Maestro

Arthur Kazuo Tojo Costa 317497

Universidade Federal de São Carlos – Campus Sorocaba

Bacharelado em Ciência da Computação

Introdução

- Sistema Operacional de Redes
  - Detalhes do hardware
  - Multiplexação do uso do sistema
- Extensão do SOL
  - Modificações de hardwares de redes
  - Novo Ambiente
  - Complementando com o conjunto de funções básicas e de uso geral

Maestro: Primeiras Informações...

- Inspirado na arquitetura 4D
- Gerenciamento de Aplicações de Controle
- Interface de implementação
  - Modificação do Estado
  - Coordenação
- Plataforma para a realização de funções de rede automática e programáticas de controle

Maestro: Primeiras Informações...

- Funções de um roteador /switch:
  - transmissão de dados (comut. pacotes)
  - plano de controle de encaminhamento
- Máquina do controlador OpenFlow:
  (Plano de Controle)
  - Instalando e removendo as entradas de fluxo
  - Liberdade de controle

Maestro: Primeiras Informações...

Tamanho da Rede  **Número de Fluxos**

Se o controlador não consegue lidar
com os novos fluxos?

Desafio de conectar os milhares de servidores?

Problemas

- Voltados na construção de Controlador de OpenFlow:
  - inicializa a conexão de cada fluxo
  - contato diretamente com todos os switches

Problemas

- Voltados na construção de Controlador de OpenFlow:
  - inicializa a conexão de cada fluxo
  - contato diretamente com todos os switches
- Gargalo?

Problemas

- NOX:
  - Modelo simples de programação
  - Eventos *single-thread*
  - processamento individual por solicitação de fluxo
  - Sem paralelismo

80% do tempo de processamento do fluxo de pedido é gasto no envio destas solicitações

Propostas

- Exploração do Parelelismo!!
  - Processamento da requisição do fluxo
  - Sem depêndencia de dados complexos
  - Política de segurança
- Suporte:
  - Modelo Simples de Programação
  - Otimização de transferência

Propostas

- Além do OpenFlow, interfaces para:
  - Manutenção do estado da rede
  - Introdução de novas funções personalizadas
  - Adição de componentes de controle:
    - sequência de execução
    - estado da rede compartilhada dos componentes.

Implementação

- Sistema em Java

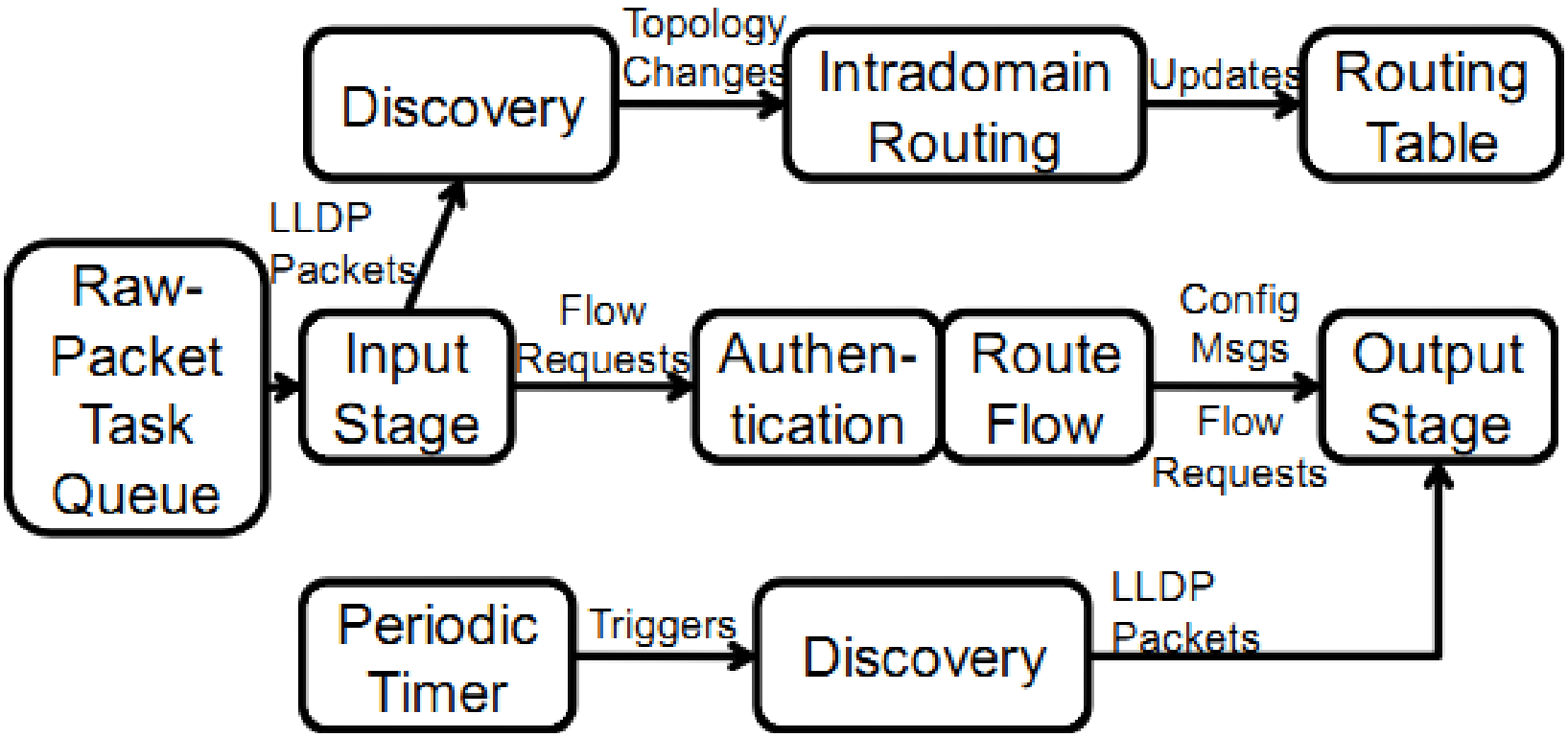
Implementação

- Sistema em Java:
  - Fáceis de escrever e manter
    - Nível de segurança para depurar
  - Carregamento Dinâmico sem recompilar
    - Ambiente flexível e expansível
  - Código Multi-Plataforma:
    - Java Virtual Machine
    - Sistema Portátil

Implementação

- Sistema em Java:
  - Eficiência
    - C / C++
  - Avaliado em Otimização
    - Escalabilidade

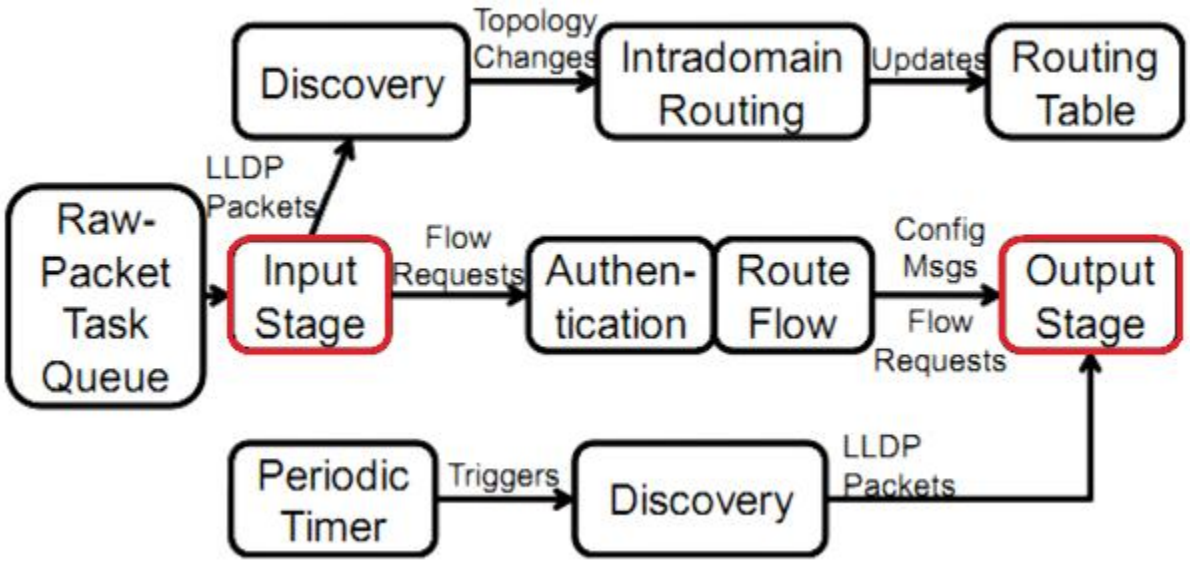
Estrutura Geral – Funcional



Estrutura Geral [I/O]

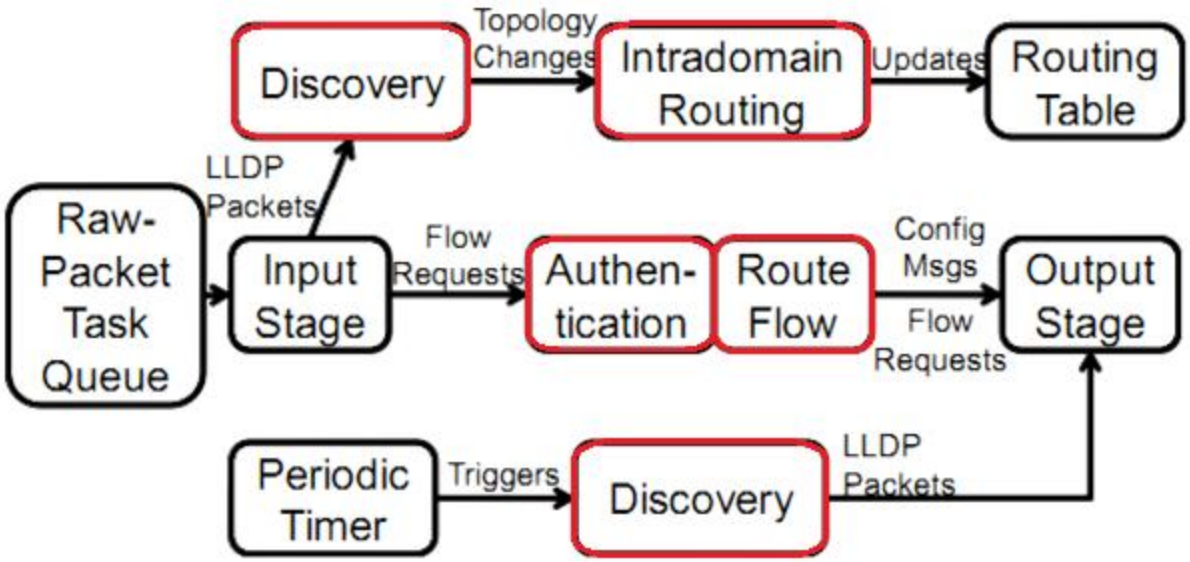
- Envia e recebe mensagens OpenFlow
- Entrada e Saída:
  - Alto → Baixo Nível

  (Fixas ao Protocolo OpenFlow)



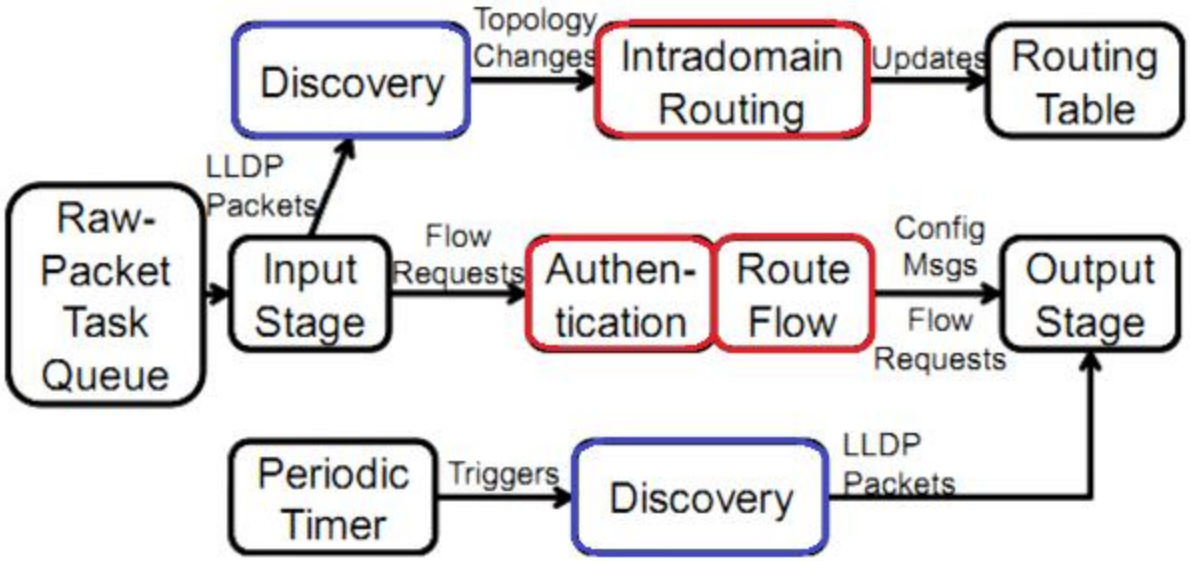
Estrutura Geral [Aplicações]

- Módulos de “Aplicações”
  - Alto Nível
  - Funcionalidades variam por implementações
- Flexível a modificação ou adição



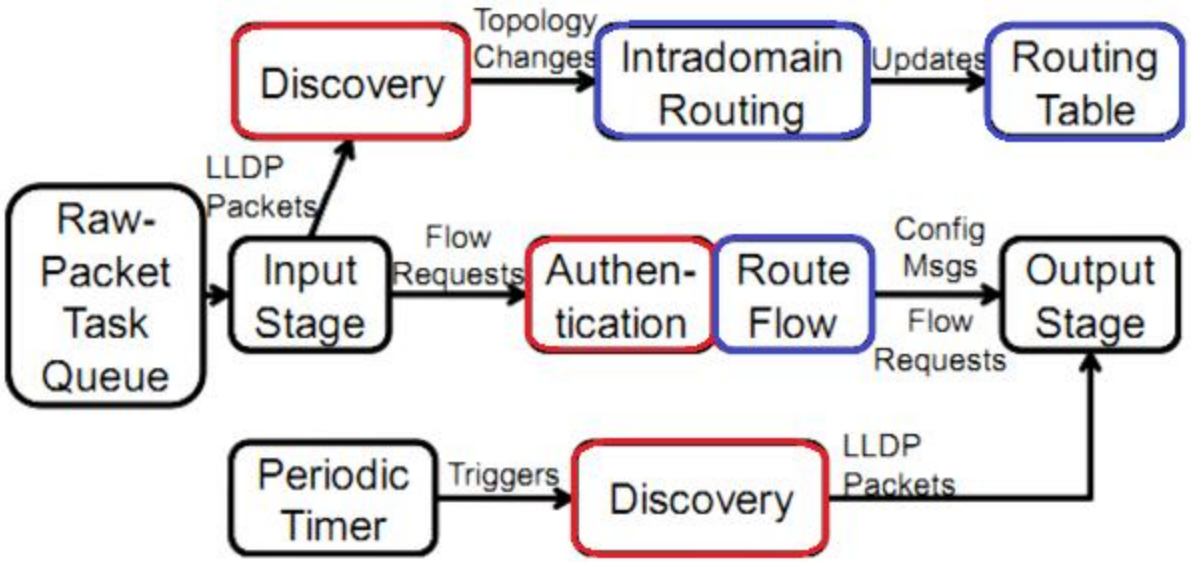
Estrutura Geral [Aplicações]

- Switch – [TCP] – Maestro
  - Envio periódico de sondagem aos vizinhos
  - Conformidade com o *Link Layer Discovery Protocol*
- Reconhecendo outros switches
  - Determina-se topologia da rede



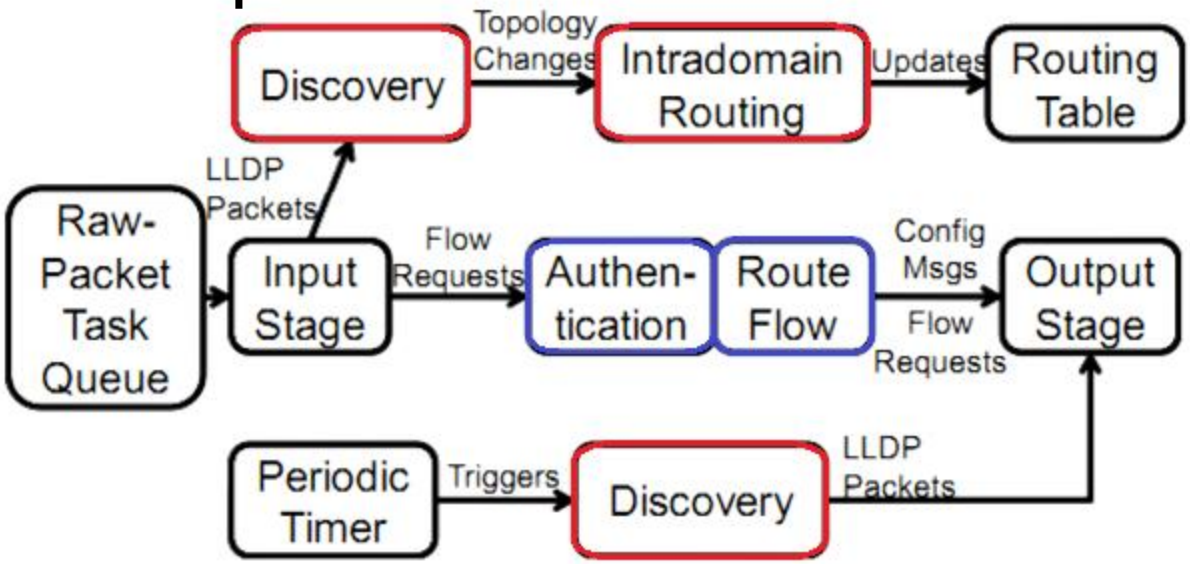
Estrutura Geral [Aplicações]

- Mudanças na Topologia
  - “Intradomain Routing” atualiza “Routing Table”
- Caminhos para os pedidos de fluxo
  - “Route Flow” utiliza “Routing Table”



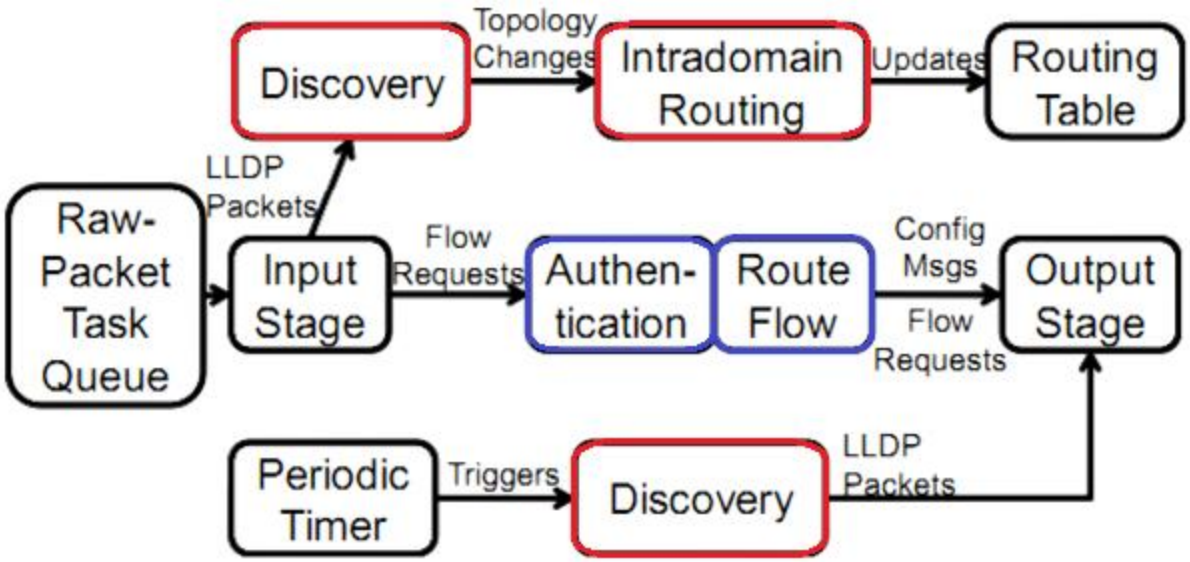
Estrutura Geral [Aplicações]

- Maestro recebe solicitação de fluxo
  - “Authentication” é requisitada
- Encontrar caminho para requisição autorizada
  - Através do “Route Flow”, comunicação entre switches
- Fase de fluxo do processo



Estrutura Geral [Aplicações]

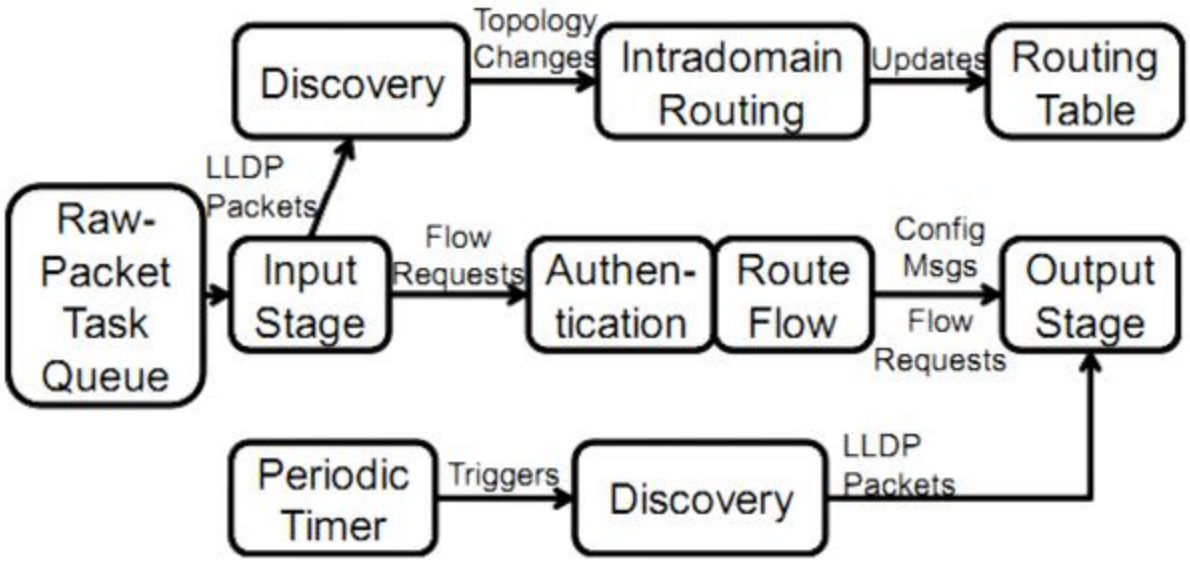
- Mensagens de config. de fluxo são enviadas
- Pacote de solicitação de fluxo é retornado



Estrutura Geral [Aplicações]

- Caminhos executados em paralelo!

- "Route Flow" em execução
  - Dados "antigos" da "Routing Table"



Estrutura Geral - Multi-Thread

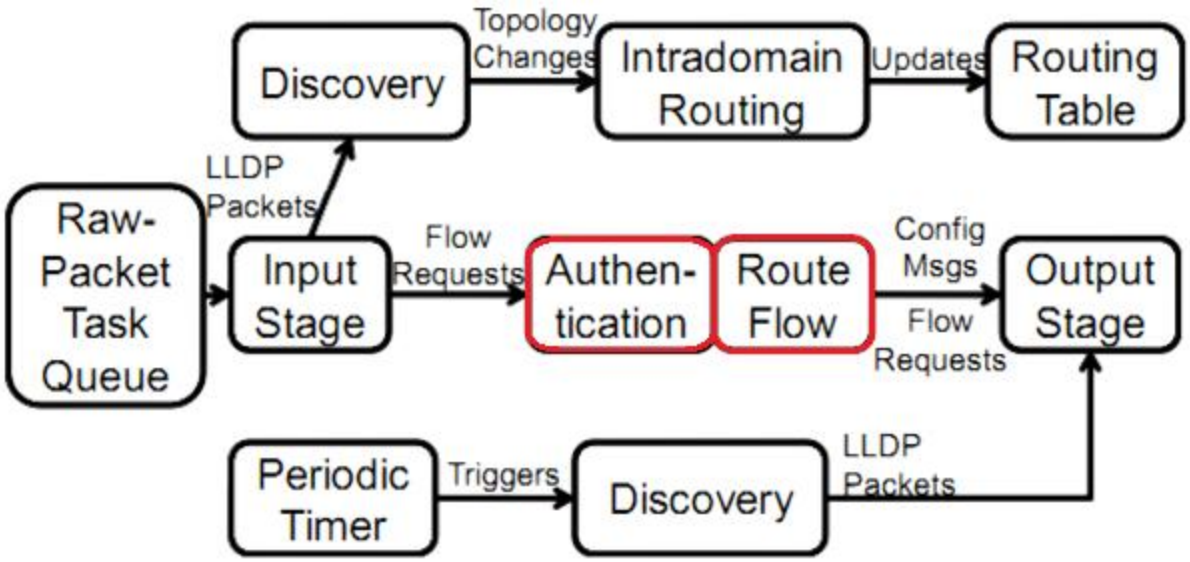
- Gerenciador de tarefas
  - estrutura unificada de interface
  - gerenciamento de computações

- Qualquer tarefa envolvida na classe java!

Número de Núcleos  **Número de Threads**

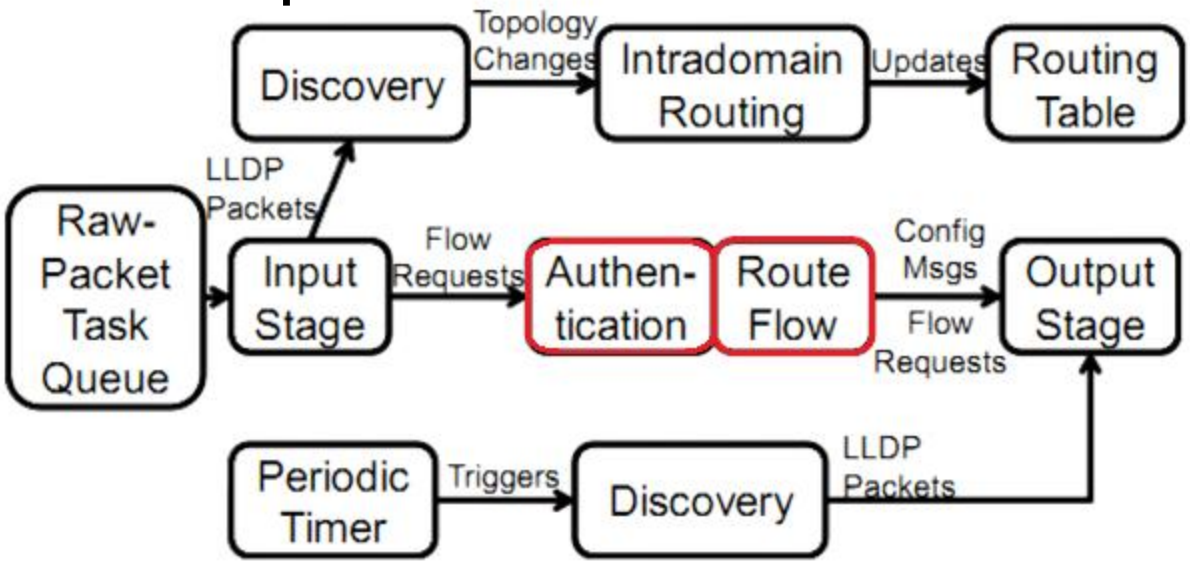
Estrutura Geral - Multi-Thread

- Pacotes OpenFlow recebidos + Cód. Entrada
  - Fila de Pacotes
- Qualquer thread pode associar a uma tarefa da fila
- Cód. Entrada = Pedidos de Fluxo (+ cód aplicação)



Estrutura Geral - Multi-Thread

- Ao final da fase de solicitação de fluxo:
- Saída:
  - Mensagens de Config Gerada
  - Solicitação de Fluxo
  - Código do estado de saída
- Threads executam para enviar ao switch de saída



Estrutura Geral - Multi-Thread

Aplicação simples e Single-Thread*

Múltiplas instâncias simultâneas por diferentes segmentos das threads

Opcional ao programador

Estrutura Geral - Multi-Thread

- Maestro:
  - Simples modelo de programação
  - Programadores de aplicativos não precisam lidar com multi-threading
- Paralelismo através da abstração do SO

Modelo Multi-Threading

- Definição de maneiras de execução otimizada:
  - Distribuição entre threads e núcleos
  - Minimização da sobre-carga entre núcleos
  - Sincronização do cache
  - Gasto de memória

Distribuir o trabalho uniformemente

"Esforços das threads e
núcleos bem distribuídos"

Distribuir o trabalho uniformemente

"Esforços das threads e
núcleos bem distribuídos"

- Pedido de fluxo pode exigir um número arbitrário de ciclos de CPU para processamento

Distribuir o trabalho uniformemente

"Esforços das threads e
núcleos bem distribuídos"

- Pedido de fluxo pode exigir um número arbitrário de ciclos de CPU para processamento

- Fila compartilhada entre threads
- Qualquer thread assume tarefa pendente

Minimizar a sobrecarga entre núcleos

“Sobrecarga no sistema quando pacotes são movidos de um núcleo para outro”

Minimizar a sobrecarga entre núcleos

"Sobrecarga no sistema quando pacotes são movidos de um núcleo para outro"

- Mais agravante entre processadores (cache)

Minimizar a sobrecarga entre núcleos

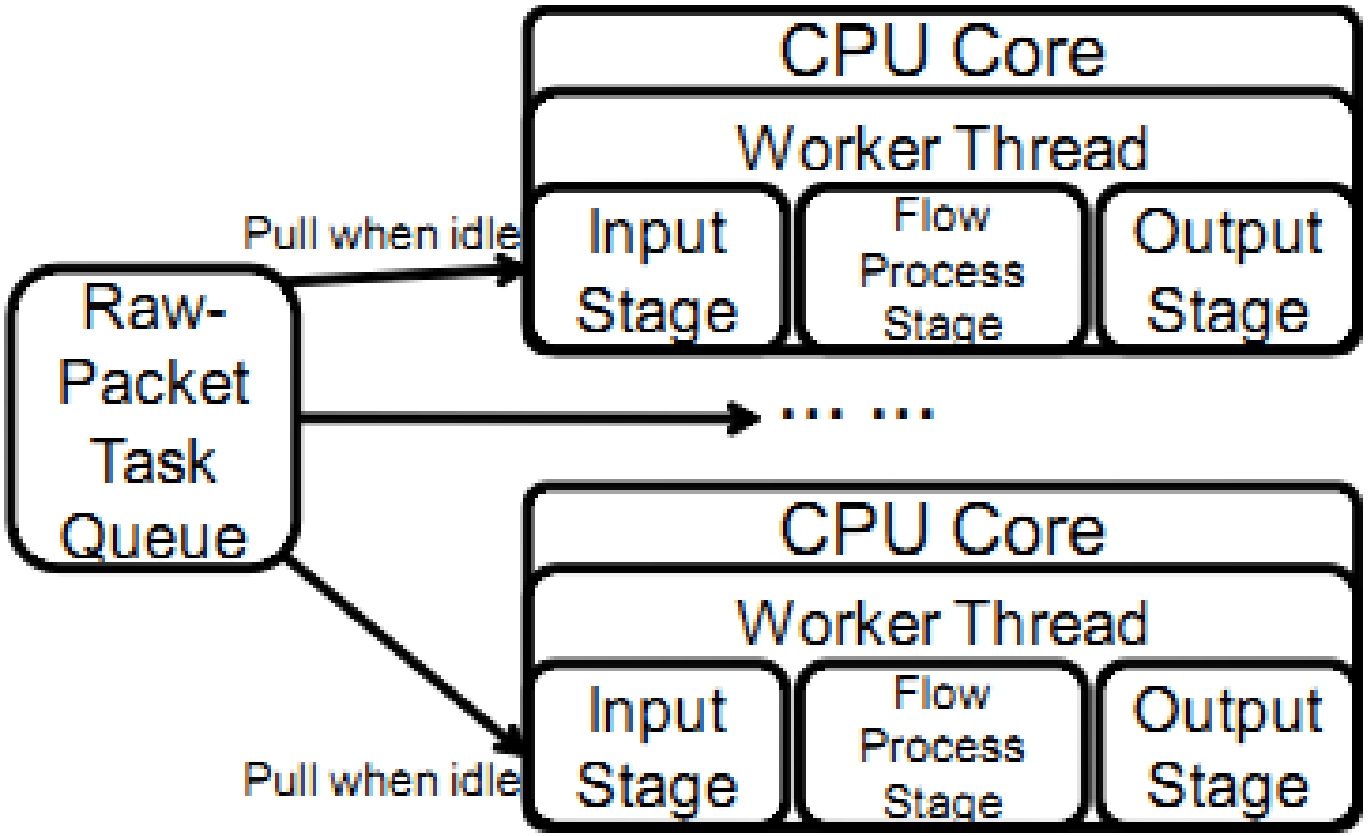
"Sobrecarga no sistema quando pacotes são movidos de um núcleo para outro"

- Mais agravante entre processadores

- "taskset": víncula thread ao núcleo específico
  - Evita *núcleo vinculativo*
- Um processador / requisição de fluxo
  - Evita "continuação" do *cache*

Minimizar a sobrecarga entre núcleos

"Sobrecarga no sistema quando pacotes são movidos de um núcleo para outro"



Minimizar o consumo de memória

“Minimizar o consumo de memória e assegurar um buffer para que os dados possam ser processados”

Minimizar o consumo de memória

"Minimizar o consumo de memória e assegurar um buffer para que os dados possam ser processados"

- Solicitações de fluxo de entrada excede a capacidade do sistema

Minimizar o consumo de memória

"Minimizar o consumo de memória e assegurar um buffer para que os dados possam ser processados"

- Solicitações de fluxo de entrada excede a capacidade do sistema

- Dados:
  - Pacotes em estado bruto;
  - Requisições de fluxo geradas pela entrada;
  - Mensagens de configurações geradas para saída;
  - Além da sobrecarga do Java
  - ... Irão acumular

Minimizar o consumo de memória

"Minimizar o consumo de memória e assegurar um buffer para que os dados possam ser processados"

- Soluções:
  - Definir limiar apropriado
    - Suficiente para requisições e alocações
  - Definir prioridade restrita no agendamento de tarefas

	Tarefas
Alta	Estágio de saída
Média	Processos de Fluxo
Baixa	Estágio de entrada

Perguntas?