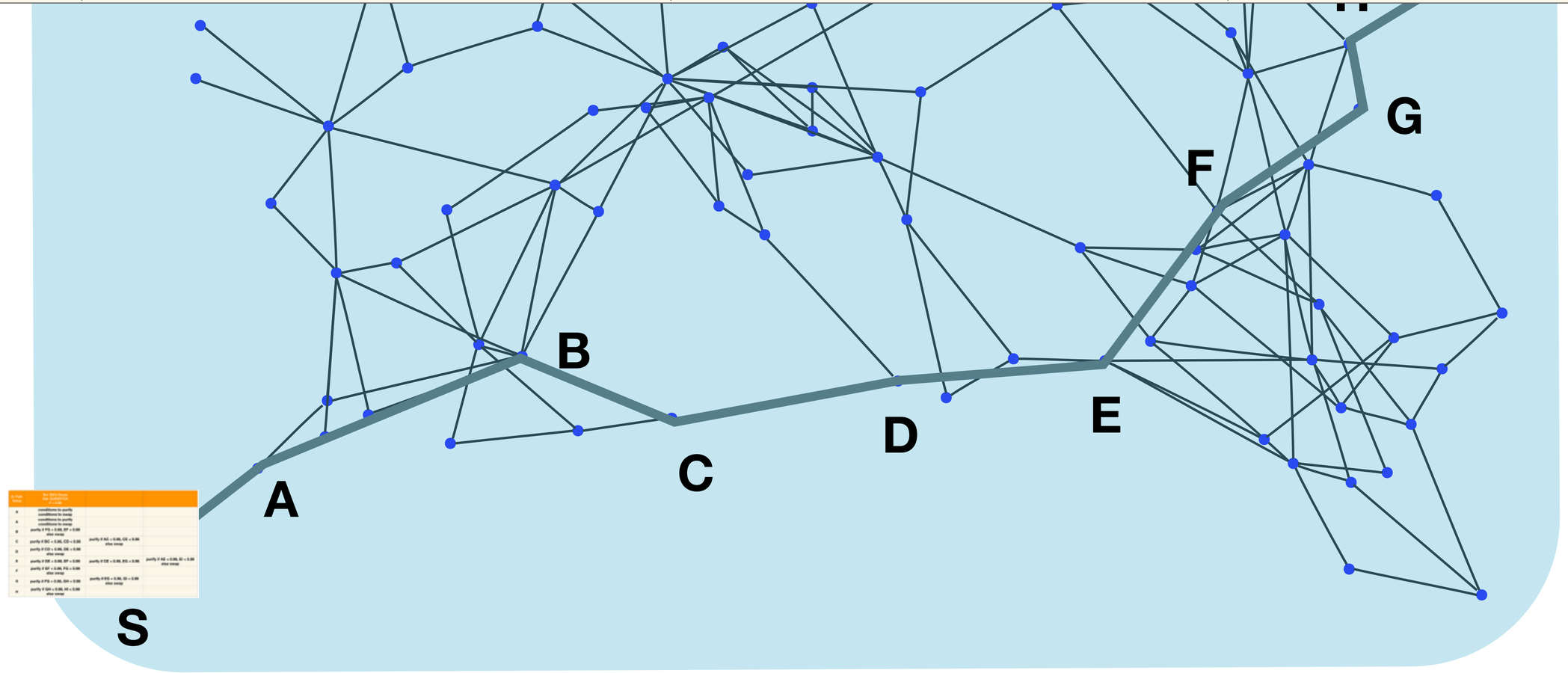
D: Path Setup	Src: Bill's House Dst: QUIDDITCH F = 0.98		
S	conditions to purify conditions to swap		
A	conditions to purify conditions to swap		
B	purify if FG < 0.98, EF < 0.98 else swap		
C	purify if BC < 0.98, CD < 0.98	purify if AC < 0.98, CE < 0.98 else swap	



D: Path Setup	Src: Bill's House Dst: QUIDDITCH F = 0.98		
S	conditions to purify conditions to swap		
A	conditions to purify conditions to swap		
B	purify if FG < 0.98, EF < 0.98 else swap		



D: Path Setup	Src: Bill's House Dst: QUIDDITCH F = 0.98		
S	conditions to purify conditions to swap		

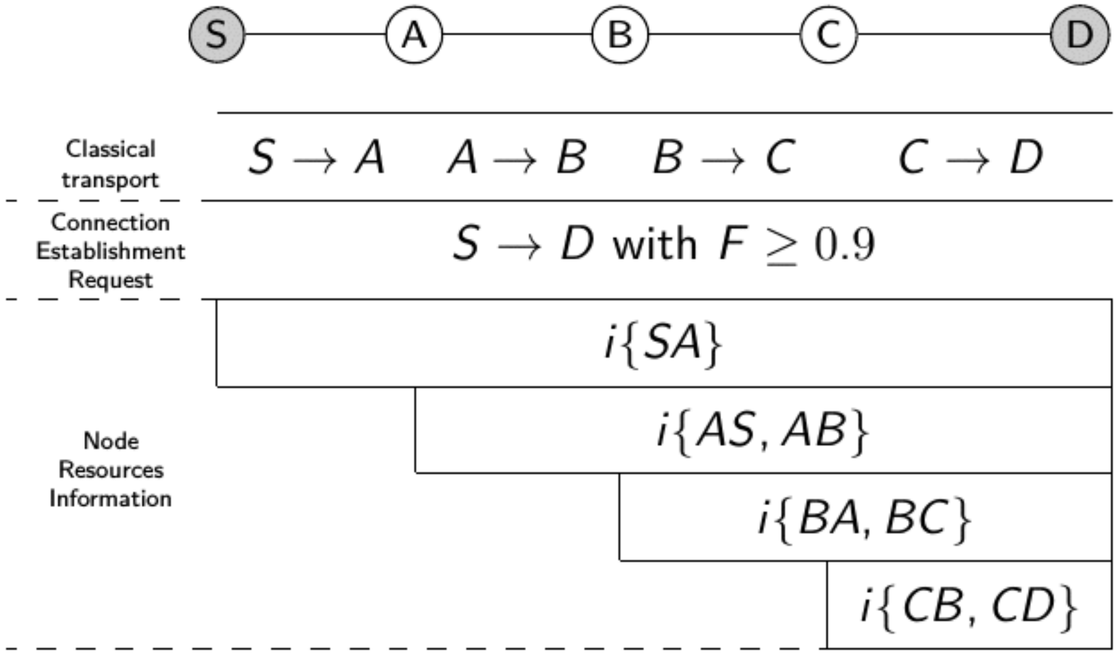


Connection Establishment Request

Used by D to establish rules and Bell pairs distribution.

Nodes provide information about the path:

- Resources
- Quality of the link, etc.

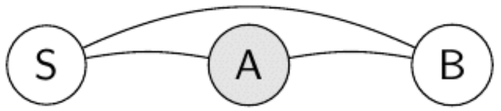


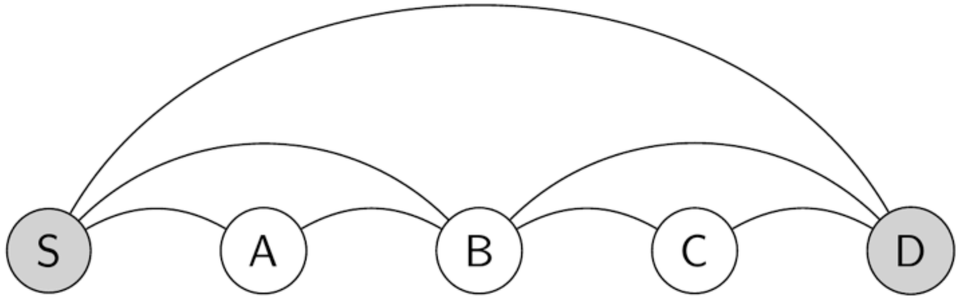
Connection Establishment (Response)

Destination node computes a swapping scheme.

Information provided by the middle nodes is important to create a consistent set of rules.

	Conditions	Actions
Pairs management	$A \sim x$ $F \leq 0.5$	Discard
	$A \sim x$ $A \sim x$ both $F \leq 0.95$	Purify
Swapping	$A \sim B$ $A \sim S$ both $F > 0.95$	Swap

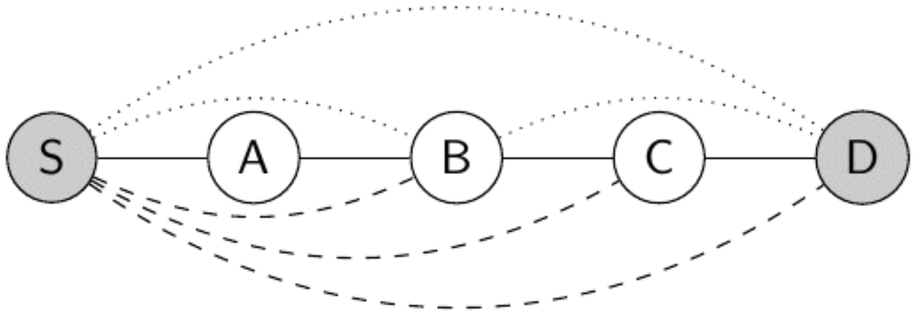




Every node receives a set of rules that will be used to maintain a consistent distributed swapping protocol.

Challenges

- Decomposition choice:
  swapping order hierarchical (top)
  or hop-by-hop (bottom)
- Limiting classical messages
- Consistency of the behaviors of every node
- Class of service



Comments from the ML

- Q: What about Segment Routing?
- A: Good thought. I'm not familiar w/ the current SR, but waypoint routing + circuit/reservation setup is what I have in mind.

  Moreover, intended to be recursive, treating each network as a node in a larger graph (more than just two-layer IGP/EGP).