OmniFlow

Integrating Load Balancing with multipath Flow Control in datacenter

guided by tutor zzqian

st: zhoulu

nju 20 may 2017

Outline

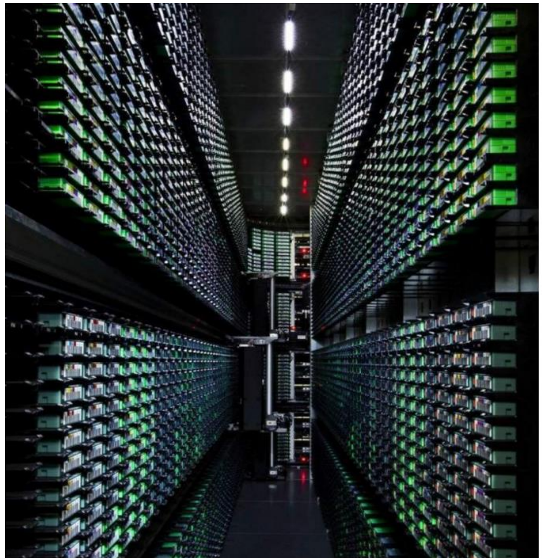
- *Background of this research*
- *The motivation to the design of OmniFlow*
- *The framework of the OmniFlow*
- *Existing Problem and solution(technical detail)*
- *Evaluation && parameter choose*
- *Conclusion and outlook*

Background of this research

To satisfy personal or business extensible computing demand

Cloud computing **：** a series of *service provided through Internet and relevant hardware platform.*

To support cloud computing : Need big datacenter to provide computing and network resource. (google: 36 data-centers all over the world：Google.cn,Google.com.hk)



The motivation to the design of OmniFlow

- *Tradition network transmission technology:*

  based on TCP/UDP

TCP(Transmission Control Protocol)

  congestion avoidance, slow start..(Aim at single path route)

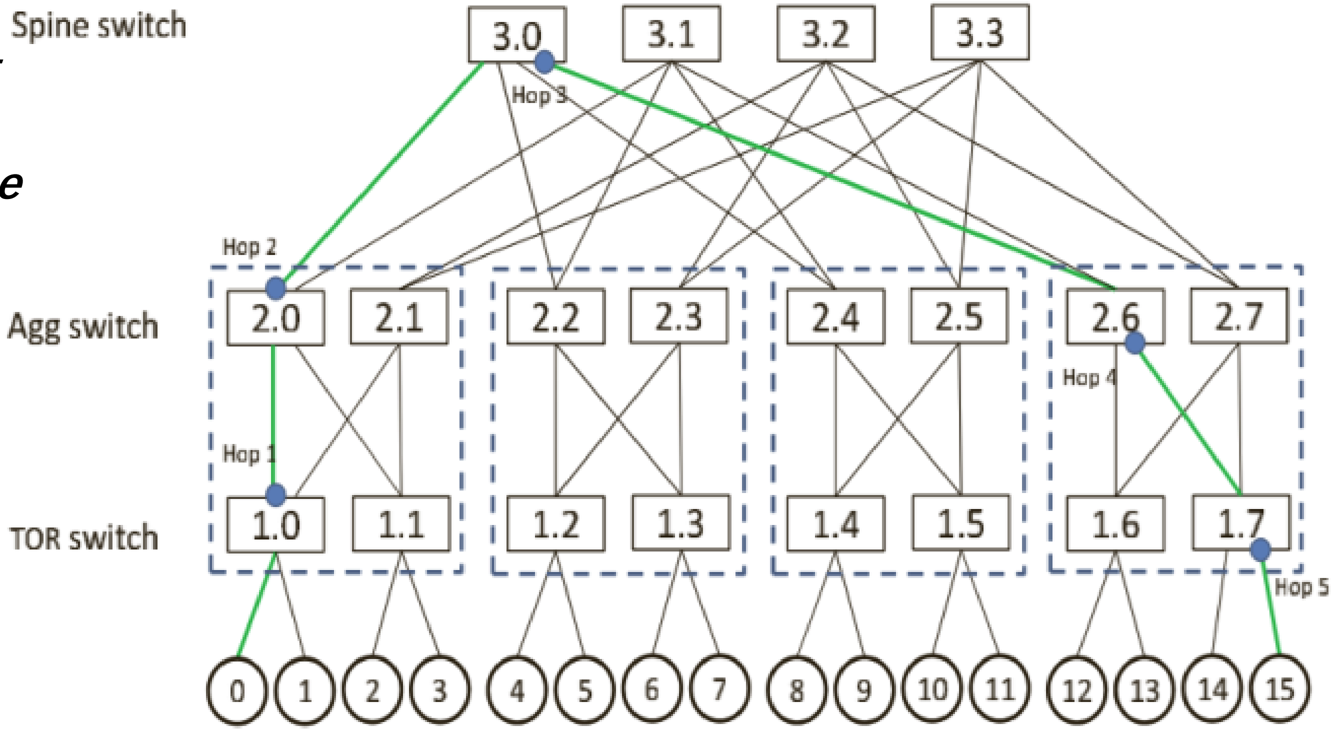
  But not suitable for datacenter:

- ➢ *network flow changes rapidly and exists traffic burst //*
- ➢ *abundant path redundancy in datacenter networks//*
- ➢ *high bandwidth, need reliable transmission technology and so on*

The motivation to the design of OmniFlow

Typical packet:
Delay-sensitive(web-service, instant messaging)
Low queue occupancies to guarantee predictable latency

Bandwidth-hungry transfer(video Steaming)
Large switch buffers to absorb traffic bursts



A typical Fat-tree for datacenter

Aim: achieve a proper balance between throughput and latency in a datacenter
Thus

Flow control &&Load balance

Flow control && load balance

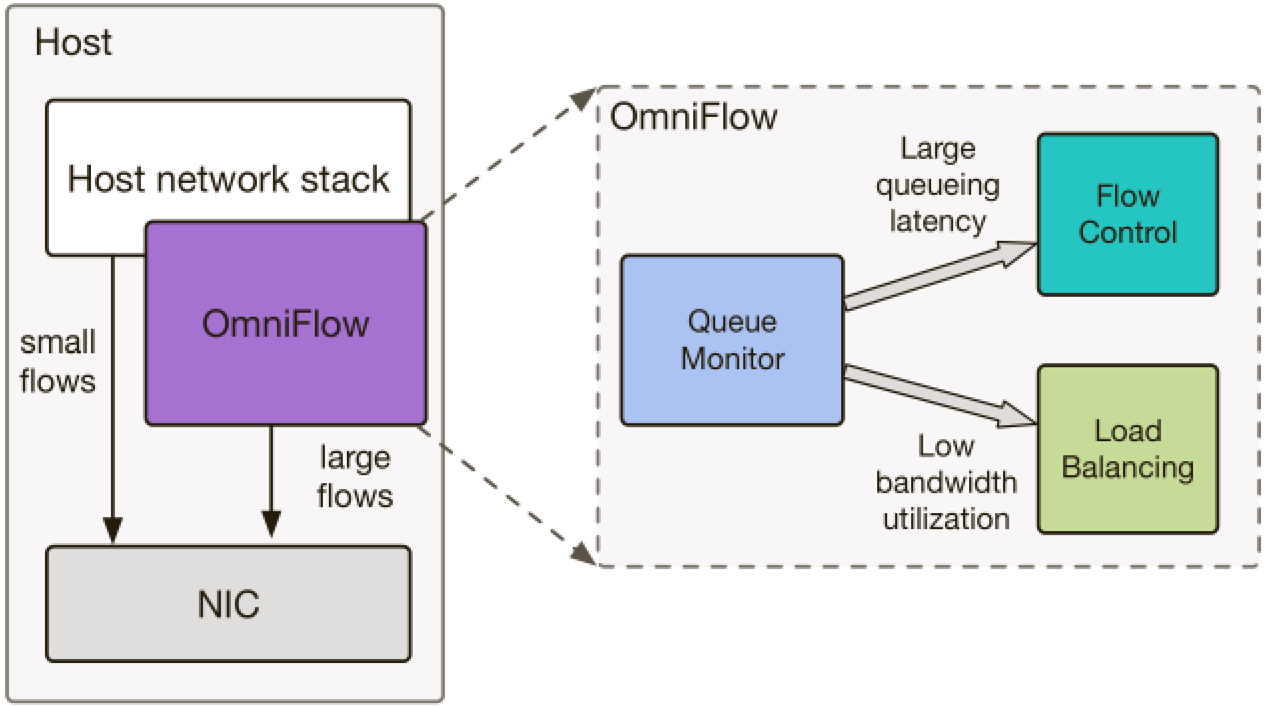
- ***Flow control：proactively decrease flow rates to reduce buffer occupancies（aim : decrease the queueing delay and data loss of short flow）***
- ***Load balance : distribute large flow uniformly into the network use the path diversity (aim : increase the throughput for large flows)***

What we are trying to do

- ***Using Load Balancing schemes to maintain low queueing latency on multiple paths***
- ***Take path redundant into consideration to get full use.***

The framework of OmniFlow

Aim: design a mechanism which can fully utilize the multiple paths between endpoints (without incurring long queueing latencies)



Thus we design an extra components:
queue monitor
to coupling Load Balancing with Flow Control

That is
① If the network is moderately-loaded ： choose Load Balancing to fully utilize the bandwidth
② If the transmission delay exceeds some threshold ： employ flow control to reduce the sending rate.

Note that our model OmniFlow mainly focus on adjusting the large flow in datacenter

Existing challenges to be solved

- ***How to simultaneously measure the queue lengths of multiple paths between two endpoints? (note the difference between the traditional network and datacenter)***

- ***When to invoke load balancing or flow control to optimize current transmission in OmniFlow?***

- ***How to effectively balance traffic across different paths without causing severe packet reordering?***

- ***How to adjust the flow rates to maintain low queueing latency on multiple paths through flow control?***

detect queue states in multiple paths

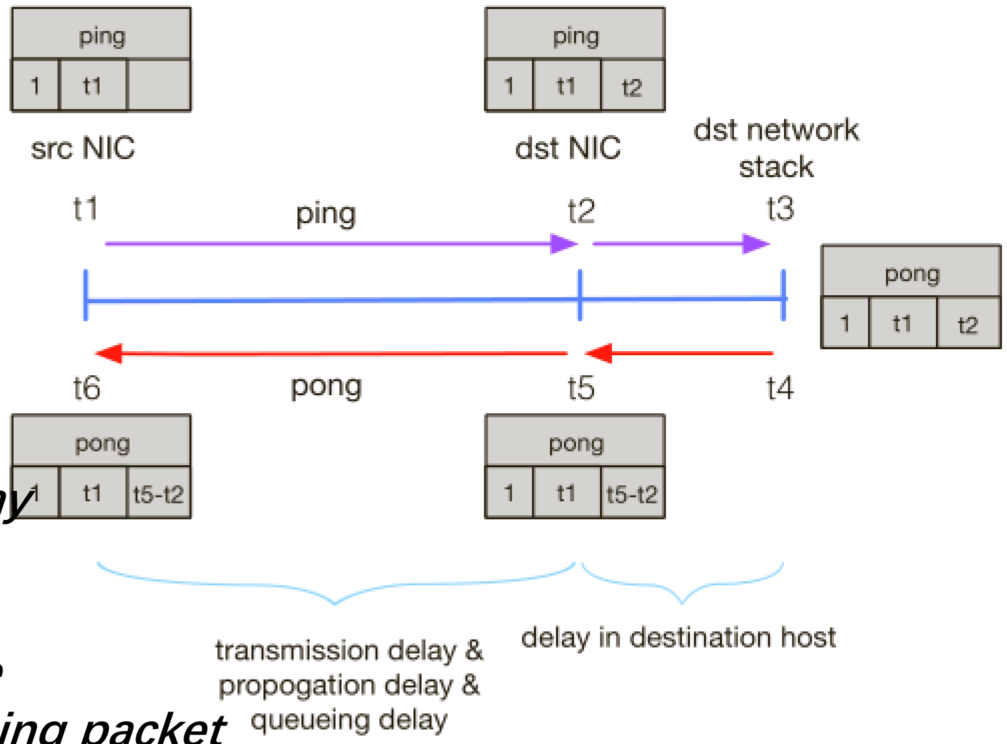
Aim: To measure the real-time queue states in the multiple paths network

Explicit congestion notification (ECN):

indicate the queue has exceeded the threshold by an ecn mark(weakness: can only reflect the congestion at a single switch)

Scheme in OmniFlow:

Add a UDP-based ping daemon to measure one way delay



For any pair of host, suppose there exists p paths ,denote L as the Length of the waiting queue， then every α RTTs， H_{src} send a ping packet Using the time relationship ,we can get the queueing length of all p Paths.

Decision Logic based on network states

Based on above latency measurement : ($L_1, L_2 \ldots L_p$)

Strategy :

- *Load balancing (if exist i, st $L_i < L_r$)*
- *Flow control （else）*

(here L_r is man-set threshold to restrict the queue length , Typically L_r wound be a relative small value to ensure low queueing latency)

If there exists a path p_i with $L_i < L_r$,then we can reroute flows onto p_i without violating the delay requirement , else, we have to adjust the flow rate.

Load Balance strategy

- *1: Transmits a flow f through a single path* P_i $\boldsymbol{first}$
- **2: If $\boldsymbol{L_i}$ >= threshold is detected, randomly reroute f onto a path**

  $\boldsymbol{P_i}$ $\boldsymbol{where}$ $\boldsymbol{L_i}$ < threshold
- **3: If $\boldsymbol{L_i}$ >= threshold , and on other healthy path can be found, those adjust its window size through flow control**

Flow control strategy

- ***Omniflow adjust the flow's windows size based on the multi-path congestion information.***

update its windows size by: $$cwnd = cwnd \cdot (1 - \frac{\Sigma_{i=1}^{\mathcal{P}} L_i}{f(\mathcal{L}_r)})$$

here f($\mathrm{L_r}$)is a function of $\boldsymbol{L_r}$ and we hope that the function is related to the parameter $\boldsymbol{L_r}$
And also can effectively help the network to drain the queued packets after congestion appears
Below we would like to derive the concrete form of the function:

denote path capacity : C
stable RTT : R
the congestion window at RTT(l) as $\boldsymbol{W^l}$

$$\sum_{i=1}^{N} W_i^l \approx C \cdot R + \mathcal{P} \cdot \mathcal{L}_r$$

$$\sum_{i=1}^{N} W_i^{l+1} = C \cdot R + \mathcal{P} \cdot \mathcal{L}_r + N = C \cdot R + \Sigma_{i=1}^{\mathcal{P}} L_i$$

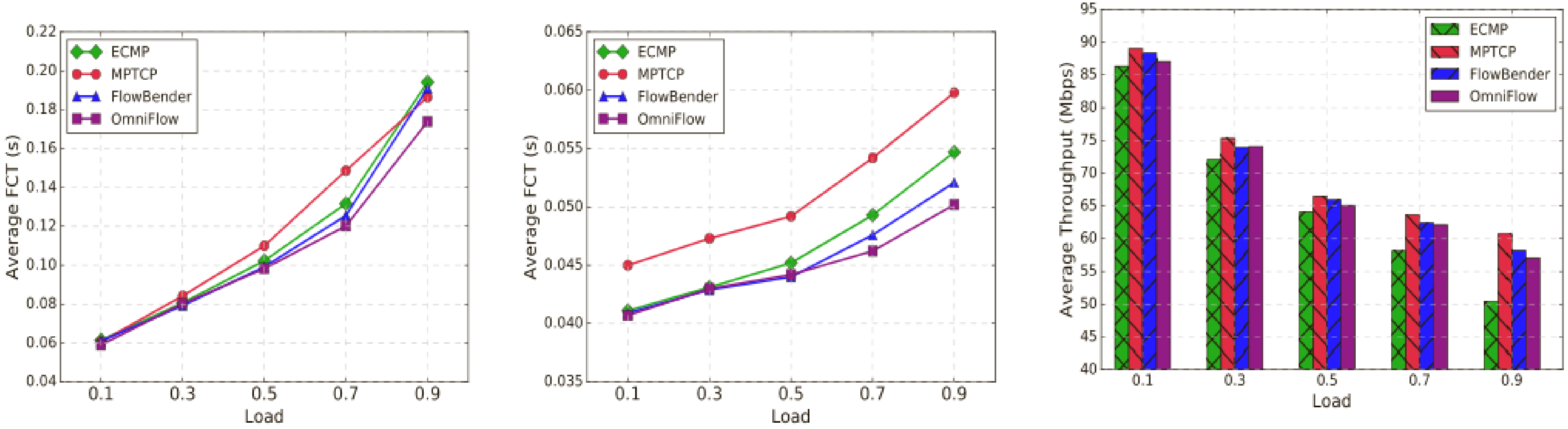
3: $$\sum_{i=1}^{N} W_i^{l+2} = (C \cdot R + \mathcal{P} \cdot \mathcal{L}_r + N) \cdot (1 - \frac{\Sigma_{i=1}^{\mathcal{P}} L_i}{f(\mathcal{L}_r)})$$

to guarantee the decrease of the window size is enough to drain the queued packets we let 3) equals to CR then we can get the form of the function

$$f(\mathcal{L}_r) = C \cdot R + \mathcal{P} \cdot \mathcal{L}_r + N$$

Evaluation

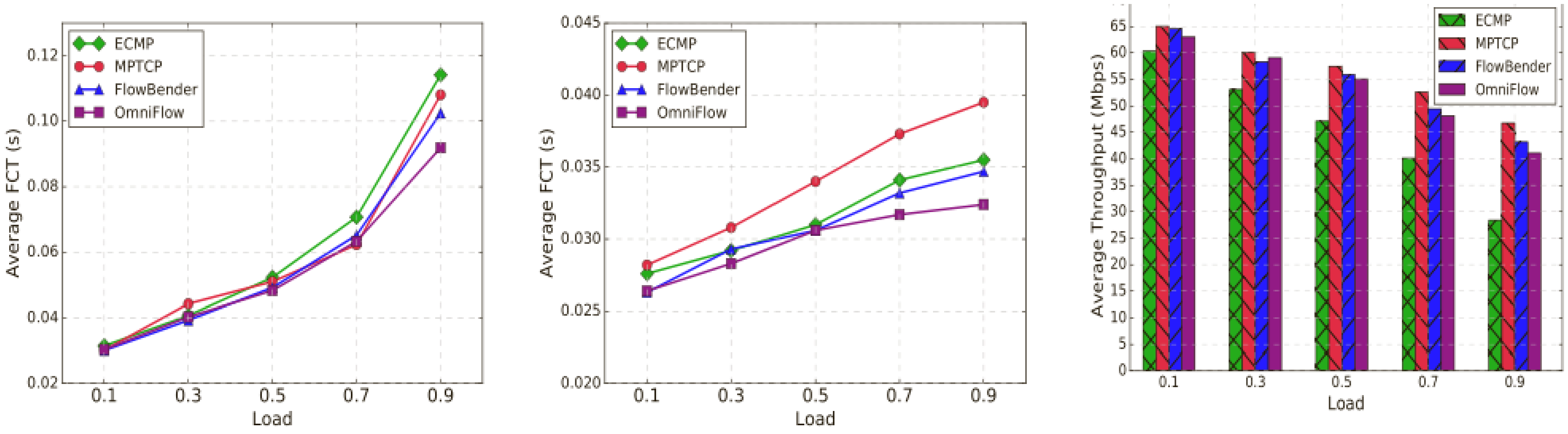
- ***Methodology & setting：***
  - *Platform ：network simulator version 2*
  - *Datacenter: fat-tree with 128 servers organized into 8 pods, each pod has 4 edge & aggregate switches*
  - *Thus every pair of host has 16 paths to connect (link capacity: 1Gbps, switch buffer 200 packets, packets size: 1500 bytes)*



(a) 整体流平均结束时间　(b) 短数据流平均结束时间　(c) 长数据流的平均吞吐率

图 4-6　网页搜索负载的实验结果

Evaluation



(a) 整体流平均结束时间 (b) 短数据流平均结束时间 (c) 长数据流的平均吞吐率

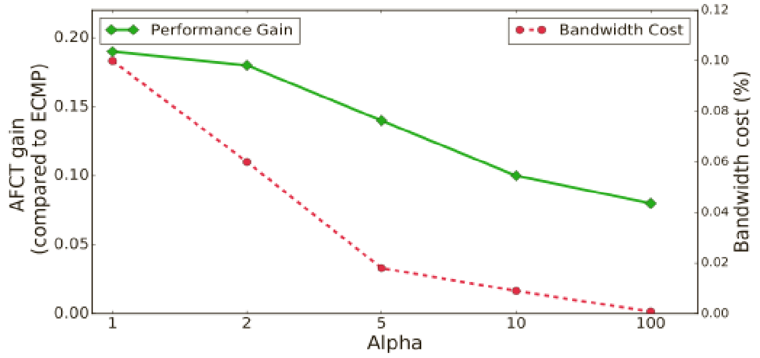
图 4-7 数据挖掘负载的实验结果

Ecmp： Equal-cost multi-path routing（randomly hashing flows onto different paths based on flow Id）
Mptcp: allowing a tcp connection to use multiple paths to maximize resource usage and increase redundancy.
FlowBender: for path

Parameter choose: α and $\boldsymbol{L_r}$

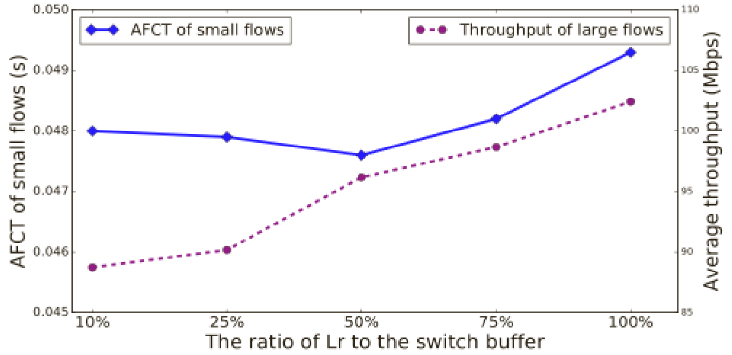
The extra band-width cost of the ping-pong packets is linear to the Probe interval α(but even α is 1, the band-width cost is still very small)

The AFCT gain(compared to ECMP) decreases rapidly
So in this experiments we suggests setting α to 2



(a) 不同 α 值下的结果

The average throughput increases as the value of $\boldsymbol{L_r}$ increase
(because large $\boldsymbol{L_r}$ allow the network to absorb more traffic burst)
The AFCT decrease



(b) 不同 $\mathcal{L}_r$ 下的结果

So in this experiments we suggests setting $\boldsymbol{L_r}$ to 50% ~75% of the switch buffer

Conclusion and outlook

- *OmniFlow: a novel datacenter transport which tightly couples Load balancing with Flow control in datacenter networks.*
  - *Leverage the UDP-based ping daemons to measure queue states accurately*
  - *Based on different network states. it adaptively reroutes flow onto healthy paths or adjusts the flow rates to reduce in-network queued packets*

Achievement

: strike a proper balance between throughput and latency

Deficiency

: UDP-based ping daemons still cost extra bandwidth resource

network states based flow adjust algorithm need more theoretical study

And so on

With the increase development of cloud computing, the datacenter transport wound be valuable

Thank you for your listening