

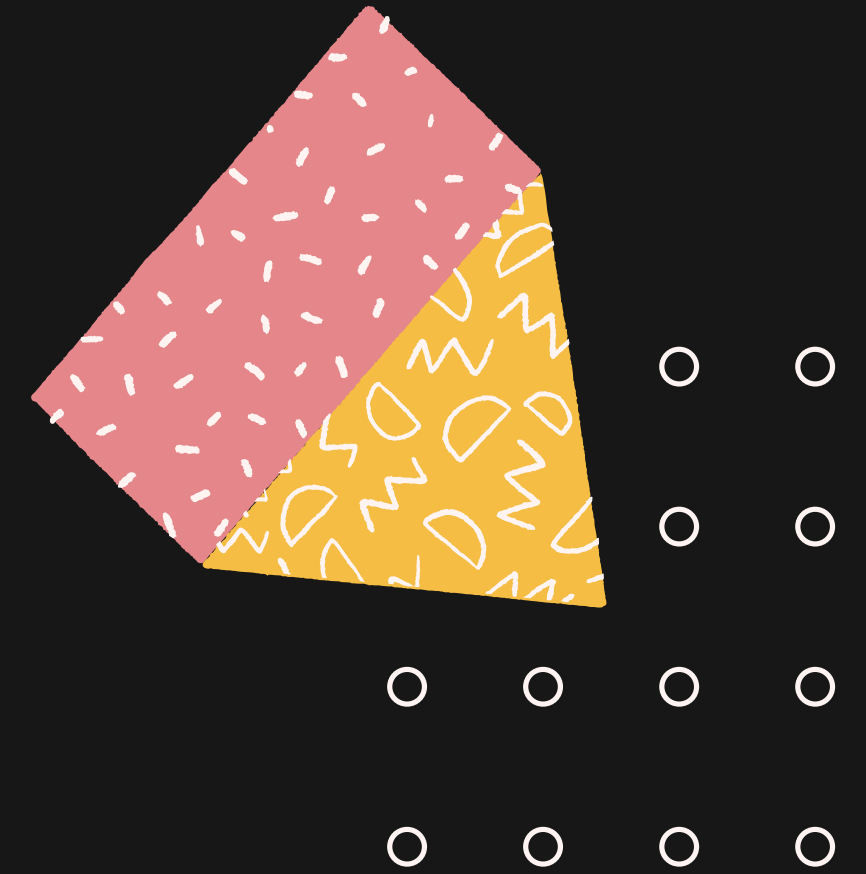


CONEXÃO P2P

E TAPESTRY

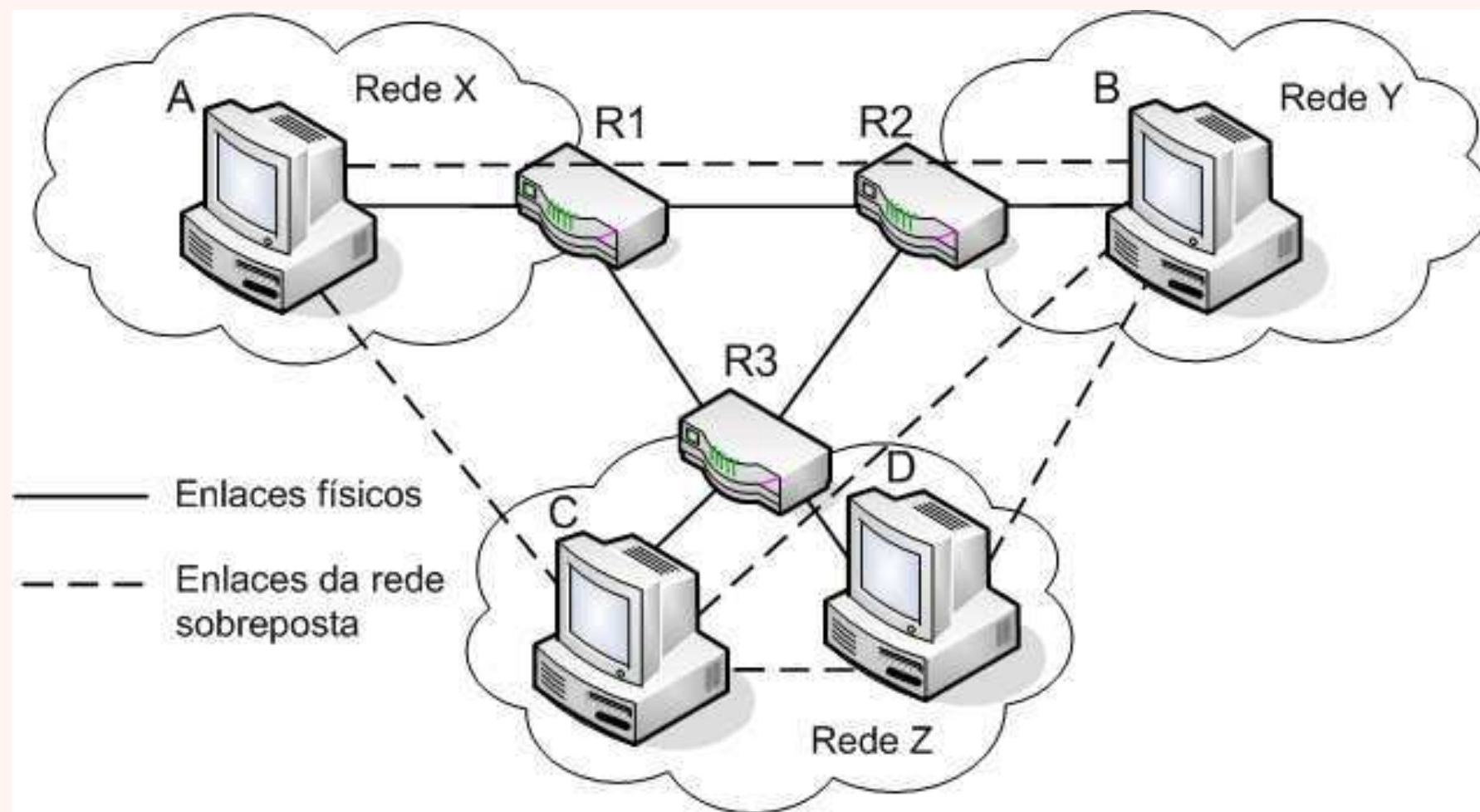
João Pedro Moura Oliveira - 190030879

ENTENDENDO A CONEXÃO P2P



- Informação descentralizada
- Compartilhamento de arquivos
- Cada nó é cliente e servidor
- Privilégios iguais para os nós
- Redes sobrepostas

REDES SOBREPOSTAS



Fonte: https://www.researchgate.net/figure/Figura-32-Um-exemplo-de-uma-rede-sobreposta-overlay_fig2_320595821

- Divisão em 2 modelos:
 - Redes Não Estruturadas
 - Redes Estruturadas
- *Distributed Hash Tables* (DHT)
- Autonomia e Descentralização
- Tolerância a falhas
- Escalabilidade

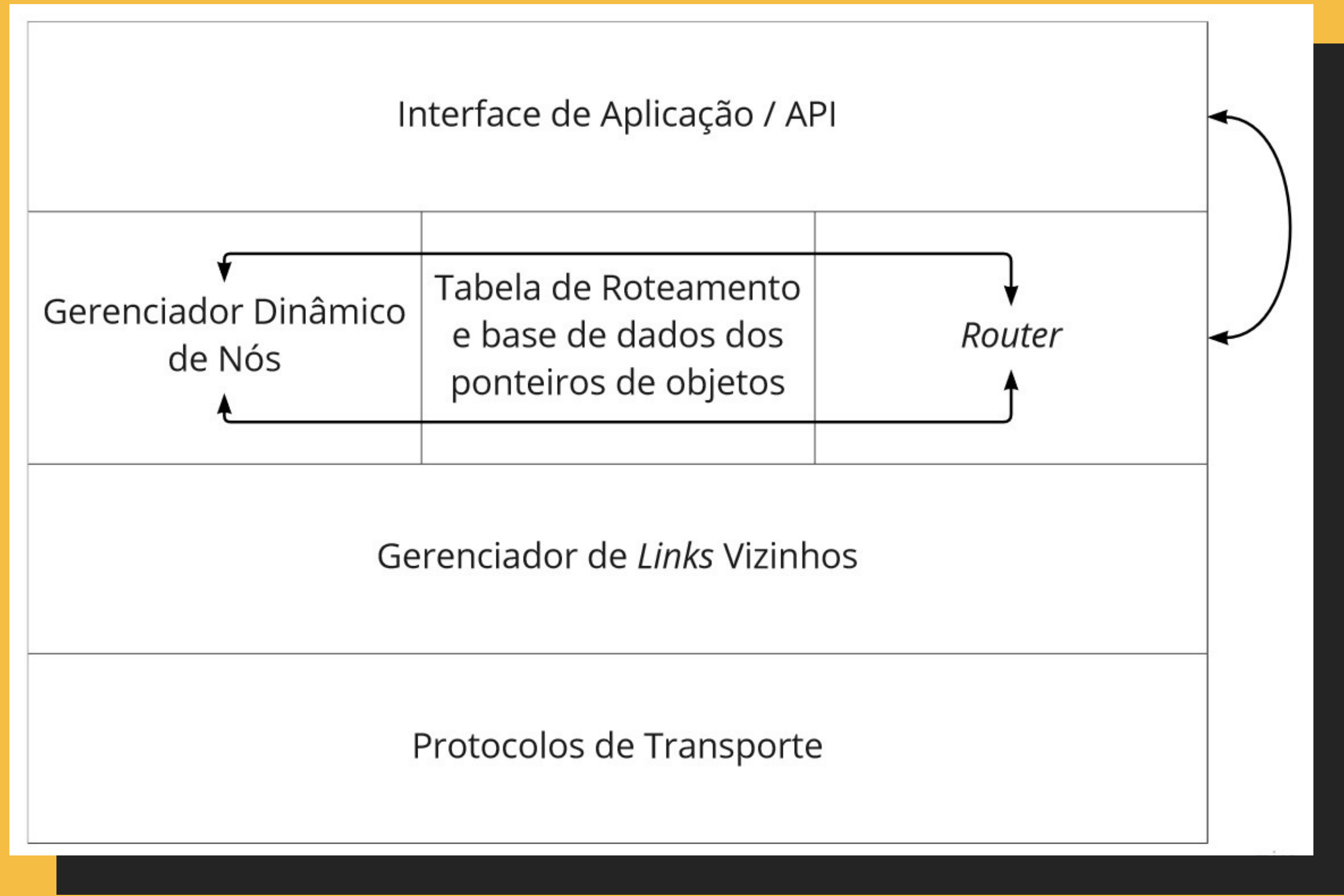


UM POUCO SOBRE O TAPESTRY



- Alta performance, escalável e que faz o uso apenas dos recursos de localização (*close-by endpoints*)
- Rede sobreposta estruturada
- Camada para a infraestrutura de Localização e Roteamento de Objetos Descentralizada (DOLR)
- Desde a inicialização leva em conta a distancia entre as redes, mantendo uma tabela ótima de rotas

Arquitetura de um nó Tapestry



Zhao B. Y. *et al.* Tapestry: A Resilient Global-Scale Overlay for Service Deployment. (Traduzido e editado)

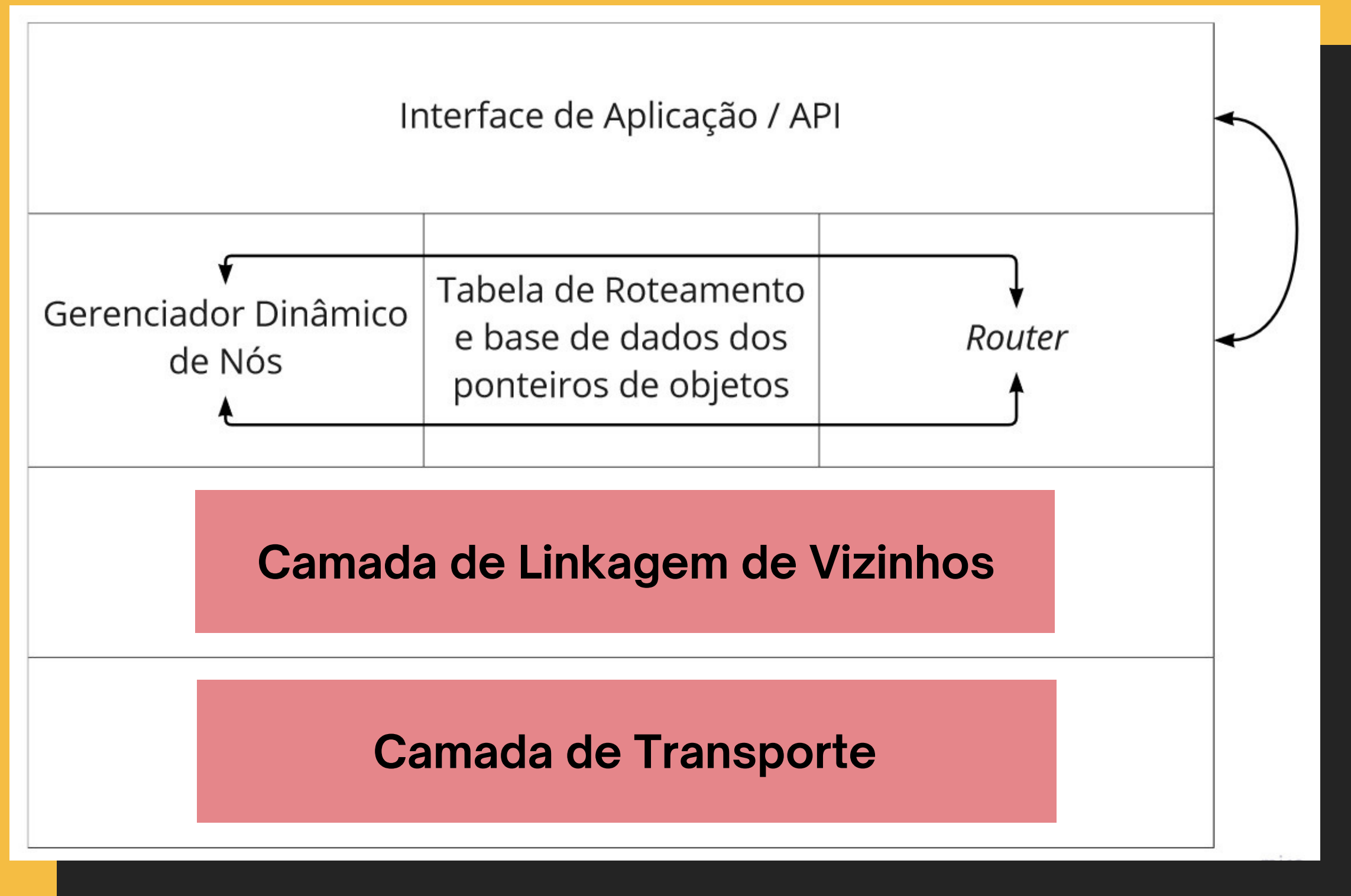


Arquitetura de um nó Tapestry



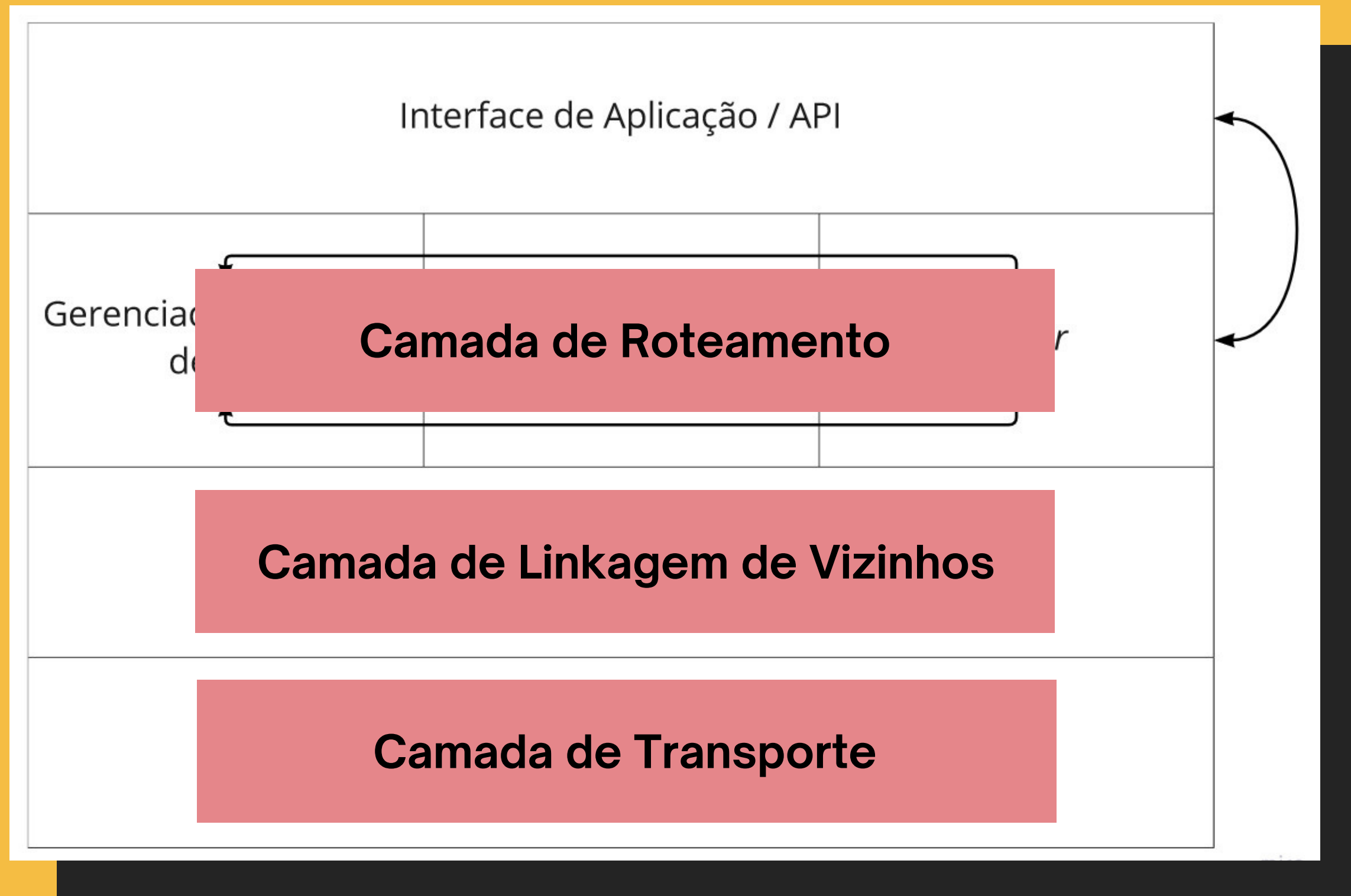
Zhao B. Y. *et al.* Tapestry: A Resilient Global-Scale Overlay for Service Deployment. (Traduzido e editado)

Arquitetura de um nó Tapestry



Zhao B. Y. *et al.* Tapestry: A Resilient Global-Scale Overlay for Service Deployment. (Traduzido e editado)

Arquitetura de um nó Tapestry



Zhao B. Y. *et al.* Tapestry: A Resilient Global-Scale Overlay for Service Deployment. (Traduzido e editado)



CHAMADAS DE API

FORWARD(*G, A, Msg*)

Invocado quando uma mensagem é recebida.
Deve invocar a chamada ROUTE().

ROUTE(*G, A, Msg, ProxHopNo*)

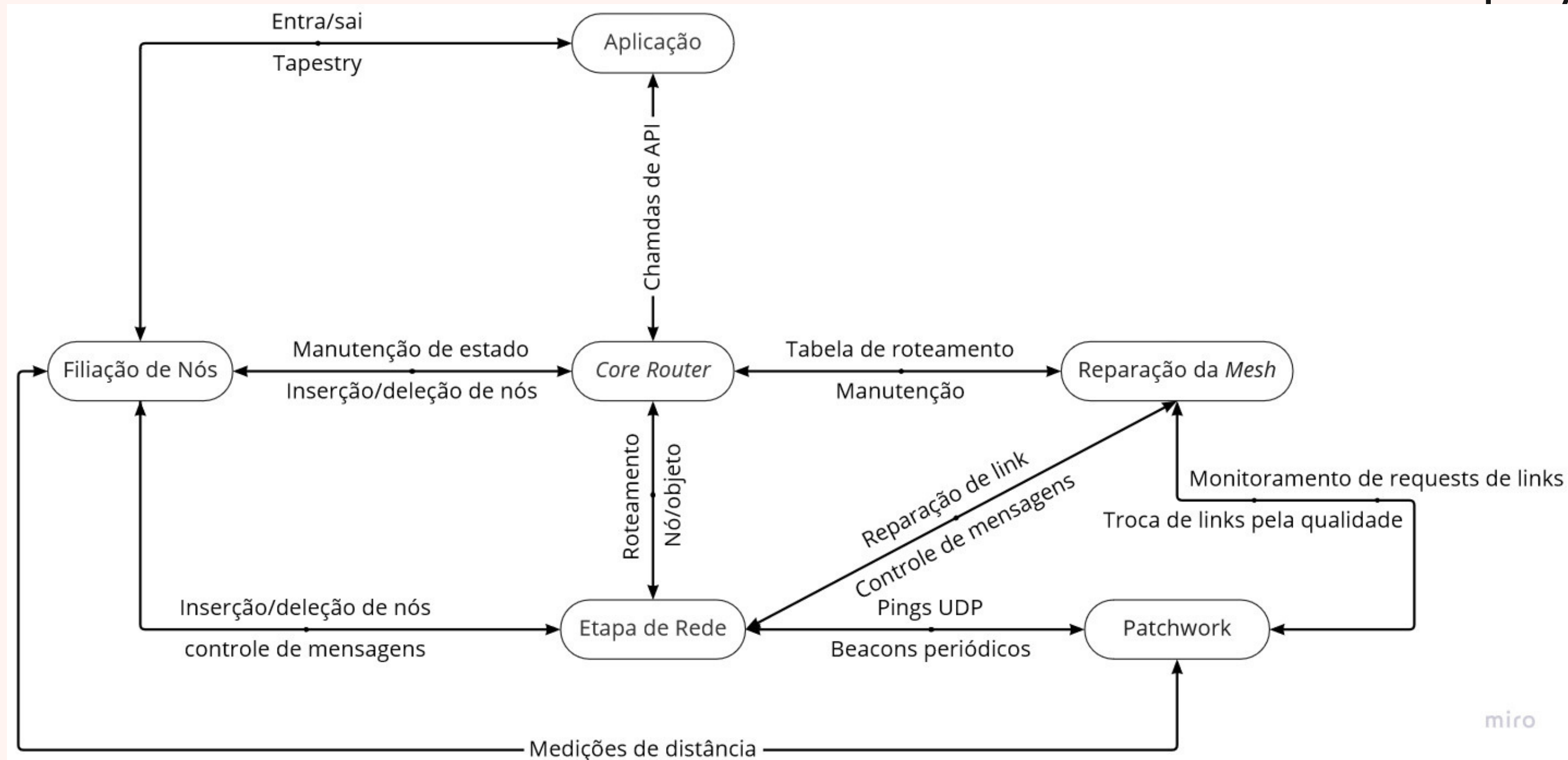
Invocado pelo *handler* ativado no FORWARD.
Usado para encaminhar a mensagem para o ProxHopNo.

DELIVER(*G, A, Msg*)

Invocado quando novas mensagens são recebidas para o Nó local.
Chamada assíncrona e retorna imediatamente.



Implementação Interna



Zhao B. Y. *et al.* Tapestry: A Resilient Global-Scale Overlay for Service Deployment. (Traduzido e editado)

ALGUMAS PERGUNTAS



O QUE FAZER QUANDO UM NOVO NÓ FOR INSERIDO ?

- a) Comunicação dos nós que **precisam saber**.
- b) Remanejamento de objetos caso o novo nó seja raiz.
- c) Construção da tabela ótima de roteamento.
- d) Notificação dos nós próximos, pois esse novo pode fazer parte da tabela de roteamento.



COMO MANEJAR UM NÓ QUE FOI DELETADO

Se tiver sido removido **voluntariamente**:

- Avisa os seus anteriores e substitui suas tabelas de rotas com um outro nó.
- Atualização da referência dos objetos daquele conjunto.

Se tiver sido removido **involuntariamente**:

- Faz o uso da redundância da tabela de rotas e dos objetos.
- Utilização dos *beacons* que detectam os erros e acionam a reparação de *Mesh*.



QUAL A PRINCIPAL DIFERENÇA ENTRE O DHT E O DOLR USADO

O DHT fixa o numero e a localização da réplica dos objetos na tabela.

O Trapestry (DOLR) permite as aplicações colocarem os objetos da forma como quiserem.

Publicação dos endereços dos objetos pela rede, facilitando o roteamento.





OBRIGADO PELA
ATENÇÃO!

