Layering As Optimization Decomposition

Mung Chiang
Electrical Engineering Department, Princeton University

IEEE CTW
May 22, 2006

Nature of the Talk

- Give an overview of the topic. Details in various papers
- Not exhaustive survey. Highlight the key ideas and challenges
- Biased presentation. Focus on work by my collaborators and me

This is an appetizer. The beef in the papers

Apology, Apology, Apology for any missing reference

Outline

- Background: Holistic view on layered architecture
- Background: NUM and G.NUM

- Horizontal Decompositions
- Vertical Decompositions
- Alternative Decompositions

- Key Messages
- Key Methodologies
- Open Issues

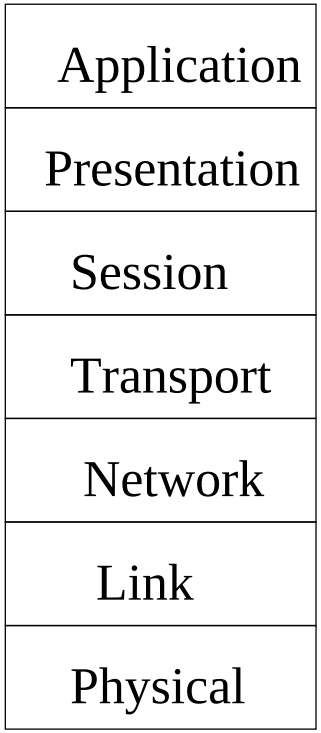
Acknowledgement

Steven Low, Rob Calderbank, John Doyle

M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition," *Proceedings of IEEE*, December 2006.

Stephen Boyd, Lijun Chen, Neil Gershenfeld, Andrea Goldsmith, Prashanth Hande, Jiayue He, Sanjay Hegde, Maryam Fazel, Cheng Jin, Koushik Kar, Jang-Won Lee, Lun Li, Xiaojun Lin, Jiaping Liu, Zhen Liu, Asuman Ozdaglar, Daniel Palomar, Pablo Parrilo, Jennifer Rexford, Devavrat Shah, Ness Shroff, R. Srikant, Chee Wei Tan, Ao Tang, Jiantao Wang, David Wei, Bartek Wydrowski, Lin Xiao, Edmund Yeh, and Junshan Zhang

Layered Network Architecture



Important foundation for data networking

Ad hoc design historically (within and across layers)

Rethinking Layering

How to, and how not to, layer?

A question on architecture: functionality allocation

But want answers to be rigorous, quantitative, simple, and relevant

- How to modularize (and connect)?
- How to distribute (and connect)?
- How to search in the design space of alternative architectures?
- How to quantify the benefits of better codes/modulation/schedule/routes... for network applications?

A common language to rethink these issues?

Layering As Optimization Decomposition

The first unifying view and systematic approach

Network: Generalized NUM

Layering architecture: Decomposition scheme

Layers: Decomposed subproblems

Interfaces: Functions of primal or dual variables

Horizontal and vertical decompositions through

- implicit message passing (e.g., queuing delay, SIR)
- explicit message passing (local or global)

3 Steps: G.NUM $\Rightarrow$ A solution architecture $\Rightarrow$ Alternative architectures

Network Utility Maximization

Basic NUM (KellyMaulloTan98):

$$\begin{array}{ll} \text{maximize} & \sum_s U_s(x_s) \\ \text{subject to} & \mathbf{Rx} \preceq \mathbf{c} \\ & \mathbf{x} \succeq 0 \end{array}$$

Generalized NUM (one possibility shown here) (Chiang05a):

$$\begin{array}{ll} \text{maximize} & \sum_s U_s(x_s, P_{e,s}) + \sum_j V_j(w_j) \\ \text{subject to} & \mathbf{Rx} \preceq \mathbf{c}(\mathbf{w}, \mathbf{P}_e) \\ & \mathbf{x} \in \mathcal{C}_1(\mathbf{P}_e) \\ & \mathbf{x} \in \mathcal{C}_2(\mathbf{F}) \ \text{ or } \ \mathbf{x} \in \Pi \\ & \mathbf{R} \in \mathcal{R} \\ & \mathbf{F} \in \mathcal{F} \\ & \mathbf{w} \in \mathcal{W} \end{array}$$

Two Cornerstones for Conceptual Simplicity

Networks as optimizers

Reverse engineering mentality: give me the solution (an existing protocol), I'll find the underlying problem implicitly being solved

- Why care about the problem if there's already a solution?
- It leads to simple, rigorous understanding for systematic design

Layering as decomposition

1. Analytic foundation for network architecture
2. Common language for thinking and comparing
3. Methodologies, analytic tools

Layering as Optimization Decomposition

What's so unique about this particular framework for cross-layer design?

- Network as optimizer
- End-user application utilities as the driver
- Performance benchmark without any layering
- Unified approach to cross-layer design (it simplifies our understanding about network architecture)
- Separation theorem among modules
- Systematic exploration of architectural alternatives

Utility

Which utility? (function of rate, useful information, delay, energy...)

1. Reverse engineering: TCP maximizes utilities

2a. Behavioral model: user satisfaction

2b. Traffic model: traffic elasticity

3a. Economics: resource allocation efficiency

3b. Economics: different utility functions lead to different fairness

Three choices: Weighted sum, Pareto optimality, Uncooperative game

- Goal: Distributed algorithm converging to globally and jointly optimum resource allocation

- Limitations to be discussed at the end

Layers

Restriction: we focus on resource allocation functionalities

- TCP: congestion control

Different meanings:

- Routing: RIP/OSPF, BGP, wireless routing, optical routing, dynamic/static, single-path/multi-path, multicommodity flow routing...
- MAC: scheduling or contention-based
- PHY: power control, coding, modulation, antenna signal processing...

Insights on both:

- What each layer can do (Optimization variables)
- What each layer can see (Constants, Other subproblems' variables)

Adoption By Industry

Industry adoption of Layering As Optimization Decomposition:

- Internet resource allocation: TCP FAST (Caltech)
- Protocol stack design: Internet 0 (MIT)
- Broadband access: FAST Copper (Princeton, Stanford, Fraser)

This talk is about the underlying common language and methodologies

Network Utility Maximization

Basic NUM (KellyMaulloTan98):

$$\begin{array}{ll} \text{maximize} & \sum_s U_s(x_s) \\ \text{subject to} & \mathbf{Rx} \preceq \mathbf{c} \\ & \mathbf{x} \succeq 0 \end{array}$$

Generalized NUM (one possibility shown here) (Chiang05a):

$$\begin{array}{ll} \text{maximize} & \sum_s U_s(x_s, P_{e,s}) + \sum_j V_j(w_j) \\ \text{subject to} & \mathbf{Rx} \preceq \mathbf{c}(\mathbf{w}, \mathbf{P}_e) \\ & \mathbf{x} \in \mathcal{C}_1(\mathbf{P}_e) \\ & \mathbf{x} \in \mathcal{C}_2(\mathbf{F}) \ \text{ or } \ \mathbf{x} \in \Pi \\ & \mathbf{R} \in \mathcal{R} \\ & \mathbf{F} \in \mathcal{F} \\ & \mathbf{w} \in \mathcal{W} \end{array}$$

Dual-based Distributed Algorithm

Basic NUM with concave smooth utility functions:

Convex optimization (Monotropic Programming) with zero duality gap

Lagrangian decomposition:

$$\begin{aligned} L(\mathbf{x}, \boldsymbol{\lambda}) &= \sum_s U_s(x_s) + \sum_l \lambda_l \left(c_l - \sum_{s:l \in L(s)} x_s \right) \\ &= \sum_s \left[U_s(x_s) - \left(\sum_{l \in L(s)} \lambda_l \right) x_s \right] + \sum_l c_l \lambda_l \\ &= \sum_s L_s(x_s, \lambda^s) + \sum_l c_l \lambda_l \end{aligned}$$

Dual problem:

$$\begin{aligned} \text{minimize} \quad & g(\boldsymbol{\lambda}) = L(\mathbf{x}^*(\boldsymbol{\lambda}), \boldsymbol{\lambda}) \\ \text{subject to} \quad & \boldsymbol{\lambda} \succeq 0 \end{aligned}$$

Dual-based Distributed Algorithm

Source algorithm:

$$x_s^*(\lambda^s) = \text{argmax}\left[U_s(x_s) - \lambda^s x_s\right], \quad \forall s$$

- Selfish net utility maximization locally at source s

Link algorithm (gradient or subgradient based):

$$\lambda_l(t+1) = \left[\lambda_l(t) - \alpha(t)\left(c_l - \sum_{s:l\in L(s)} x_s(\lambda^s(t))\right)\right]^+, \quad \forall l$$

- Balancing supply and demand through pricing

Certain choices of step sizes $\alpha(t)$ of distributed algorithm guarantee convergence to globally optimal $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$

Primal-Dual

Different meanings:

- Primal-dual interior-point algorithm
- Primal-dual solution
- Primal or dual driven control
- Primal or dual decomposition

Coupling in constraints (easy: flow constraint, hard: SIR feasibility)

Coupling in objective (easy: additive form, hard: min max operations)

Primal Decomposition

Simple example:

$$x + y + z + w \leq c$$

Decomposed into:

$$\begin{aligned} x + y &\leq \alpha \\ z + w &\leq c - \alpha \end{aligned}$$

New variable α updated by various methods

Interpretation: Direct resource allocation (not pricing-based control)

Engineering implications: Adaptive slicing (GENI)

Pricing feedback: dual decomposition

Adaptive slicing: primal decomposition

Horizontal Decompositions

Reverse engineering:

- Layer 4 TCP congestion control: Basic NUM (LowLapsley99, RobertsMassoulie99, MoWalrand00, YaicheMazumdarRosenberg00, KunniyurSrikant02, LaAnatharam02, LowPaganiniDoyle02, Low03, Srikant04...)
- Layer 4 TCP heterogeneous protocol: Nonconvex equilibrium problem (TangWangLowChiang05)
- Layer 3 IP inter-AS routing: Stable Paths Problem (GriffinSheperdWilfong02)
- Layer 2 MAC backoff contention resolution: Non-cooperative Game (LeeChiangCalderbank06a)

Forward engineering for horizontal decompositions also carried out recently

Vertical Decompositions

A partial list of work along this line:

- Jointly optimal congestion control and adaptive coding or power control (Chiang05a, LeeChiangCalderbank06b)
- Jointly optimal congestion and contention control (KarSarkarTassiulas04, ChenLowDoyle05, WangKar05, YuenMarbach05, ZhengZhang06, LeeChiangCalderbank06c)
- Jointly optimal congestion control and scheduling (ErilymazSrikant05)
- Jointly optimal routing and scheduling (KodialamNandagopal03)
- Jointly optimal routing and power control (XiaoJohanssonBoyd04, NeelyModianoRohrs05)
- Jointly optimal congestion control, routing, and scheduling (LinShroff05, ChenLowChiangDoyle06)

Vertical Decompositions

- Jointly optimal routing, scheduling, and power control (CruzSanthanam03, XiYeh06)
- Jointly optimal routing, resource allocation, and source coding (YuYuan05)
- TCP/IP interactions (WangLiLowDoyle05, HeChiangRexford06) and jointly optimal congestion control and routing (KellyVoice05, Hanetal05)
- Network lifetime maximization (NamaChiangMandayam06)
- Application adaptation and congestion control/resource allocation (ChangLiu04, HuangLiChiangKatsaggelos06)

Vertical Decompositions

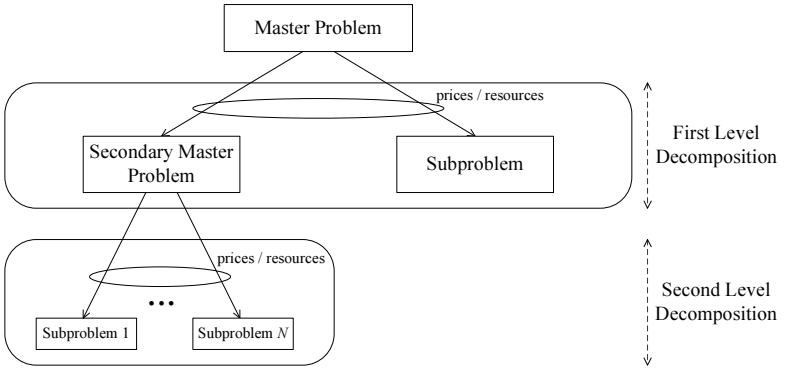
- Specific designs not important
- Common language and key messages methodologies important

Goal: Shrink, not grow knowledge tree on cross-layer design

Alternative Decompositions

Many ways to decompose:

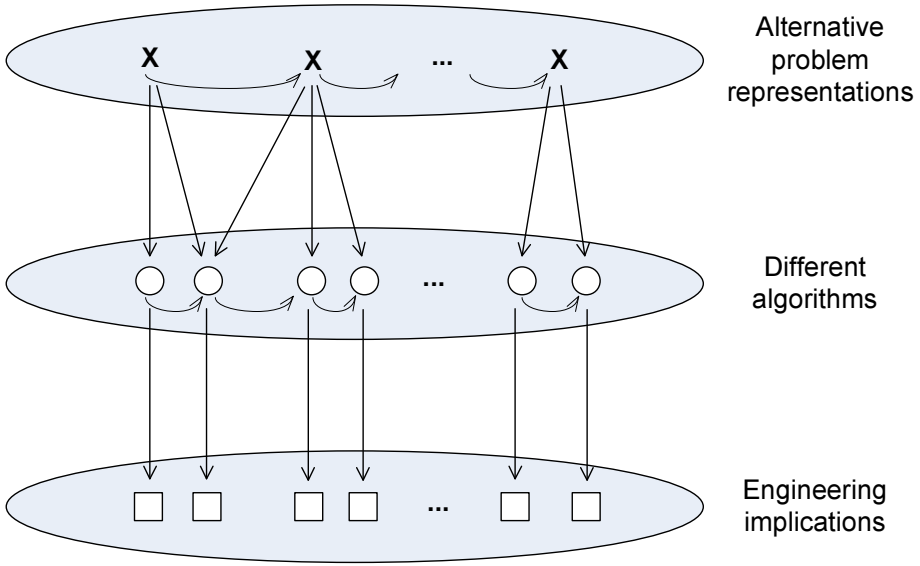
- Primal and dual decomposition
- Partial decomposition
- Multi-level decomposition



Lead to alternative architectures (PalomarChiang06) with different

- Communication overhead
- Computation distribution
- Convergence behavior

Alternative Decompositions



Systematically explore the space of alternative decompositions

Key Messages

- Protocols in layers 2,3,4 can be reverse engineered. Reverse engineering in turn leads to better design in a rigorous manner.
- There is a unifying approach to cross-layer design, illustrating both opportunities and risks.
- Loose coupling through "layering price" can be optimal and robust, and congestion price (or queuing delay, or buffer occupancy) is often the right "layering price" for stability and optimality, with important exceptions as well.
- User-generated pricing following end-to-end principle
- There are many alternatives in decompositions, leading to different divisions of tasks across layers and even different time-scales of interactions.
- Convexity of the generalized NUM is the key to devising a globally optimal solution.
- Decomposability of the generalized NUM is the key to devising a distributed solution.

Key Methodologies

- Dual decomposition for linear coupling constraints.
- Consistency pricing for coupled objective functions.
- Descent lemma for proof of convergence of dual-based distributed subgradient algorithm.
- Stability proof through Lyapunov function construction, singular perturbation theory, and passivity argument.
- Log change of variables to turn multiplicative coupling into linear coupling, and to turn nonconvex constraints to convex ones.
- Sufficient conditions on curvature of utility functions for it to remain concave after a log change of variables.
- Construction of conflict graph, contention matrix, and transmission modes in contention based MAC design.
- Maximum differential congestion pricing for node-based back-pressure scheduling (part of the connections between distributed convex optimization and stochastic control).

Future Research Issues

- Technical: Stability under delay...
- Modeling: routing in ad hoc network, ARQ, MIMO...
- Time issues
- Why deterministic fluid model?

Shannon 1948: remove finite blocklength, Law of Large Numbers kicks in (later finite codewords come back...)

Kelly 1998: remove coupled queuing dynamics, optimization and decomposition view kicks in (later stochastics come back...)

- What if it's not convex optimization?

Rockafellar 1993: Convexity is the watershed between easy and hard (what if it's hard?)

- Is performance the only optimization objective?

Future Research: Time Issues

- Rate of convergence
- Timescale separation
- Transient behavior bounding
- Utility as a function of latency
- Utility as a function of transient rate allocations

Future Research: Stochastic Issues

	Stability or Validation	Average Performance	Outage Performance	Fairness
Session Level	$\star\star$	$\star$		$\star$
Packet Level	$\star$	$\star$		
Channel Level	$\star\star$	$\star$		
Topology Level				

Table 1: State-of-the-art in Stochastic Network Utility Maximization.

With a good layering architecture:

- Stochastic doesn't hurt
- Stochastic may help

More from Junshan and Ness talks

Future Research: Nonconvexity Issues

- Nonconcave utility (eg, real-time applications)
- Nonconvex constraints (eg, power control in low SIR)
- Integer constraints (eg, single-path routing)
- Exponentially long description length (eg, scheduling)

Convexity not invariant under embedding in higher dimensional space or nonlinear change of variable

- Sum-of-squares method (Stengle73, Parrilo03)
- Geometric programming (DuffinPetersonZener67, Chiang05b)

From optimal/complicated to suboptimal/simple modules (LinShroff05)

Future Research: Network X-ities Issues

From Bit to Utility to Control and Management

Over-optimized? Optimizing for what?

- Evolvability
- Scalability
- Diagnosability

Pareto-optimal tradeoff between Performance and Network X-ities

From Forward Engineering to Reverse Engineering to

- Design for Optimizability

New Mentalities

Layering As Optimization Decomposition, but move away from:

- One architecture fits all
- Deterministic fluids
- Asymptotic convergence
- Optimality
- Optimization

Think about "right" decomposition in the "right" way

Contacts

chiangm@princeton.edu
www.princeton.edu/∼chiangm