Avalanche: เมื่อไมโครซอฟท์หันมามองโลก P2P

วสันต์ ลิ่วลมไพศาล
lewcpe at gmail dot com

ในช่วงไม่กี่ปีมานี้ คงต้องยอมรับว่ากระแสโปรโตคอลการรับส่งไฟล์แบบ P2P[1] (Peer-to-Peer) นั้นมาแรงชนิดที่จะคนเล่นอินเทอร์เน็ตแล้วไม่รู้จักคงมีกันน้อยคนเต็มที โดยเฉพาะโปรโตคอลยอดนิยมอย่างบิตทอร์เรนต์ (BitTorrent) ที่ว่ากันว่ากินแบนด์วิดท์รวมของโลกไซเบอร์ในวันนี้ไปถึงหนึ่งในสามแล้ว

แต่วันดีคืนดีเมื่อบริษัทที่ดูไม่มีทีท่าว่าจะสนใจเทคโนโลยีเหล่านี้อย่างไมโครซอฟท์เข้ามาประกาศว่าได้ปรับปรุงเทคโนโลยีนี้เพื่อให้มีความสามารถสูงขึ้นอีก 20-30% ความน่าสนใจของเรื่องนี้ก็ดึงให้ทั่วโลกต้องจับตามองกันว่า ไมโครซอฟท์ปรับปรุงโปรโตคอลได้อย่างว่าจริงหรือไม่ และหากปรับปรุงได้แล้วทำไมบริษัทซอฟท์แวร์ยักษ์ใหญ่เช่นนี้จึงสนใจในเทคโนโลยี P2P ในวันนี้เราจะมาแนะนำถึงวิธีการที่ไมโครซอฟท์ใช้มาปรับปรุง P2P พร้อมๆ สำรวจข่าวคราวและความเป็นไปได้ที่เราจะได้เห็นโปรโตคอลนี้ในอนาคตกัน

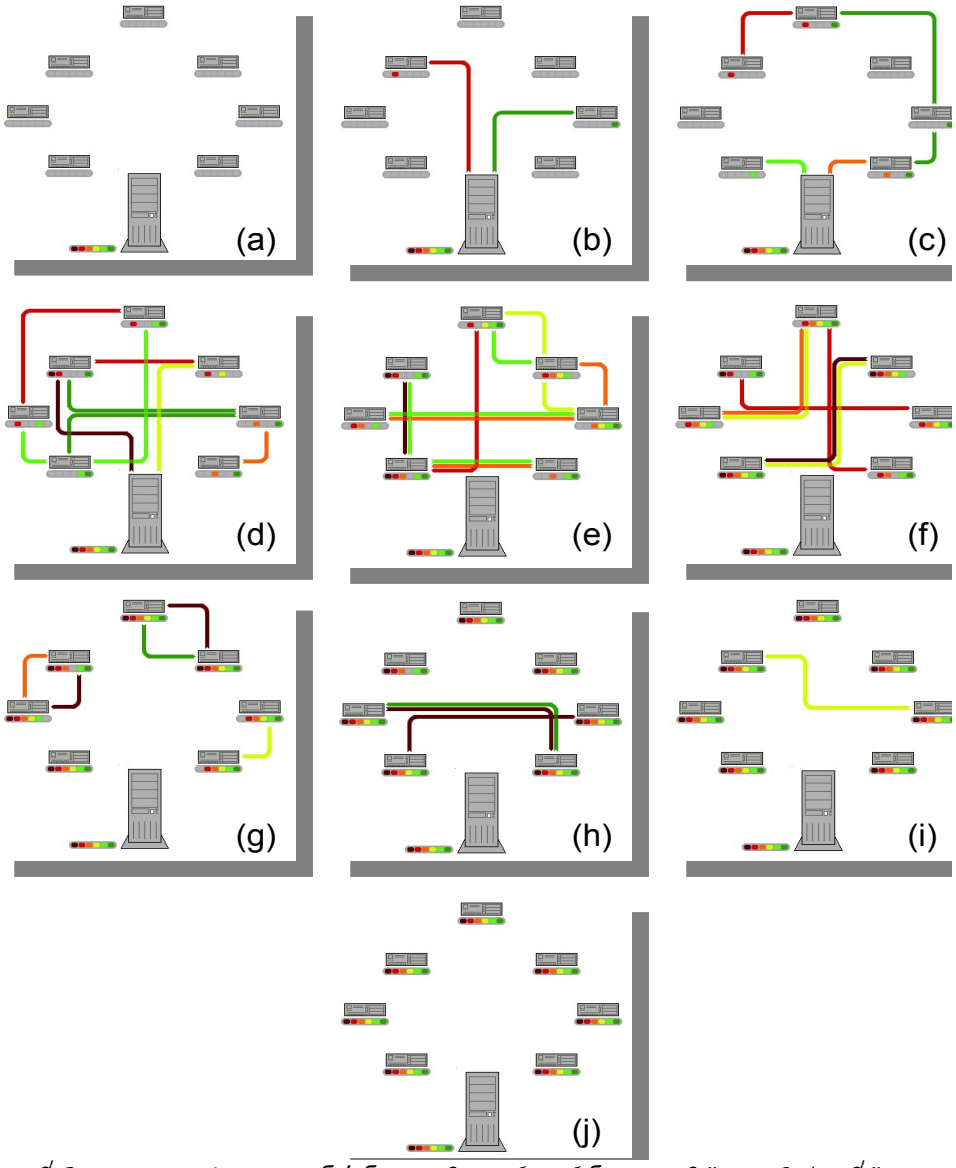
การทำงานของบิตทอร์เรนต์

ในวันนี้โลกของ P2P นั้นผู้ครองตลาดส่วนมากคงเป็นใครไปไม่ได้นอกจากบิตทอร์เรนต์ ที่เรารู้จักกันดีเนื่องจากเป็นมาตรฐานเปิด และมีประสิทธิภาพสูงมาก ทำให้บิตทอร์เรนต์ได้รับความนิยมเป็นอย่างสูงในปัจจุบัน

ภาพที่ 1: โลโก้บิตทอร์เรนต์

บิตทอร์เรนต์นั้นสร้างขึ้นด้วยแนวคิดพื้นฐานของระบบ P2P ทั่วไปคือแบ่งไฟล์ออกเป็นส่วนๆ แล้วค่อยโหลดส่วนที่ตนขาดอยู่จากผู้อื่น ซึ่งเราจะรู้ว่ามีผู้ใดบ้างที่มีไฟล์เดียวกับเราก็จากการประกาศของศูนย์กลางที่เรียกว่าแทรกเกอร์ (Tracker) ซึ่งโดยทั่วๆ ไปแล้วแทรกเกอร์มักจะไม่ให้สมาชิกทั้งหมดรู้จักกันเพราะโดยมากแล้วไม่มีความจำเป็น เนื่องจากไฟล์หนึ่งๆ อาจะมีจำนวนผู้โหลดหลายพันหลายหมื่นคน แต่โดยมากแล้ว แบนด์วิทของแต่ละคนนั้นสามารถเชื่อมต่อกับปลายทางได้ไม่ถึงร้อยคนก็เต็มแบนด์วิทแล้ว

1 P2P คือเครือข่ายการส่งข้อมูลที่อาศัยเครื่องลูกข่ายให้บริการต่างๆ มากกว่าที่จะพึ่งพาจากเครื่องแม่เป็นหลัก ข้อมูลเพิ่มเติมเกี่ยวกับเครือข่าย P2P สามารถค้นหาได้จาก http://en.wikipedia.org/wiki/P2P



ภาพที่ 2: แสดงการทำงานของโปรโตคอลบิตทอร์เรนต์ โดยสมมติข้อมูล 6 ส่วนที่ต้องการกระจายไปยังเครื่องจำนวน 7 เครื่อง จะเห็นว่าเครื่องแม่นั้นกระจายแต่ละส่วนออกไปเพียงครั้งเดียว แล้วปล่อยให้เครื่องในเครือข่ายแลกข้อมูลกันและกันจนกว่าทุกเครื่องจะได้ข้อมูลครบทุกส่วน

ที่มาภาพ: http://upload.wikimedia.org/wikipedia/en/3/3d/Torrentcomp_small.gif

ขั้นตอนการดาวน์โหลดของบิตทอร์เรนต์คือ เมื่อเครื่องหนึ่งเชื่อมต่อเข้าหาเครือข่าย มันจะเลือกขอก้อนของข้อมูลซึ่งถูกแบ่งเป็นก้อนเล็กๆ โดยการเลือกก้อนแบบสุ่ม เมื่อได้ส่วนของข้อมูลมาแล้ว มันจะพิจารณาว่าก้อนใดบ้างที่ ในบริเวณที่มันเชื่อมต่ออยู่ด้วยนั้นขาดแคลนที่สุด หรือที่เรียกว่าการเลือกข้อมูลจากความขาดแคลนในท้องถิ่น (Locally Rare) เมื่อคำนวณแล้วจึงพยายามขอส่วนนั้นมาโดยอาจต้องเอาส่วนที่มีอยู่เดิมไปแลกมา[2] จนได้ไฟล์ทั้งหมดครบ

ข้อเสียของบิตทอร์เรนต์

ขั้นตอนการทำงานทั้งหมดของบิตทอร์เรนต์ทำให้บิตทอร์เรนต์เป็นโปรโตคอลที่มีประสิทธิภาพสูงที่สุดอันหนึ่งในปัจจุบัน แต่ในโลกการใช้งานจริงแล้ว เมื่อเครือข่ายมีขนาดใหญ่มากๆ เราพบว่าบางครั้ง มีข้อมูลบางส่วนไม่ได้ถูกกระจายไปโดยทั่วถึง ทำให้เกิดความขาดแคลนข้อมูลส่วนนี้ไปทั่วเครือข่าย (Globally Rare) โดยสุดท้ายแล้วทุกคนในเครือข่ายต้องรอข้อมูลส่วนนี้เพื่อให้ได้ข้อมูลที่ครบถ้วน

มีผู้เสนอหนทางแก้ไขกรณีที่เป็นปัญหานี้หลายหนทางด้วยกัน ยกตัวอย่างเช่น การให้ศูนย์กลางเครือข่ายคำนวณว่ามีข้อมูลส่วนใดที่มีความขาดแคลนอยู่ แล้วจึงแจ้งไปยังลูกข่ายเพื่อให้ลูกข่ายช่วยกันกระจายข้อมูลส่วนนั้นออกไป ซึ่งการแก้ปัญหาเช่นนี้ทำให้เครื่องศูนย์กลางจำเป็นต้องมีประสิทธิภาพสูงมากๆ เพื่อที่จะคำนวณหาค่าความขนาดแคลนของแต่ละส่วนออกมาได้

อีกกรณีหนึ่งคือการที่ทุกส่วนของข้อมูลในบิตทอร์เรนต์นั้นแยกกันเป็นอิสระ ทำให้เกิดความเสี่ยงที่จะให้มีผู้ไม่หวังดีใส่ข้อมูลขยะเข้ามาในเครือข่ายได้ ซึ่งเมื่อข้อมูลขยะนีเข้ามาในเครือข่ายแล้ว มันจะถูกกระจายออกไปจนทำให้มีผู้ได้รับข้อมูลที่ใช้งานไม่ได้เป็นจำนวนมาก

การสร้างรหัสแบบเครือข่าย

จากความบกพร่องของเครือข่ายบิตทอร์เรนต์ มีนักวิจัยจำนวนมาก พยายามเสนอหนทางแก้ปัญหาทั้งหลายที่เกิดขึ้น วิธีหนึ่งที่ได้รับความสนใจโดยเฉพาะจากไมโครซอฟท์ คือการสร้างรหัสแบบเครือข่าย[3] (Network Coding) ซึ่งมุ่งสร้างส่วนของข้อมูลซึ่งเป็นรหัสที่สามารถแก้กลับออกมาเป็นข้อมูลได้เมื่อมีจำนวนรหัสมากพอ

เพื่อที่จะเข้าใจในการสร้างรหัสแบบเครือข่าย ให้เราลองนึกถึงระบบสมการเชิงเส้น ซึ่งบอกว่าถ้าเรามีตัวแปรไม่ทราบค่า x อยู่ n จำนวนเป็น $x_1, x_2, x_3, .., x_n$ เพื่อที่จะหาค่าของ x เหล่านี้ เราต้องมีสมการเป็น n สมการ เพื่อที่จะสามารถหาทุกๆ x กลับออกมาได้

2 เรียกอย่างเป็นทางการว่า Tit-for-Tat เป็นกติการในการใช้เครือข่ายที่พัฒนาขึ้นจากทฤษฎีเกม (Game Theory) ของจอห์น แนช ที่เป็นที่รู้จักกันจากภาพยนต์ชีวประวัติของเขา เรื่อง The Beautiful Mind ข้อมูลเพิ่มเติมเกี่ยวกับกติกาการแลกเปลี่ยนข้อมูลนี้สามารถหาได้จาก http://en.wikipedia.org/wiki/Tit-for-tat

3 ถูกเสนอขึ้นมาครั้งแรกโดย R. Ahlswede ในผลงานวิจัยที่ชื่อว่า Network Information Flow ได้รับตีพิมพ์ในวารสาร IEEE Transactions on Information Theory ฉบับ IT-46 หน้าที่ 1204-1206 ปี ค.ศ. 2000

$$2x_1 + x_2 = 0$$
$$x_1 + x_2 = 1$$

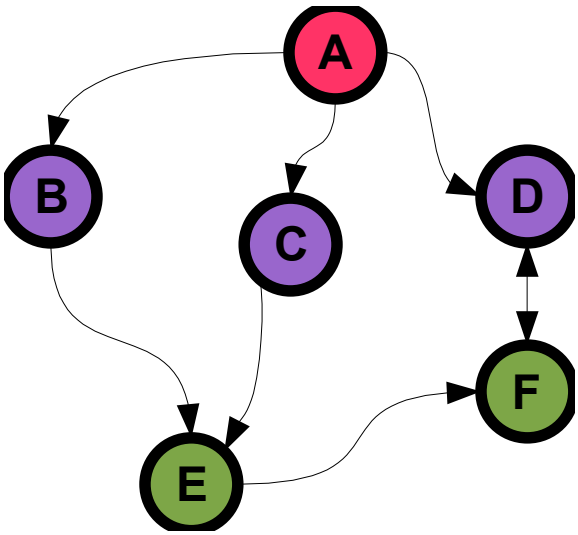
ภาพที่ 3: ตัวอย่างสมการ 2 ตัวแปร

ยกตัวอย่างเช่นเรามีตัวแปรไม่ทราบค่าอยู่ 2 ตัว และมี 2 สมการดังภาพที่ 3 เราจะรู้ได้โดยการแก้สมการว่า $x_1 = -1$ และ $x_2 = 2$

และด้วยหลักการเดียวกันนี้เอง หากเรามองตัวแปรไม่ทราบค่า x ให้เป็นข้อมูลที่เราต้องการส่งจริงๆ เราจะพบว่าเราไม่ต้องส่งค่า -1 และ 2 นี้ออกไปในเครือข่ายเลย แต่เราต้องการส่งเพียงค่าสัมประสิทธ์ไปพร้อมๆ กับผลบวกของสมการเท่านั้น และเมื่อปลายทางได้รับสมการจำนวนมากพอแล้ว เราจะพบว่าปลายทางสามารถแก้ข้อมูลออกมาได้เอง หลักการนี้ทำให้แต่ละส่วนของเครือข่ายจะไม่ต้องกังวลในวิธีการที่จะเลือกส่วนของข้อมูลที่จะส่งอีกต่อไป

ข้อดีของการสร้างรหัสแบบเครือข่าย

ต่อไปเราจะเปรียบเทียบความแตกต่างระหว่างโปรโตคอลบิตทอร์เรนต์ กับโปรโตคอลที่ใช้การสร้างรหัสแบบเครือข่าย เพื่อความง่ายเราจะยกตัวอย่างดังภาพที่ 4

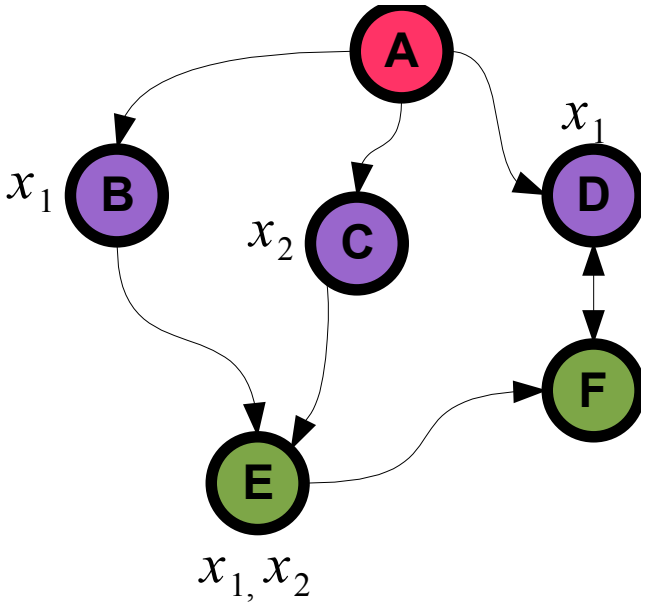


ภาพที่ 4: แสดงตัวอย่างเครือข่ายในการส่งข้อมูลโดยเริ่มส่งจากเครื่อง A

เรายกตัวอย่างการส่งค่าตัวแปรไม่ทราบค่าเช่นในสมการจากภาพที่ 3 เป็นขั้นดังต่อไปนี้

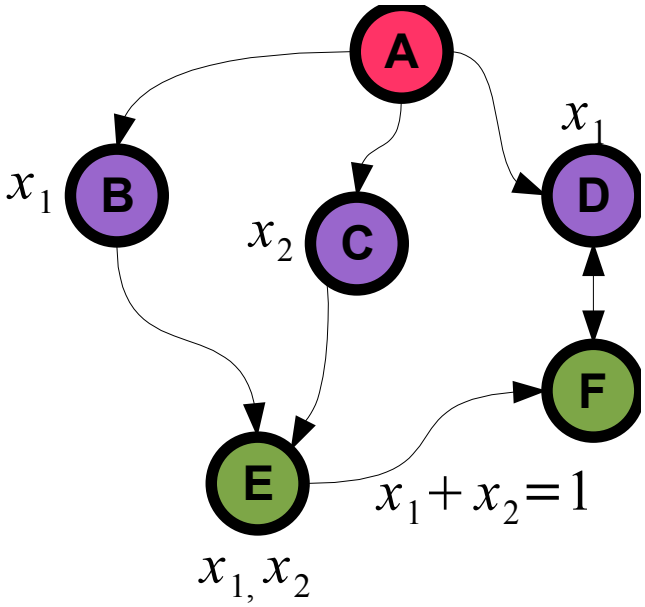
1. เราให้ให้เครื่อง A ส่งข้อมูลให้เครื่อง B, C และ D เป็นค่า x_1 , x_2 และ x_1 ตามลำดับ
2. ต่อมาเครื่อง B และ C ส่งข้อมูลส่วนที่ตนมีให้กับเครื่อง E ทำให้เครื่อง E เป็นเครื่องแรก

3. เครื่อง F สามารถเลือกที่จะรับข้อมูลจากเครื่อง E หรือ C ก็ได้



ภาพที่ 5: แสดงข้อมูลเมื่อเวลาผ่านไปจนมีเครื่อง E เป็นเครื่องแรกในเครือข่ายที่มีข้อมูลครบ

เมื่อเครื่อง F สามารถเลือกที่จะรับข้อมูลได้ใน ในโปรโตคอลบิตทอร์เรนต์ทำให้เครื่อง F จำเป็นต้องเลือกเอาข้อมูลส่วนใดส่วนหนึ่งเพียงส่วนเดียว โดยเครื่อง F อาจเลือกรับข้อมูล x_1 หรือ x_2 ก็ได้ แต่หากเครื่อง F เลือกส่วน x_1 จะทำให้เครื่อง D ไม่สามารถร้องขอข้อมูลที่เป็นประโยชน์จากเครื่อง F ได้ทำให้เครื่อง D ต้องร้องขอข้อมูล x_2 จากเครื่องอื่นๆ ซึ่งส่วนนี้เป็นข้อเสียหลักของบิตทอร์เรนต์ในปัจจุบันตามคำอ้างของผู้วิจัยที่ไมโครซอฟท์



ภาพที่ 6: แทนที่เครื่อง F จะต้องเลือกข้อมูลจากเครื่อง E เครื่อง E กลับส่งสมการมาให้เครื่อง F ซึ่งจะทำให้เครื่อง F สามารถแก้ข้อมูลทั้งหมดออกมาได้ในอนาคต

แต่ในระบบสร้างรหัสแบบเครือข่าย แทนที่เครื่อง E จะให้เครื่อง F เลือกที่จะรับข้อมูลส่วนใด เครื่อง E กลับสร้างสมการ $x_1 + x_2 = 1$ แล้วส่งไปให้เครื่อง F ในเวลาต่อมาเมื่อเครื่อง D ส่งข้อมูล x_1 มาให้เครื่อง F แล้ว เราจะพบว่าเครื่อง F สามารถส่งข้อมูลใดๆ กลับไปให้เครื่อง D เพื่อให้เครื่อง D สามารถสร้างข้อมูลเดิมกลับออกมาได้เช่นเดียวกัน

ประสิทธิภาพของระบบรหัสแบบเครือข่าย

แม้ว่าการส่งค่าสัมประสิทธิ์ของสมการในหัวข้อที่แล้วจะดูเหมือนว่าจะทำให้ต้องส่งข้อมูลจำนวนมาก ซึ่งก็จะทำให้สิ้นเปลืองแบนวิดท์อยู่ดี แต่ในความเป็นจริงแล้ว เราสามารถส่งข้อมูลทั้งหมดโดยเสียพื้นที่เพิ่มเติมเพียงประมาณ 3% เท่านั้น[4]

เอกสารของไมโครซอฟท์อ้างว่าอวาลานซ์[5] (Avalanche) นั้นมีประสิทธิภาพสูงกว่าบิตทอร์เรนต์ 20-30% ทีเดียว ด้วยข้อได้เปรียบทีได้แสดงให้เห็นในหัวข้อที่แล้ว อย่างไรก็ตาม อวาลานซ์ยังคงเป็นเพียงแบบจำลองอยู่ในซอฟท์แวร์เครื่องเดียว โดยยังไม่ได้ทดสอบจากการใช้งานในโลกความเป็นจริง ซึ่งในจุดนี้นายแบรม โคเฮน[6] ผู้ก่อตั้งบิตทอร์เรนต์ก็ออกมาวิจารย์ถึงจุดนี้ เนื่องจากมีโปรโตคอลจำนวนมากในโลกที่ล้มเหลวเมื่อนำมาใช้งานจริง

4 P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," 51st Allerton Conf. Communication, Control and Computing, Oct. 2003.

5 อวาลานซ์ (Avalanche) เป็นชื่อที่ไมโครซอฟท์ใช้เรียกโปรแกรมจริงที่ไมโครซอฟท์ได้สร้างขึ้นตามโมเดลการส่งข้อมูลที่ไมโครซอฟท์ได้เสนอขึ้นมานี้ โดยคำนี้แปลตรงตัวว่าปริมาณที่มากมหาศาล ซึ่งน่าจะมาจากการล้อเลียนกับบิตทอร์เรนต์ที่แปลว่ากระแสที่รุนแรง

6 แบรม โคเฮน (Bram Cohen) เป็นผู้สร้างบิตทอร์เรนต์ขึ้นมาในปีค.ศ. 2001 ปัจจุบันเขาทำงานเต็มเวลาเพื่อพัฒนาบิตทอร์เรนต์

โคเฮนระบุว่าประสิทธิภาพนั้นไม่ได้มาจากเพียงแค่ข้อมูลที่ถูกส่งอย่างมีประสิทธิภาพเท่านั้น หากแต่ต้องการความเร็วของเครื่องปลายทางอีกด้วย ในทุกวันนี้แม้แต่บิตทอร์เรนต์เองก็ยังพบปัญหาที่โปรแกรมลูกข่ายจำนวนมากไม่สามารถทำงานอย่างมีประสิทธิภาพได้ ซึ่งการสร้างรหัสจากข้อมูลใหญ่ๆ นั้นก็ดูเหมือนจะต้องใช้ประสิทธิภาพเครื่องที่สูงเอามากทีเดียว

ความสนใจของไมโครซอฟท์

การเข้ามาของไมโครซอฟท์นั้นเข้าใจได้ว่าไมโครซอฟท์เองต้องการลดค่าใช้จ่ายในด้านของแบนด์วิทด์ของไมโครซอฟท์เองที่เพิ่มขึ้นทุกวัน โดยหากระบบนี้ประสบความสำเร็จ จะทำให้ไมโครซอฟท์สามารถลดค่าใช้จ่ายในการแจกจ่ายแพทซ์ต่างๆ ตลอดจนโปรแกรมฟรีต่างๆ ของไมโครซอฟท์ได้โดยมีค่าใช้จ่ายที่ต่ำลง

ไมโครซอฟท์เองมีการออกมาตราการประหยัดแบนด์วิทด์ออกมาบ้างในช่วงปีที่ผ่านมา โดยมีการลดข้อมูลในการให้บริการข่าวสารผ่านทาง XML อย่าง RSS โดยในทุกวันนี้ไมโครซอฟท์ต้องรับภาระให้บริการดาวน์โหลดแพตซ์ต่างๆ แทบทุกวัน

อนาคต

ในอนาคตอันใกล้ ไมโครซอฟท์อาจจะมีการเปิดบริการให้ดาวน์โหลดแพทซ์สำคัญๆ อย่าง Critical Patch ซึ่งมีออกมาเรื่อยๆ และไมโครซอฟท์ให้บริการกับทุกคนแม้แต่ผู้ใช้วินโดว์ผิดลิขสิทธิ์ แต่กับโปรแกรมใหญ่ๆ อย่างเช่นการให้ดาว์นโหลดโปรแกรมอย่าง Acrylic Beta หรือโปรแกรมอื่นๆ นั้นเป็นเรื่องน่าสนใจว่าไมโครซอฟท์จะแจกโปรแกรมเหล่านี้ผ่านทางเครือข่ายอวาลานซ์ด้วยหรือไม่ เนื่องจากไมโครซอฟท์เองคงต้องการควบคุมการแจกจ่ายโปรแกรมเหล่านี้ไว้ในมือ

ในส่วนของตัวโปรโตคอลนั้น หากใช้งานได้จริงเรื่องน่าสงสัยคือไมโครซอฟท์นั้นจะใจดีพอที่จะเปิดมาตรฐานของตนออกเป็นมาตรฐานเปิดรึเปล่า ซึ่งแม้จะไมโครซอฟท์จะพยายามปิดกั้นเพียงไร แต่ผมเองก็เชื่อว่าสุดท้ายแล้วต้องมีโปรแกรมที่เข้ากันได้ทั้งแม่ข่าย และลูกข่ายออกมาให้เราดาวน์โหลดกันต่อไป แต่สุดท้ายแล้วเครือข่ายแบบรหัสนี้น่าจะได้รับความนิยมในที่สุด ไม่ว่าจะเป็นแบบเข้ากันได้กับไมโครซอฟท์หรือฝั่งโอเพ่นซอร์สจะสร้างขึ้นมาใหม่ก็ตามที

BLOGNONE Tech News That's Worth

SOME RIGHTS RESERVED

© สงวนลิขสิทธิ์ พ.ศ. 2548 วสันต์ ลิ่วลมไพศาล อนุญาตให้เผยแพร่ได้โดยต้องระบุที่มา, ต้องเผยแพร่เอกสารเหมือนต้นฉบับ และห้ามเผยแพร่เพื่อการค้า

เอกสารฉบับนี้ถูกสร้างขึ้นจากโปรแกรม OpenOffice.org Writer เวอร์ชั่น 1.9.109