A Benes Packet Network

Longbo Huang & Jean Walrand

EECS @ UC Berkeley
Longbo Huang
Institute for Interdisciplinary Information Sciences (IIIS)
Tsinghua University
Data centers are important computing resources

Provide most of our computing services

- Web service: Facebook, Email
- Information processing: MapReduce
- Data storage: Flickr, Google Drive

Google data centers within US
Data centers are important computing resources.

Google data centers within US
Data centers are important computing resources.

We focus on data center networking!

Google data centers within US
The data center networking problem

Networking is the foundation of data centers’ functionality
- Hundreds of thousands of interconnected servers
- Dynamic traffic flowing among servers
- Large volume of data requiring small latency
- Traffic statistical info may be hard to obtain
The data center networking problem

Networking is the foundation of data centers’ functionality

- Hundreds of thousands of interconnected servers
- Dynamic traffic flowing among servers
- Large volume of data requiring small latency
- Traffic statistical info may be hard to obtain

Questions:

- How to connect the servers?
- How to route traffic to achieve best rate allocation?
- How to ensure small delay?
- How to adapt to traffic changes?
Benes Network + Utility Optimization + Backpressure

Benes Network:
- High throughput
- Small delay (logarithmic in network size)
- Connecting 2N servers with $O(N\log N)$ switch modules

Flow Utility Maximization
- Ensure best allocation of resources

Backpressure:
- Throughput optimal
- Robust to system dynamics
- Require no statistical info
Benes Network

Building a $2^n \times 2^n$ Benes network
Benes Network

Routing circuits:

1 → 3
2 → 1
3 → 2
4 → 4
Benes Network

Routing circuits:

1 → 4
2 → 2^n
3 → 1
4 → 2^n – 1
......
2^n – 1 → 2
2^n → 3

- non-blocking for circuits
- full-throughput for packets
The flow utility maximization problem:

\[
\max : \sum_{sd} U_{sd}(r_{sd})
\]

s.t. Stability
Backpressure can be directly applied. However, each node needs $2^n$ queues, one for each destination.

- Random arrival $A_{sd}(t)$
- Flow control, admit $R_{sd}(t)$ in $[0, A_{sd}(t)]$
- Each $(s, d)$ flow has utility $U_{sd}(r_{sd})$
- Each link has capacity 1pk/s
Grouped-Backpressure (G-BP)

The idea:
- Divide traffic into two groups
- Perform routing & scheduling on the mixed traffic
- Rely on Backpressure & symmetry for stability

Key components
1. A fictitious reference system for control
2. A special queueing structure
3. An admission & regulation mechanism
4. Dynamic scheduling
G-BP Component 1 - Reference System

These nodes remain the same
G-BP Component 2 – Queueing Structure

- Each switch node in columns 1 to n-1 maintains 4 queues (same for both systems)
G-BP Component 2 – Queueing Structure

- Each input server in column 0 maintains 2 queues (same for both systems)
G-BP Component 2 – Queueing Structure

- Each node in column $n$ maintains 2 queues for $D_1$ and $D_2$ (also in the physical system)
G-BP Component 2 – Queueing Structure

- Each node in columns n to 2n-1 maintains 2 queues (only the physical system)
G-BP Component 3 – Admission & Regulation

Admission queue at input:

\[ H_{sd}(t) \]

\[ \gamma_{sd}(t) \rightarrow H_{sd}(t) \rightarrow R_{sd}(t) \]

Regulation queue at output:

\[ q_d(t) \]

\[ \sum_s R_{sd}(t) \rightarrow \quad 1 - \epsilon \]
G-BP Component 3 – Admission & Regulation

Admission decisions at input:
- Update $\gamma_{sd}(t)$:
  $$\max : \quad VU_{sd}(\gamma_{sd}(t)) - H_{sd}(t)\gamma_{sd}(t),$$
  $$\text{s.t.} \quad 0 \leq \gamma_{sd}(t) \leq A_{\text{max}}$$

- Admit packets:
  $$(\text{up flow to } d \text{ in } D_1)$$
  $$R_{sd}(t) = \begin{cases} 
    A_{sd}(t) & \text{if } H_{sd}(t) > Q_m^1(t) + q_d(t), \\
    0 & \text{else.}
  \end{cases}$$

Note: $q_d(t)$ is “idealized”

In practice:
- delayed arrivals at $d$
- delayed feedback to $s$
G-BP Component 3 – Admission & Regulation

Admission decisions at input:
- Update $\gamma_{sd}(t)$:
  \[ \max : \quad VU_{sd}(\gamma_{sd}(t)) - H_{sd}(t)\gamma_{sd}(t), \]
  \[ \text{s.t.} \quad 0 \leq \gamma_{sd}(t) \leq A_{\text{max}} \]
- Admit packets:
  \( R_{sd}(t) = \begin{cases} A_{sd}(t) & \text{if } H_{sd}(t) > Q_{m}^{1}(t) + q_{d}(t), \\ 0 & \text{else.} \end{cases} \)

The need to admit

Input server admits pkts

Source congestion

Destination congestion, passed to source
Admission decisions at input:

- Update $\gamma_{sd}(t)$:
  
  \[
  \max : \quad V U_{sd}(\gamma_{sd}(t)) - H_{sd}(t) \gamma_{sd}(t),
  \]
  
  s.t. \quad 0 \leq \gamma_{sd}(t) \leq A_{\text{max}}

- Admit packets:
  \[
  R_{sd}(t) = \begin{cases} 
  A_{sd}(t) & \text{if } H_{sd}(t) > Q_m^2(t) + q_d(t), \\
  0 & \text{else.}
  \end{cases}
  \]

(low flow to d in $D_2$)
Admission decisions at input:

- Update $\gamma_{sd}(t)$:

$$\max : VU_{sd}(\gamma_{sd}(t)) - H_{sd}(t)\gamma_{sd}(t),$$

s.t. $0 \leq \gamma_{sd}(t) \leq A_{\text{max}}$

- Admit packets:

$$R_{sd}(t) = \begin{cases} A_{sd}(t) & \text{if } H_{sd}(t) > Q^2_m(t) + q_d(t), \\ 0 & \text{else.} \end{cases}$$

(low flow to d in $D_2$)

The need to admit

Input server *rejects* pkts

Source congestion

Destination congestion, passed to source
Grouped-Backpressure

Admission control
G-BP Component 4 – Dynamic Scheduling

Which flow to serve over this link?
Define flow weights:

\[ W^{1U} = \left[ 2Q^{1U}_m(t) - Q^{1U}_{m'}(t) - Q^{1L}_{m'}(t) \right]^+ \]
G-BP Component 4 – Dynamic Scheduling

Define flow weights:

\[
W^{1U} = \left[ 2Q_{m}^{1U}(t) - Q_{m'}^{1U}(t) - Q_{m'}^{1L}(t) \right]^{+}
\]

\[
W^{2U} = \left[ 2Q_{m}^{2U}(t) - Q_{m'}^{2U}(t) - Q_{m'}^{2L}(t) \right]^{+}
\]
G-BP Component 4 – Dynamic Scheduling

- If $W_1^{1U} > W_2^{2U}$ & $W_1^{1U} > 0$, send 1U packets over link [m, m’]
- At m’, randomly put the arrival into 1U or 1L
G-BP Component 4 – Dynamic Scheduling

- If $W_1^U < W_2^U$ & $W_2^U > 0$, send 2U packets over link $[m, m']$
- At $m'$, randomly put the arrival into 2U or 2L
G-BP Component 4 – Dynamic Scheduling

- If queue is not empty, transmit packet
- Else remain idle
Grouped-Backpressure

Admission control

G-Backpressure based on fic sys
G-BP Component 4 – Dynamic Scheduling

- If queue is not empty, transmit packet
- Place packets into corresponding queues
Grouped-Backpressure

Admission control

Free-flow forwarding

G-Backpressure based on fic sys
Grouped-Backpressure – Performance

Theorem: Under the G-BP* algorithm, (i) both physical & fictitious networks are stable, and (ii) we achieve:

\[ U(r^{G-BP}) \geq U(r^{opt}) - O\left(\frac{1}{V} + \epsilon\right) \]

* This is the idealized algorithm ....
Grouped-Backpressure – Performance

**Theorem:** Under the G-BP algorithm, (i) both physical & fictitious networks are *stable*, and (ii) we achieve:

\[ U(r^{G-BP}) \geq U(r^{opt}) - O\left(\frac{1}{V} + \epsilon\right) \]

**Remarks:**
- No statistical info is needed
- Distributed hop-by-hop routing & scheduling
- Four queues per node (BP needs \(2^n\))
Grouped-Backpressure – Analysis Idea

- **Update** $\gamma_i(t)$
  \[
  \text{max : } VU(\gamma_i) - H_i(t)\gamma_i
  \]
  \[
  \text{s.t. } \gamma_i \in [0, A_{\max}]
  \]

- **Admit packets:**
  - If $H_i(t) > Q_i(t) + q(t)$, admit arrivals
  - Else, do not admit
Grouped-Backpressure – Analysis Idea

- **Update \( \gamma_i(t) \)**
  \[
  \max : \quad VU(\gamma_i) - H_i(t)\gamma_i \\
  \text{s.t.} \quad \gamma_i \in [0, A_{\text{max}}]
  \]

- **Admit packets:**
  - If \( H_i(t) > Q_i(t) + q(t) \), admit arrivals
  - Else, do not admit

\( H_1(t), H_2(t) \) are bdd

\( q(t) \) is bounded
Grouped-Backpressure – Analysis Idea

Admission queue
\[ \gamma_1(t) \rightarrow H_1(t) \rightarrow R_1(t) \]

Upper layer
\( R_1(t) \rightarrow Q_1 \rightarrow Q_3 \) 0.5 \( \rightarrow Q_5 \) 0.5
\( R_2(t) \rightarrow Q_2 \rightarrow Q_4 \) 0.5 \( \rightarrow Q_6 \) 0.5

Rates into \( Q_5(t) \), \( Q_6(t) \) are \((1-\epsilon)/2 < 0.5\)

Regulation queue
\[ R_1(t) + R_2(t) \rightarrow q(t) \rightarrow 1 - \epsilon \]

\( q(t) \) is bounded

\( H_1(t), H_2(t) \) are bdd

\( r_1 + r_2 \leq 1 - \epsilon \)

\( Q_5(t), Q_6(t) \) stable

N1: BP  N2: FF
Grouped-Backpressure – Analysis Idea

Admission queue
\[ \gamma_1(t) \rightarrow H_1(t) \rightarrow R_1(t) \]

Upper layer
\[ R_1(t) \rightarrow Q_1 \rightarrow Q_3 \]

Upper layer
\[ R_2(t) \rightarrow Q_2 \rightarrow Q_4 \]

Regulation queue
\[ R_1(t) + R_2(t) \rightarrow q(t) \rightarrow 1 - \epsilon \]

\[ Q_3 \rightarrow Q_5 \]
\[ Q_4 \rightarrow Q_6 \]

\[ Q_5(t), Q_6(t) \text{ stable} \]

\[ Q_1(t) - Q_4(t) \text{ stable by Backpressure} \]

Network stability

N1: BP

N2: FF
Grouped-Backpressure – Intuition

The flow optimization problem:

\[
\max : \quad VU(r) \\
\text{s.t.} \quad r \leq 1
\]

Due to the random arrival

The augmented & relaxed flow opt problem:

\[
\max : \quad VU(\gamma) \\
\text{s.t.} \quad \gamma \leq r \\
\quad \quad r \leq 1 - \epsilon
\]

Taking the dual decomposition

The dual form:

\[
g(H, q) = \sup_{\gamma, r} \left\{ VU(\gamma) - H(\gamma - r) - q(r - (1 - \epsilon)) \right\} \\
= \sup_{\gamma, r} \left\{ VU(\gamma) - H\gamma + (H - q)r + q(1 - \epsilon) \right\}
\]
Grouped-Backpressure – Intuition

The flow optimization problem:

\[ \begin{align*}
\text{max} & : \quad VU(r) \\
\text{s.t.} & : \quad r \leq 1
\end{align*} \]

Due to the random arrival

The augmented & relaxed flow opt problem:

\[ \begin{align*}
\text{max} & : \quad VU(\gamma) \\
\text{s.t.} & : \quad \gamma \leq r \\
& : \quad r \leq 1 - \epsilon
\end{align*} \]

Taking the dual decomposition

The dual form:

\[ g(H, q) = \sup_{\gamma, r} \left\{ VU(\gamma) - [H(\gamma - r)] + q(r - (1 - \epsilon)) \right\} \]

\[ = \sup_{\gamma, r} \left\{ VU(\gamma) - H\gamma + (H - q)r + q(1 - \epsilon) \right\} \]
Grouped-Backpressure – Proof Steps

Step 1 - Define a Lyapnov function:

\[ L(t) \triangleq \frac{1}{2} H^2(t) + \frac{1}{2} q^2(t) \]

Step 2 - Compute a Lyapunov drift \( \Delta(t) = \mathbb{E}\{ L(t+1) - L(t) \mid X(t) \} \)

\[ \Delta(t) - V \mathbb{E}\{ U(\gamma(t)) \mid X(t) \} \]

\[ \leq B - \mathbb{E}\{ VU(\gamma(t)) + H(t)[R(t) - \gamma(t)] + q(t)[1 - \epsilon - R(t)] \mid X(t) \} \]

\[ = B - \mathbb{E}\{ VU(\gamma(t)) - H(t)\gamma(t) + [H(t) - q(t)] R(t) + q(t)(1 - \epsilon) \mid X(t) \} \]

Step 3 - Plug in the opt solution of the relaxed problem, \( \gamma_{\epsilon}^* = r_{\epsilon}^* \)

\[ \Delta(t) - V \mathbb{E}\{ U(\gamma^{GBP}(t)) \mid X(t) \} \]

\[ \leq B - \mathbb{E}\{ VU(\gamma_{\epsilon}^*) + H(t)[R_{\epsilon}^* - \gamma_{\epsilon}^*] + q(t)[1 - \epsilon - R_{\epsilon}^*] \mid X(t) \} \]

\[ \leq B - VU(\gamma_{\epsilon}^*) \]

Step 4 - Do a telescoping sum

\[ U(\bar{\gamma}^{GBP}) \geq U(r^*) - \frac{B}{V} - O(\epsilon) \]

Step 5 - H(t) is stable

\[ \bar{\gamma}^{GBP} \leq \bar{r}^{GBP} \Rightarrow U(\bar{r}^{GBP}) \geq U(r^*) - \frac{B}{V} - O(\epsilon) \]
Grouped-Backpressure – Simulation*

Setting: 16x16 Benes network, \( \varepsilon=0.01 \), utility=\( \log(1+r) \)

* This is the idealized algorithm ....
Grouped-Backpressure – Simulation

Setting: 16x16 Benes network, $\epsilon=0.01$, utility=$\log(1+r)$

Note: For 1Gbps links and 500-Byte packets
Grouped-Backpressure – Simulation

Delay versus network size – logarithmic growth
\[ V=20, \varepsilon=0.01 \]

Delay reduced by “biasing” BP

![Graph showing delay versus network size with logarithmic growth and V=20, \varepsilon=0.01]
Assume each packet has 500 bytes, each link has 1Gbit/second. Then every slot is 4 microsecond.

Grouped-Backpressure – Simulation

Delay versus network size – logarithmic growth

\( V=20, \epsilon=0.01 \)
Grouped-Backpressure – Simulation

Setting: 16x16 Benes network, ε=0.01, utility=wlog(1+r)

Adaptation to change of traffic – At time 5, weights $w_{sd}$ change
Summary

- Using **Benes network** and **Backpressure** for data center networking
  - **Scalable:** built with basic switch modules
  - **Simple:** four queues per node
  - **Small delay:** logarithmic in network size
  - **High throughput:** supports all rates in capacity region
  - **Distributed:** hop-by-hop routing and scheduling

- Future research: Implementation issues
Thank you very much!

More info: [www.eecs.berkeley.edu/~huang](http://www.eecs.berkeley.edu/~huang)